Thèse n° 9939

Multi-agent Learning with Privacy Guarantees

Présentée le 22 juin 2023

Faculté des sciences et techniques de l'ingénieur
Laboratoire de Systèmes Adaptatifs
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Elsa RIZK

Acceptée sur proposition du jury

Prof. M. Jaggi, président du jury
Prof. A. H. Sayed, directeur de thèse
Prof. V. Matta, rapporteur
Prof. A. Zoubir, rapporteur
Prof. V. Cevher, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

To my family…

Acknowledgements

I would like to express my deepest gratitude to all those who have contributed to the completion of this thesis.

First and foremost, I would like to thank my thesis advisor, Professor Ali H. Sayed, for his unwavering support, guidance, and encouragement throughout this journey. His knowledge, expertise, and patience were invaluable, and I could not have completed this thesis without his mentorship. He motivated me to exceed what I believed were my limitations. Among other competencies, I have honed my writing and communication skills, which I plan to apply in my future professional endeavors.

I would also like to extend my appreciation to the members of my thesis committee, Professor Martin Jaggi, Professor Volkan Cevher, Professor Vincenzo Matta, and Professor Abdelhak Zoubir, for their valuable feedback and insightful comments. Their expertise and critical insights helped fine-tune this thesis, and I am grateful for their time and effort.

I next would like to thank all the collaborators and colleagues I had the pleasure of meeting from the Adaptive Systems Lab at EPFL, Virginia Bordignon, Haoyuan Cai, Ying Cao, Lucas Cassano, Ping Hu, Mert Kayaalp, Malek Khammassi, Professor Visa Koivunen, Professor Ricardo Merched, Professor Roula Nassif, Konstantinos Ntemos, Flávio Pavan, Augusto Santos, Valentina Shumovskaia, Professor Stefan Vlaski, Professor Kun Yuan, Ainur Zhaikhan, and anyone else that might have worked with the lab. The discussions we had were instrumental in shaping the findings of this thesis. The sense of camaraderie among the lab members fostered a positive lab culture that facilitated collaboration to thrive rather than competition. My appreciation extends to Patricia Vonlanthen for her assistance with administrative tasks.

Next, I would like to thank all my friends made here in Switzerland and the ones across the globe. They have been crucial in re-energizing me whenever the work was demotivating. The fun times we shared helped me take my mind-off of what could sometimes be a challenging journey. I thank Ahmad, Bernd, Christina, Daniella, Danielle, Frank, Jacques, Karen, Maissa, Marie-Line, Maya, Nadine, Roman, Roula, Sarah, Tania, Virginia, the EPIC committee members, and anyone else who I met in Lausanne with whom many laughs were shared.

Finally, I am forever grateful for my family. During difficult times, they have been my support system and knew how to alleviate my stress. I want to thank my parents who provided me with the strength, guidance, and protection I needed to push with my work. I want to thank my sister Yara who acted as an advisor, supporter, and role model. I am also grateful for having my sister Lynn and her husband Jose with me in Lausanne. They offered a local safe-haven for me whenever the stresses of this journey were taking over. If it were not for my family, I would not be standing today with a completed thesis, which is why I dedicate this work to them.

To everyone who has played a part in this thesis, I offer my sincere thanks. Your contributions have been immeasurable, and I am grateful for your support and guidance along the way.

Lausanne, February 2023 E. R.

Abstract

A multi-agent system consists of a collection of decision-making or learning agents subjected to streaming observations from some real-world phenomenon. The goal of the system is to solve some global learning or optimization problem in a distributed or decentralized manner, where agents collect data locally and interact either with their neighbours or with some central processor.

Such multi-agent systems are prevalent in multiple real-world applications, such as autonomous driving, multi-robot systems, multi-sensor systems, target surveillance, and disaster response, to name a few. Decentralized and distributed solutions are often motivated by the nature of the system, e.g., weather data is naturally distributed across different geographic locations, or these solutions are preferable due to their enhanced robustness to link and node failures.

In designing multi-agent systems, one normally formulates a global risk function, consisting of the aggregate of local risks, and then seeks to approximate its optimizer through localized interactions among neighbouring agents. During this process, the agents will be required to share processed data and/or iterates with their neighbours. The issue of privacy then becomes critical in enabling the safe communication of information over edges linking the elements of the multi-agent network. There have been several works in the literature that enforce privacy by adding random noise sources on top of the shared information. Most of these works establish that the resulting architectures are differentially private, but they assume that the gradients of the risk functions are bounded. Unfortunately, this condition is rarely valid in practice and this fact is often overlooked in most studies. For example, even quadratic risk functions have unbounded gradients because these gradients will be affine functions of the unknown parameter. Moreover, most studies fail to recognise that their differentially private solutions and the added noise sources end up degrading the mean-square error (MSE) performance of the learning algorithms from $O(\mu)$ down to $O(\mu^{-1})$, where μ is the small learning parameter. These are serious limitations that remain unaddressed in existing approaches to differential privacy in multi-agent systems.

In this dissertation, we resolve these two issues. First, we do not assume bounded gradients for the risk functions. And yet, we are still able to establish that the multi-agent systems remain differentially private, albeit with high probability. We achieve this conclusion by showing that the noise sources should not be added in an *ad-hoc* manner, as is common in existing approaches, but rather that they should be constructed in a manner that is cognizant of the graph topology. Otherwise, the noises end up generating a magnifying effect that degrades performance. For this reason, we introduce a *locally* homomorphic noise construction and show that, under this process, the MSE performance of the multi-agent system will remain at $O(\mu)$ while being differentially private at the same time. This is a reassuring conclusion. We illustrate these results for the special case of federated learning architectures, but also extend them to more general distributed learning and optimisation strategies over decentralised architectures.

Motivated by these considerations, the first part of the dissertation studies a particular distributed multi-agent system, known as federated learning (FL). The federated setting consists of a central server that orchestrates the learning process among a collection of edge devices. We refer to this set-up as a distributed architecture (as opposed to the decentralized architecture without central processing, studied in the third part of the dissertation). Some use cases of the technology can be found in the healthcare industry, the insurance sector, IoT applications, and other technologies such as predictive text or voice recognition. We examine a couple of questions related to the performance of the FL algorithm. First, we establish its convergence under three demanding characteristics related to data heterogeneity, asynchronous operation, and partial agent participation. Then, we show how performance can be improved by introducing a mechanism based on importance sampling to choose the participating agents and the data samples during each stage of the learning process in some optimized manner. Finally, we introduce a privatization layer and explain how its implementation does not degrade performance.

The second part of the dissertation introduces a more realistic architecture for the federated setting. Instead of assuming a *single-server* structure, we now consider a network of servers linked together by a graph topology. In other words, we use the results from the first part to show how to construct a reliable multi-server (or graph-based) FL architecture with privacy guarantees. The new setting is referred to as graph federated learning (GFL), and it is more robust to server breakdowns and communication failures. We examine two privacy schemes, one of them is similar to earlier approaches in the literature for multi-agent systems and relies on the addition of ad-hoc Laplacian noise over the edges, while the second approach relies on graph-homomorphic noise sources. We show how these schemes influence performance, with the second method keeping the MSE performance at expected levels while guaranteeing privacy.

Finally, in the third part of the dissertation, we consider a broader multi-agent setting, without a centralized processor. It consists of a decentralized architecture where all processing is localized and agents can only interact with their neighbours. We again devise a reliable privatization scheme for this more general setting, which is useful for decentralized optimization and learning strategies. The scheme again ensures differential privacy without degrading the expected MSE performance of the network.

Keywords: multi-agent system, distributed learning, decentralized learning, federated learning, diffusion learning, differential privacy.

Résumé

Un système multi-agents consiste en un ensemble d'agents décisionnels ou d'apprentissage soumis à des observations en continu d'un phénomène du monde réel. L'objectif du système est de résoudre un problème global d'apprentissage ou d'optimisation de manière distribuée ou décentralisée, les agents collectant des données localement et interagissant soit avec leurs voisins, soit avec un processeur central.

Ces systèmes multi-agents sont prévalents dans de nombreuses applications du monde réel, telles que la conduite autonome, les systèmes multi-robots, les systèmes multi-capteurs, la surveillance des cibles et la réponse aux catastrophes, pour n'en citer que quelques-unes. Les solutions décentralisées et distribuées sont souvent imposées par la nature du système, par exemple, les données météorologiques sont naturellement distribuées sur différents sites géographiques, ou ces solutions sont préférables en raison de leur robustesse accrue aux défaillances des liens et des nœuds.

Lors de la construction de systèmes multi-agents, on formule normalement une fonction de risque globale, constituée de l'agrégat des risques locaux, puis on cherche à approcher son optimiseur par des interactions localisées entre agents voisins. Au cours de ce processus, les agents devront partager les données traitées et/ou les itérations avec leurs voisins. La question de la confidentialité devient alors critique pour permettre une communication sûre des informations sur les bords reliant les éléments du réseau multi-agents. Il existe plusieurs études dans la littérature qui renforcent la confidentialité en ajoutant des sources de bruit aléatoires en plus des informations partagées. La plupart de ces travaux établissent que les architectures résultantes sont différentiellement confidentiel, mais ils doivent supposer que les gradients des fonctions de risque sont bornés. Malheureusement, cette condition est rarement valable en pratique et ce fait est souvent négligé dans la plupart des études. Par exemple, même les fonctions de risque quadratiques ont des gradients non bornés car ces gradients seront des fonctions affines du paramètre inconnu. En outre, la plupart des études ne reconnaissent pas que leurs solutions différentiellement confidentielles et les sources de bruit ajoutées finissent par dégrader la performance quadratique moyenne des algorithmes d'apprentissage de $O(\mu)$ à $O(\mu^{-1})$, où μ est le petit paramètre d'apprentissage. Il s'agit d'une limitation sérieuse qui n'est toujours pas prise en compte dans les approches existantes de la confidentialité différentielle dans les systèmes multi-agents.

Dans cette thèse, nous résolvons ces deux problèmes. Premièrement, nous ne supposons pas de gradients bornés pour les fonctions de risque. Et pourtant, nous serons toujours capables d'établir que les systèmes multi-agents restent différentiellement confidentiels dans le sens de la haute probabilité. Nous parvenons à cette conclusion en montrant que les sources de bruit ne doivent pas être ajoutées de façon *ad hoc*, comme c'est souvent le cas dans les approches existantes, mais qu'elles doivent être construites en tenant compte de la topologie du graphe. Sinon, les bruits finissent par générer un effet amplifiant qui dégrade la performance. Pour cette raison, nous introduisons une construction de bruit homomorphique *localement* et montrons que, sous ce processus, la performance de l'erreur quadratique moyenne du système multi-agent restera à $O(\mu)$ tout en étant différentiellement confidentielle en même temps. Il s'agit d'une conclusion rassurante. Nous illustrons ces résultats pour le cas particulier des architectures d'apprentissage fédérées, mais nous les étendons également à des stratégies d'apprentissage et d'optimisation distribuées plus générales sur des architectures décentralisées.

Motivée par ces considérations, la première partie de la thèse étudie un système multi-agent distribué particulier, connu sous le nom d'apprentissage fédéré (FL). Le cadre fédéré consiste en un serveur central qui orchestre le processus d'apprentissage parmi une collection de dispositifs périphériques. Nous appelons cette configuration une architecture distribuée (par opposition à l'architecture décentralisée sans traitement central, étudiée dans la troisième partie de la thèse). Certains cas d'utilisation de la technologie peuvent être trouvés dans l'industrie des soins de santé, le secteur des assurances, les applications IoT, et d'autres technologies telles que le texte prédictif ou la reconnaissance vocale. Nous examinons quelques questions liées à la performance de l'algorithme FL. D'abord, nous établissons sa convergence sous trois caractéristiques exigeantes liées à l'hétérogénéité des données, au fonctionnement asynchrone et à la participation partielle des agents. Ensuite, nous montrons comment la performance peut être améliorée en introduisant un mécanisme basé sur l'échantillonnage par importance pour choisir les agents participants et les échantillons de données à chaque étape du processus d'apprentissage de manière optimisée. Enfin, nous introduisons une couche de privatisation et expliquons comment sa mise en œuvre ne dégrade pas les performances.

La deuxième partie de la thèse introduit une architecture plus réaliste pour le cadre fédéré. Au lieu de supposer une structure à serveur unique, nous considérons maintenant un réseau de serveurs reliés entre eux par une topologie de graphe. En d'autres termes, nous utilisons les résultats de la première partie pour montrer comment construire une architecture FL fiable multi-serveur (ou basée sur un graphe) avec des garanties de confidentialité. Ce nouveau paramètre est appelé apprentissage fédéré par graphe (GFL), et il est plus robuste aux pannes de serveur et aux défaillances de communication. Nous examinons deux schémas de confidentialité, l'un d'entre eux est similaire aux approches précédentes dans la littérature pour les systèmes multi-agents et repose sur l'ajout de bruit Laplacien ad-hoc sur les bords, tandis que la seconde approche repose sur des sources de bruit homomorphiques au graphe. Nous montrons comment ces schémas influencent les performances, la seconde méthode maintenant les performances MSE aux niveaux attendus tout en garantissant la confidentialité.

Enfin, dans la troisième partie de la thèse, nous considérons un cadre multi-agent plus large, sans processeur centralisé. Il s'agit d'une architecture décentralisée où tous les traitements sont localisés et où les agents ne peuvent interagir qu'avec leurs voisins. Nous concevons à nouveau un schéma de privatisation fiable pour ce cadre plus général, qui est utile pour les stratégies d'optimisation et d'apprentissage décentralisées. Le schéma garantit à nouveau la confidentialité différentielle sans dégrader la performance MSE attendue du réseau.

Mots-clés : système multi-agent, apprentisage distribué, apprentisage decentralizé, apprentisage fédéré, apprentissage par diffusion, confidentialité différentielle.

Contents

1 Introduction

From the study of simulated life [1], to target surveillance [2,3], and digital health [4,5], a system of multiple cognitive agents interacting together is at the core of such examples. We refer to these configurations as multi-agent systems, and they are used in a wide range of applications [6–9]. These structures do not rely on a single intelligent agent to perform the task of interest and are, therefore, more resilient to failure and also more scalable [10]. One important problem setting for multi-agent systems is the concept of decentralized learning. It is useful to solve machine learning problems of various types, such as classification, detection, segmentation, and inference. Examples include recommender systems to chatbots and medical image processing.

In what follows, we introduce some key concepts that will be called upon in the future chapters. In particular, we explain the difference between various multi-agent systems: centralized, distributed, and decentralized. We also formulate a typical global empirical risk minimization problem and describe algorithms for its solution based on stochastic gradient descent, mini-batch, and importance sampling implementations. Moreover, since agents need to share information among themselves in a distributed setting, the question of privacy becomes critical. We therefore review briefly the concept of differential privacy in preparation for its application in future chapters.

1.1 Multi-agent Systems

Multi-agent systems consist of intelligent agents that interact to solve problems that surpass individual capabilities. We distinguish among three popular architectures.

We start with the centralized architecture, which consists of a central processor that either aggregates all the data in one location or at multiple locations (Figure 1.1 *left*). The central processor is responsible for running the learning algorithm and for processing all the data. Examples of such systems include social media platforms like Twitter or Facebook where the companies control the entire system and its data. In domains where a global view from a single processor is present and the distribution of the task is impossible, it becomes more sensical to adopt such a viewpoint. These systems are less complex and easier to control. However, they are more vulnerable to security attacks, bottlenecks, and system breakdowns.

In comparison, a distributed system consists of one server that is connected to multiple agents with their own local data (Figure 1.1 *middle*). The central server works as the orchestrator of the whole system, while the agents act as data processors and learners. An example of such system is federated learning [11], which can be used for digital health applications [5], smart homes/cities [12], and text prediction [13], among other applications. The advantage of such systems is that we no longer require large bandwidth for data transferring, and processing is moved towards the local agents. Furthermore, these systems are more agile and more resilient than fully centralized architectures. However, they continue to rely on one central entity for the organization. Thus, they remain sensitive to breakdowns and failures.

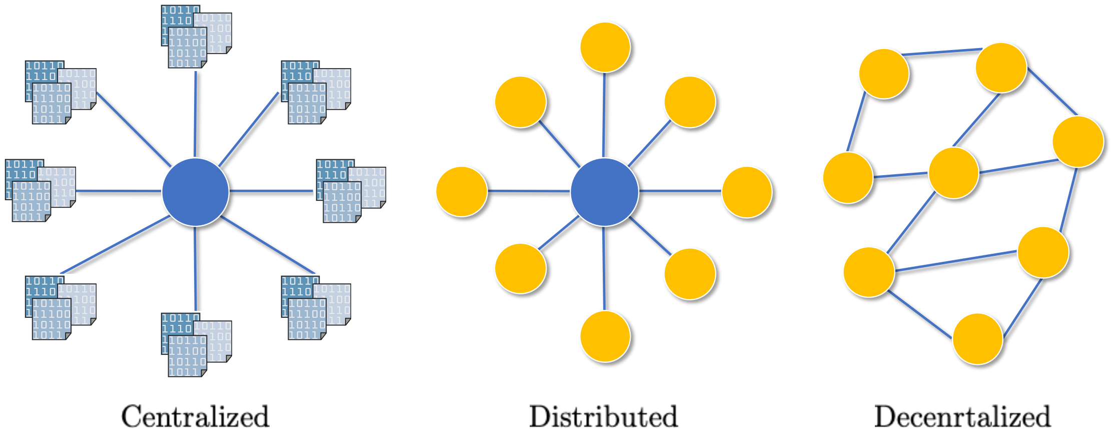


Figure 1.1 – An illustration of multi-agent systems.

A fully distributed architecture is what we refer to as a decentralized system. It drops the central server and transfers all work to the agents (Figure 1.1 *right*). The agents are connected by a graph topology and they work together as equals. Such systems can be found in swarm learning [14], social networks [15], and blockchains [16], for example. Compared to centralized and distributed systems, decentralized architectures are more tolerent to edge or node failures. They are also more resilient and robust, and can match the performance of fully centralized solutions. However, decentralized systems are more vulnerable to attacks from malicious agents and need more care with privacy.

We will associate a combination matrix A with every decentralized architecture. We denote the elements of the matrix by a_{mp}, which is the weight attributed by agent p to information arriving from agent m. We consider stochastic matrices; either left-stochastic satisfying:

$$\mathbb{1}^{\mathsf{T}} A = \mathbb{1}, \quad a_{mp} \geq 0, \tag{1.1}$$

or doubly stochastic satisfying:

$$\mathbb{1}^{\mathsf{T}} A = \mathbb{1}, \quad A\mathbb{1} = \mathbb{1}, \quad a_{mp} \geq 0. \tag{1.2}$$

We will also assume that the graph is strongly connected, which guarantees that the combination matrix A is primitive. This fact implies from the Peron-Frobenius theorem [9, 17] that A has a single eigenvalue at one with all other eigenvalues inside the unit circle. We refer to the eigenvector corresponding to the eigenvalue at one as the Perron eigenvector. It satisfies the following properties:

$$Aq = q, \quad q_p > 0, \quad \mathbb{1}^{\mathsf{T}} q = 1. \tag{1.3}$$

For a doubly stochastic matrix of size $P \times P$, the Perron eigenvector is given by:

$$q = \frac{1}{P}\mathbb{1}. \tag{1.4}$$

1.2 Empirical Risk Minimization

In this dissertation we study multi-agent systems that focus on learning. These learning algorithms generally result from optimization problems seeking some model $w^\star$ that fits the data distribution. We motivate the setting by focusing initially on single-agent learning. We consider strongly-convex optimization problems, where the optimizer $w^\star$ is unique. The optimal model is defined as the minimizer of the expectation of some convex loss function $Q(w; \boldsymbol{x})$ over the distribution of the random data $\boldsymbol{x}$:

$$w^\star = \underset{w}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}} Q(w; \boldsymbol{x}). \tag{1.5}$$

The main issue with the above formulation is that the distribution of the data is usually unknown beforehand, and thus the stochastic risk is unknown in closed form. Instead, samples x_n of the data are accessible, and thus the optimization problem is usually reformulated into an empirical risk minimization:

$$w^o = \underset{w}{\operatorname{argmin}} \left\{ J(w) \triangleq \frac{1}{N} \sum_{n=1}^{N} Q(w; x_n) \right\}, \tag{1.6}$$

where $J(w)$ denotes the risk function defined as a sample average over the data. The minimizer of the empirical risk, now denoted by w^o, is dependent on the sample data $\{x_n\}_{n=1}^{N}$. Therefore, if the available dataset is not descriptive of the data distribution, then the optimal model w^o is not general enough. This is what is referred to as generalization error. Thus, w^o is as good as the available dataset. It can be shown that w^o is close enough to $w^\star$ under the assumption of ergodicity and for large enough sample size N [17–21].

1.3 Single-agent Learning

There exist various algorithms to learn (i.e., approximate) w^o. The classical gradient descent (GD) algorithm is given by [17, 22]:

$$w_i = w_{i-1} - \mu \nabla_{w^\mathsf{T}} J(w_{i-1}), \tag{1.7}$$

with i indicating the iteration index and μ the step-size. With enough time, and under some assumptions on the risk and loss functions, GD reaches w^o with zero error. However, the main problem with the GD algorithm is the calculation of the true gradient of $J(w)$. In the case of streaming data, we would need to wait for all data points to become available in order to calculate the gradient during each iteration. Furthemore, the calculation of the gradient may be costly when the number of samples is large.

Therefore, one solution is to approximate the gradient by using:

$$\widehat{\nabla_{w^\mathsf{T}} J(\cdot)} = \nabla_{w^\mathsf{T}} Q(\cdot; \boldsymbol{x}_{b_i}), \tag{1.8}$$

for some randomly chosen sample $b \in \{1, 2, \cdots, N\}$. The resulting stochastic gradient (SG) algorithm is given by:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \nabla_{w^\mathsf{T}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_{b_i}). \tag{1.9}$$

The iterate $\boldsymbol{w}_i$ is now denoted in boldface to indicate the randomness introduced from sampling the data point $\boldsymbol{x}_{b_i}$. Due to this randomness, the performance of the algorithm is affected. It is known that, on average, SG results in a final model whose mean-square error (MSE) is $O(\mu)$ away from the minimizer w^o [17]. That is:

$$\limsup_{i\to\infty} \ \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 \ = \ O(\mu) \tag{1.10}$$

To improve the margin of error, we can approximate the gradient by using a mini-batch:

$$\widehat{\nabla_{w^\mathsf{T}} J(\cdot)} = \frac{1}{B} \sum_{b \in \mathcal{B}_i} \nabla_{w^\mathsf{T}} Q(\cdot; \boldsymbol{x}_b), \tag{1.11}$$

where $\mathcal{B}_i$ is the set of sampled data points at iteration i. The algorithm will then become:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{B} \sum_{b \in \mathcal{B}_i} \nabla_{w^\mathsf{T}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_b). \tag{1.12}$$

The sampling of the mini-batch can be done with or without replacement. Both methods reduce the error by some factor τ_B, i.e., mini-batch SG reaches a final model whose MSE is in the neighbourhood of $O(\mu)/\tau_B$ from w^o. This factor is given by [17]:

$$\tau_B = \begin{cases} B, & \text{with replacement} \\ B\frac{N-1}{N-B}. & \text{without replacement} \end{cases} \tag{1.13}$$

Thus, given a choice, it is preferable to sample the data points without replacement.

Even more generally, we can assign probabilities to the training samples in order to further reduce the error. This procedure is justified when some data samples happen to be more relevant to the learning process. For example, a data sample that has a large gradient makes the algorithm take larger steps away from the previous iterate. Thus, assigning higher probabilities to such samples will result in them being sampled more often. Thus, if we let p_n be the sampling probability of sample n, then the gradient is approximated by:

$$\widehat{\nabla_{w^\mathsf{T}} J}(\cdot) = \frac{1}{B} \sum_{b \in \mathcal{B}_i} \frac{1}{Np_b} \nabla_{w^\mathsf{T}} Q(\cdot; \boldsymbol{x}_b), \tag{1.14}$$

and the new algorithm becomes:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{B} \sum_{b \in \mathcal{B}_i} \frac{1}{Np_b} \nabla_{w^\mathsf{T}} Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_b). \tag{1.15}$$

We now add the factor $1/(Np_b)$ to ensure that the stochastic gradient remains an unbiased estimate of the true gradient, i.e.:

$$\mathbb{E}\widehat{\nabla_{w^\mathsf{T}} J}(\cdot) = \nabla_{w^\mathsf{T}} J(\cdot). \tag{1.16}$$

We can retrieve the original mini-batch estimate if we were to set $p_b = 1/N$.

With the stochastic gradient (1.14), we can show for $B = 1$ that the variance of the gardient noise $\boldsymbol{s}_i$, defined as the difference between the true gradient and the stochastic gradient, is bounded as [17]:

$$\mathbb{E}\|\boldsymbol{s}_i\|^2 \leq \beta_s^2 \mathbb{E}\|w^o - \boldsymbol{w}_{i-1}\|^2 + \sigma_s^2, \tag{1.17}$$

for some constant β_s^2 and where:

$$\sigma_s^2 \triangleq \frac{2}{N^2}\sum_{n=1}^{N}\frac{1}{p_n}\|\nabla_{w^\mathsf{T}}Q(w^o;x_n)\|^2. \tag{1.18}$$

Thus, knowing that σ_s^2 appears in the bound of the mean-square error (MSE), we can choose the probabilities p_n to minimize the error variance. In particular, by setting the sampling probabilities to:

$$p_n^o = \frac{\|\nabla_{w^\mathsf{T}}Q(w^o;x_n)\|}{\sum\limits_{b=1}^{N}\|\nabla_{w^\mathsf{T}}Q(w^o;x_b)\|}, \tag{1.19}$$

we reduce the bound on the MSE. However, this expression is a function of the minimizer w^o. One remedy is to replace it by the iterates $\boldsymbol{w}_{i-1}$ to get:

$$\widehat{p}_n^o = \frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{i-1};x_n)\|}{\sum\limits_{b=1}^{N}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{i-1};x_b)\|}. \tag{1.20}$$

The above expression is still inefficient due to the calculation in the denominator. We can replace the denominator by a mini-batch approximation, and only update the probabilities of the agents that were sampled, i.e., for $n \in \mathcal{B}_i$ the approximate optimal probability becomes:

$$\widehat{p}_n^o = \frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{i-1};x_n)\|}{\sum\limits_{b\in\mathcal{B}_i}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{i-1};x_b)\|}\left(1-\sum_{b\in\mathcal{B}_i^c}\widehat{p}_b^o\right), \tag{1.21}$$

where the multiplicative factor ensures the probabilities add up to one. We set the initial probabilities to the uniform case, $\widehat{p}_n^o = 1/N$.

For example, assume we have a total of five samples ($N = 5$), and during each iteration $B = 2$ samples are chosen. At time $i = 1$, we start with the initial probabilities set to $\widehat{p}_n^o = 1/5$. Say, samples one and two are chosen $\mathcal{B}_1 = \{1, 2\}$, then the new probability for sample one is updated as follows:

$$\widehat{p}_1^o = \frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_1)\|}{\sum\limits_{b=1,2}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_b)\|}\left(1-\sum_{b=3}^{5}\widehat{p}_b^o\right) = \frac{2}{5}\frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_1)\|}{\sum\limits_{b=1,2}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_b)\|}, \tag{1.22}$$

and a similar expression is found for the second sample. We can verify that the probabilities add up to 1:

$$\begin{aligned}\sum_{n=1}^{5}\widehat{p}_n^o &= \frac{2}{5}\frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_1)\|}{\sum\limits_{b=1,2}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{i-1};x_b)\|}+\frac{2}{5}\frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_2)\|}{\sum\limits_{b=1,2}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_b)\|}+\sum_{b=3}^{5}\frac{1}{5}\\ &= \frac{2}{5}+\frac{3}{5}=1. \end{aligned} \tag{1.23}$$

If during the next iteration samples three and four are chosen instead, then their probabilities will be updated, for $n = 3, 4$:

$$\widehat{p}_n^o = \frac{2}{5}\frac{\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_n)\|}{\sum\limits_{b=3,4}\|\nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_0;x_b)\|}. \tag{1.24}$$

1.4 Multi-agent Learning

We now motivate learning algorithms for multi-agent systems.

To begin with, the centralized system is, in fact, equivalent to the single agent case. Thus, the above-listed algorithms can be directly applied.

As for decentralized and distributed systems, we first reformulate the empirical risk minimization problem. If we assume there are K agents equipped with a local dataset $\{x_{k,n}\}_{n=1}^{N_k}$, then the overall risk function is the average of the local risks:

$$w^o = \underset{w}{\operatorname{argmin}}\,\frac{1}{K}\sum_{k=1}^{K}\left\{J_k(w)\triangleq\frac{1}{N_k}\sum_{n=1}^{N_k}Q(w;x_{k,n})\right\}. \tag{1.25}$$

One algorithm for the distributed system is federated averaging (FedAvg) [11]. It assumes during each iteration, that the server possesses a past model $\boldsymbol{w}_{i-1}$. It then samples a subset L of the K agents, which we denote by $\mathcal{L}_i$, to participate in this round. Each agent $k \in \mathcal{L}_i$ is sent the past model $\boldsymbol{w}_{i-1}$ and runs a total of E_k local update steps called epochs to get a new model $\boldsymbol{w}_{k,E_k}$. At the end of the local update steps, the agents share their final model with the server to be aggregated. The new updated model $\boldsymbol{w}_i$ is the average of all the final local models $\boldsymbol{w}_{k,E_k}$. Thus, if we denote by $\boldsymbol{w}_{k,e}$ the local model of agent k at epoch e, and we let $\boldsymbol{w}_{k,0} = \boldsymbol{w}_{i-1}$, then the local update steps are given by:

$$\boldsymbol{w}_{k,e} = \boldsymbol{w}_{k,e-1} - \mu\widehat{\nabla_{w^\mathsf{T}}J}_k(\boldsymbol{w}_{k,e-1}), \tag{1.26}$$

and the aggregation step at the server is defined as:

$$\boldsymbol{w}_i = \frac{1}{L}\sum_{k\in\mathcal{L}_i}\boldsymbol{w}_{k,E_k}. \tag{1.27}$$

The above two steps can then be written more compactly as:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{L} \sum_{k \in \mathcal{L}_i} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}). \tag{1.28}$$

This description resembles the centralized solution with a stochastic gradient defined as:

$$\widehat{\nabla_{w^\mathsf{T}} J}(\cdot) = \frac{1}{L} \sum_{k \in \mathcal{L}_i} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\cdot), \tag{1.29}$$

and evaluated at different models. The main problem with such a derivation is that even though the stochastic gradient is evaluated at the previous global model $\boldsymbol{w}_{i-1}$, it is not an unbiased estimate of the true gradient. We observe that it suffices to divide the local stochastic gradients $\widehat{\nabla_{w^\mathsf{T}} J_k}(\cdot)$ by the total epoch E_k. Thus in the remainder of this dissertation, we will use the following federated learning algorithm description; the new local step is given by:

$$\boldsymbol{w}_{k,e} = \boldsymbol{w}_{k,e-1} - \mu \frac{1}{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}), \tag{1.30}$$

and the centralized description is now:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{L} \sum_{k \in \mathcal{L}_i} \frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}). \tag{1.31}$$

We move on to algorithms for decentralized networks. The goal is still to solve the optimization problem (1.25). The main algorithms can be split into two steps: an adaptation step and a communication step. The adaptation step consists of updating the local model, while the communication step involves the sharing of information between neighbours. The different algorithms differ in the order of the two steps and the nature of the updates.

We first start with the consensus strategy [17, 23, 24]:

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1}, \tag{1.32}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,i-1}), \tag{1.33}$$

which consists of a communication step followed by an adaptation step. The neighbours first communicate their past models, and then update the aggregate of the past models. The adaptation step is *unbalanced* since it evluates the gradient at a different model $\boldsymbol{w}_{k,i-1}$ than the model that is being updated, $\boldsymbol{\psi}_{k,i-1}$. This issue has been shown to cause problems with the convergence of the algorithm [25].

Next, we present two diffusion algorithms, combine-then-adapt (CTA) diffusion and adapt-then-combine (ATC) diffusion [17, 25]. They solve the imbalance in consensus by making sure the gradients are evaluated at the same model that is being updated. In CTA diffusion, agents first share their past models and then perform the adaptation step:

$$\begin{aligned} \boldsymbol{\psi}_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{w}_{\ell,i-1}, & (1.34) \\ \boldsymbol{w}_{k,i} &= \boldsymbol{\psi}_{k,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{\psi}_{k,i-1}). & (1.35) \end{aligned}$$

ATC diffusion switches the steps; first comes adapatation and then the sharing of models:

$$\begin{aligned} \boldsymbol{\psi}_{k,i-1} &= \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,i-1}), & (1.36) \\ \boldsymbol{w}_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i-1}. & (1.37) \end{aligned}$$

Both diffusion algorithms overcome the convergence issues faced by the consensus algorithm.

A more general description of distributed algorithms can be derived that encapsulates the three previously described algorithms.The agents start by aggregating the past models from their neighbours. They then perform a local update step. They finally aggregate the updated models from their neighbours. Thus, we consider three combination matrices A_0, A_1, A_2, and we write the algorithm as:

$$\begin{aligned} \boldsymbol{\phi}_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k} \boldsymbol{w}_{\ell,i-1}, & (1.38) \\ \boldsymbol{\psi}_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{0,\ell k} \boldsymbol{\phi}_{\ell,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{\phi}_{k,i-1}), & (1.39) \\ \boldsymbol{w}_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k} \boldsymbol{\psi}_{\ell,i}. & (1.40) \end{aligned}$$

By setting $A_0 = A$ and the rest to identity, the algorithm reduces to consensus. By choosing $A_1 = A$ or $A_2 = A$, with the other matrices set to identity, we get the CTA diffusion and ATC diffusion, respectively.

1.5 Privacy in Multi-agent Systems

In multi-agent systems where data is distributed amongst different agents, it is important to safeguard sensitive information whether from the central server, or the neighbours.

Privatization schemes generally fall under two umbrellas. The first set of mechanisms are referred to as cryptographic methods, and examples of these methods include secure multiparty computation [26], homomorphic encryption [27], secure aggregation [28], and zero-knowledge proofs [29]. The second group of privatization schemes is what is referred to as differentially private methods. The key difference between these two paradigms is that the former relies on sophisticated deterministic functions that ensure information leakage is minimal yet still performs the required task, while the latter uses different masking schemes based on random noises that might alter the result of the performed task. For example, if we are calculating the average of the models generated by multiple agents, secure multiparty computation modifies the local models without altering the average. In comparison, a differentially private mechanism adds random noise to the models and modifies the computed average. Thus, cryptographic methods might be more complicated to implement, but they preserve the outcome. Yet, generating random noise is generally easier to implement, except they change the outcome.

Since in this dissertation, we focus on differentially private methods, we describe them in more detail. To gain some intuition about the concept, we walk through an example. Assume Alice, Bob, and Charlie are trying to calculate their average income. No person is comfortable sharing this information. If Alice is adamant on not announcing her income, then she will alter it in a way that does not raise suspicions.

Assume Alice works in the banking sector, and the typical income of someone in her position is US$140,000. Then, if she claims she earns US$135,000 when in fact she earns US$150,000, neither Bob nor Charlie would question her answer. However, if Alice claims she earns US$100,000, then that would give reason for Bob and Charlie to be suspicious. Therefore, Alice would need to choose a probable income for someone in her position, yet far enough from her actual income such that her answer does not give too much information away.

Differential privacy tries to abstract the above concept into a measure of privacy. Let w_a denote the true income of Alice, and let w'_a be the answer she reveals. She will not reveal her answer without privatizing it first using some privatization mechanism. We would say that the mechanism adopted by Alice to share her income is differentially private if the outcome of the mechanism when w'_a is used is close in distribution to when w_a were used. This translates into the following condition:

$$\mathbb{P}(\boldsymbol{M}(w_a) \in \mathcal{O}) \leq e^{\epsilon} \times \mathbb{P}(\boldsymbol{M}(w'_a) \in \mathcal{O}), \tag{1.41}$$

where $\boldsymbol{M}(\cdot)$ denotes the privatization mechanism (which replaces w by $\boldsymbol{M}(w)$), $\mathcal{O}$ is a subset of possible outcomes under this mechanism, and ϵ is some positive constant. Thus, the smaller the value of ϵ is, the closer the distributions are to each other, and the harder it is to guess if Alice shared w_a or w'_a.

One commonly used scheme for differential privacy is the Laplace mechanism. It consists of perturbing the message with a Laplacian noise, $\boldsymbol{g} \sim \text{Lap}(0, \sigma_g)$. Alice, Bob, and Charlie would share their income masked by some Laplacian noise. To show that this mechanism is indeed $\epsilon-$differentially private, it is enough to check condition (1.41). If

$\boldsymbol{g}_a$ is the Laplacian noise added to Alice's message, then the density function is given by:

$$f(w_a + \boldsymbol{g}_a) = \frac{\sqrt{2}}{2\sigma_g} \exp\left(-\frac{\sqrt{2}}{\sigma_g} |w_a + \boldsymbol{g}_a| \right), \tag{1.42}$$

and the condition (1.41) for a continuous random variable is given by:

$$\begin{aligned} \frac{f(w_a + \boldsymbol{g}_a)}{f(w'_a + \boldsymbol{g}_a)} &= \exp\left(\frac{\sqrt{2}}{\sigma_g} \left(|w'_a + \boldsymbol{g}_a| - |w_a + \boldsymbol{g}_a| \right) \right) \\ &\leq \exp\left(\frac{\sqrt{2}}{\sigma_g} |w'_a - w_a| \right). \end{aligned} \tag{1.43}$$

Thus, the Laplace mechanism is $\epsilon-$differentially private with:

$$\epsilon = \frac{\sqrt{2}}{\sigma_g} |w'_a - w_a|. \tag{1.44}$$

The larger the noise variance is, the harder it is for an attacker to extract private information. However, the worse the output is. Thus, if in the previous example the true average income is US$130,000, and each of Alice, Bob, and Charlie mask their true income with a Laplacian noise with high variance such that the new average income is now US$155,000 (see Table 1.1), this new average is misleading and not representative of the actual income distribution. Therefore, the choice of the noise variance is crucial to ensure privacy while still maintaining the integrity of the process.

Table 1.1 – Example of the average income among three people.

	Alice	Bob	Charlie	**Average**
Income	US$150,000	US$90,000	US$120,000	**US$120,000**
Perturbed Income	US$200,000	US$160,000	US$105,000	**US$155,000**

In the context of learning algorithms, agents would add some random noise to the shared models. Perturbing the models will affect the ability of the algorithm to estimate the optimal model. Thus, in differential privacy, a utility-privacy trade-off exists, and it is up to the designer to construct a masking scheme that ensures privacy without rendering the learned model useless.

1.6 Organization and Main Contribution

In this dissertation, we tackle the issue of privacy in multi-agent systems. We resolve two limitations that have yet to be addressed in the literature. Most studies assume the gradients of the risk and loss functions are bounded, which is rarely the case. Instead, we avoid this condition and are still able to devise a scheme that is differentially private with high probability. Additionally, the designed privatization schemes severly hinder performance by increasing the bound on the MSE from $O(\mu)$ to $O(\mu^{-1})$. By introducing a local graph-homomorphic scheme for decentralized systems, and by using a gradient perturbation method for distributed systems, we are able to maintain the bound on the MSE at $O(\mu)$.

The first chapter considers federated learning, a specific distributed system that consists of a central manager of the learning algorithm. We first establish convergence of federated averaging (FedAvg). Next, we improve performance by studying optimal sampling policies for selecting agents and their data. Usually, only uniform sampling schemes are used. However, in the first chapter, we examine the effect of importance sampling and devise schemes for sampling agents and data non-uniformly guided by a performance measure. Finally, we show that improvement in the differentially private federated learning algorithm can be attained through the addition of random noise to the updates, as opposed to the models.

The second chapter extends the federated structure into a network of federated units, which we call graph federated learning (GFL). The federated architecture is not robust and is sensitive to communication and computational overloads due to its one-master multi-client structure. We investigate two noise generation schemes. The first is the commonly used method that generates ad-hoc noise; while the second, graph-homorphic process, generates dependent noise that is well tuned to the graph structure. We show that the latter method improves the effect of the added noise on the MSE performance, as opposed to the former method.

Finally, the third chapter considers a network of agents and drops the central controlling unit. We study the privatization of decentralized learning and optimization strategies. We exploit an alternative graph-homorphic construction and show that it improves performance while guaranteeing privacy.

2 Federated Learning under Importance Sampling

In this chapter, we study a particular multi-agent system, known as federated learning (FL). It consists of a central server that coordinates the learning process among a collection of edge devices. We refer to this set-up as a distributed architecture (as opposed to the decentralized architecture without central processing, studied in the third part of the dissertation). Some use cases of the technology can be found in the healthcare industry, the insurance sector, IoT applications, or other technologies such as predictive text or voice recognition. We examine a couple of questions related to the performance of the FL algorithm. First, we establish its convergence under three demanding characteristics related to data heterogeneity, asynchronous operation, and partial agent participation. Then, we show that its performance can be improved by introducing a mechanism that is based on importance sampling to choose the participating agents and the data samples during each stage of the learning algorithm. Finally, we introduce a privatization layer and show that it enforces differential privacy without significant degradation to performance. The material in this chapter is based on the work in [30].

2.1 Introduction

We focus on algorithms that fall into the broad class of stochastic gradient (SG) methods. We consider a collection of K heterogeneous agents that may have different computational powers. Each agent k has locally N_k data points, which we denote by $\{x_{k,n}\}$; the subscript k refers to the agent, while the subscript n denotes the sample index within agent k's dataset. The goal of the agents is to find an optimizer for the aggregate risk function:

$$w^o \overset{\Delta}{=} \underset{w\in\mathbb{R}^M}{\text{argmin}} \frac{1}{K}\sum_{k=1}^{K} J_k(w), \tag{2.1}$$

where each $J_k(\cdot)$ is an empirical risk defined in terms of a loss function $Q_k(\cdot;\cdot)$:

$$J_k(w) \overset{\Delta}{=} \frac{1}{N_k}\sum_{n=1}^{N_k} Q_k(w;x_{k,n}). \tag{2.2}$$

Multiple strategies exist for solving such problems. They can be categorized into two main classes: a) partially decentralized strategies, which include a central processor with access to all data and which controls the distribution of the data into the nodes for processing [25,31–33]; and b) fully decentralized strategies, which consist of multiple agents connected by a graph topology and operating locally without oversight by a central processor [9,34–36]. Federated learning [11,28,37–47] offers a midterm solution, which consists of several agents collecting and processing local data that are then aggregated at the central processor.

When implementing SG, most strategies choose the samples from the training data according to a uniform distribution. In this chapter, we will consider more general *non-uniform* sampling schemes, where the agents are sampled according to some distribution π_k and the local data at agent k are in turn sampled according to some other distribution $\pi_n^{(k)}$. The index k in π_k and $\pi_n^{(k)}$ refers to the agents, while the subscript n in $\pi_n^{(k)}$ refers to the data. In this setting, the central processor selects the subset of agents for processing according to π_k and, once selected, an agent k will sample its data according to $\pi_n^{(k)}$. The importance sampling process in this chapter therefore involves two layers. We use the superscript (k) to denote the sampling distribution of the data at agent k. The sampling distributions $\{\pi_k, \pi_n^{(k)}\}$ are not fixed; instead we will show how to *adapt* them in order to enhance performance. At the same time, we will provide a detailed convergence analysis and establish performance and privacy limits.

2.1.1 Related Work

Several works studied the convergence of the federated learning algorithm under differing assumptions. These assumptions usually relate to the nature of the data (iid or non-iid), nature of the risk function (convex or non-convex), agent participation (full or partial), and operation (synchronous or asynchronous) [37,38,48–58]. Moreover, the data at different agents may not be sampled from the same distribution; this may occur in personalization problems, like recommender systems. One main difference between FL and other distributed paradigms is that FL does not require each agent to participate during each iteration of the learning process. This is what is referred to as partial agent participation. The second main difference is that participating agents during any given iteration would run multiple local steps, whose number may differ among the agents. In other words, while one agent may be able to run 5 local steps before it is required to share its final model with the server, another agent may only be able to run 2 local steps. Other works examine the convergence behavior of variations of the traditional FL algorithm, such as FedProx [37], hierarchical version of federated averaging (FedAvg) [58], multi-task federated learning [38], and dynamic FedAvg [59] — see Table 2.1.

By contrast, not much work has been done on selection schemes for agents and data in federated learning. Given the architecture of a federated learning solution, this is a natural and important question to consider. The existing works in client scheduling can be split into two categories: those seeking better performance, and those seeking fairness. Of the works pertaining to the first category, reference [60] develops a new client selection scheme, called FedCS, where the goal of the central server is to choose as many agents as possible that can complete an iteration by a required deadline, after acquiring information about the agents' resources. Reference [61] builds on this previous work to deal with non-IID data, and allows the server to collect some of the data from the agents and participate in the training of the model. The authors of [62] consider non-uniform sampling of agents and suggest approximate sampling probabilities that maximize the average inner product of the local gradient with the global gradient. To increase the convergence rate, [63] introduces a multi-armed bandit online client scheduling scheme in federated learning. References [64, 65] formulate an optimization problem over resource allocation and agent selection to improve training loss and energy transmission. References [66–68] fall under the second category; in agnostic federated learning [66], the data distribution is assumed to be a mixture of the local distributions, and a minimax problem for agent selection is solved. Reference [67] generalizes the previous work by reweighting the cost function and assigning higher weights to agents with higher loss. The work in [68] study different agent selection schemes, random scheduling, round robin, and proportional fairness, in a wireless federated learning network under a limited bandwidth constraint; while the authors in [69] propose a selection scheme that takes into account the staleness of the received parameters and the instantaneous channel qualities. The work in [70] proposes a budget dividing scheme among different agents by maximizing the global utility and minizing the inequality among the agents.

While there exist works that study the effect of importance sampling in distributed learning [71–75], all of these works apply importance sampling to the data at each agent. To our knowledge, there are no works that examine the *combined effect* of two hierarchical layers of sampling: one for the nodes and another for their data. By introducing a two-layer importance sampling scheme to the federated learning paradigm, we can tackle the problem of importance sampling both in relation to agents *and* in relation to data.

The main difference between this work and those falling under agent scheduling is the reason behind the sampling mechanism, i.e., agents and data points are chosen with the purpose of minimizing the variance of the gradient noise. This work can also be compared to active learning. However, active learning is a semi-supervised learning scheme which aims at labelling unlabelled data.

Table 2.1 – List of references on the convergence analysis of federated learning under different assumptions. This chapter based on [30] along with our previous work in [59] are the only ones to tackle the 3 challenges of federated learning(non-iid data, asynchronous mode of operation, and partial agent participation).

References	**Algorithm**	**Function Type**	**Data Heterogeneity**	**Operation**	**Agent Participation**	**Other Assumptions**
[49]	dist. gradient descent	convex	***non-iid***	synchronous	full	smooth
[50]	dist. SGD	convex	iid	synchronous	full	smooth
[51, 52]	dist. SGD	non-convex	iid	synchronous	full	smooth
[53]	dist. SGD	non-convex	***non-iid*** iid	synchronous ***asynchronous***	full	-
[54]	dist. SGD	convex	***non-iid***	synchronous	full	bounded gradients
[55]	dist. momentum SGD	non-convex	***non-iid***	synchronous	full	-
[56]	FedAvg	convex some non-convex	***non-iid***	***asynchronous***	full	-
[57]	FedAvg	convex	***non-iid***	synchronous	***partial***	bounded gradients
[37]	FedProx	non-convex	***non-iid***	***asynchronous***	***partial***	-
[58]	HierFAVG	convex non-convex	***non-iid***	synchronous	full	-
[38]	MOCHA	convex	***non-iid***	synchronous	full	-
[59]	Dynamic FedAvg	convex	***non-iid***	***asynchronous***	***partial***	model drift
[30]	ISFedAvg	convex	***non-iid***	***asynchronous***	***partial***	importance sampling

2.1.2 Sampling and Inclusion Probabilities

Before describing the problem setting, we need to clarify the difference between two notions: (a) sampling probability and (b) inclusion probability. Consider the following illustrative example. Consider $N = 4$ balls of which we wish to choose $B = 2$ balls non-uniformly and without replacement. Let the sampling probabilities be $\pi_n = \{1/3, 1/6, 1/3, 1/6\}$. This means that, initially, balls 1 and 3 are twice as likely to be selected compared to balls 2 and 4. For the first trial, all the inclusion probabilities are equal to the sampling probabilities, i.e.:

$$\mathbb{P}(n \text{ chosen on } 1^{st} \text{ trial}) = \pi_n. \tag{2.3}$$

However, since we are sampling *without replacement*, the inclusion probabilities for the second trial depend on the outcome of the first trial, i.e.:

$$\mathbb{P}(n \text{ chosen on } 2^{nd} \text{ trial}|m \text{ chosen on } 1^{st} \text{ trial}) = \frac{\pi_n}{(1-\pi_m)}. \tag{2.4}$$

Using the sampling probabilities, we can evaluate the likelihood that each ball will end up belonging to the selected set of 2 balls. In particular, the probability that ball 1 is chosen either in the first or second trial is given by:

$$\begin{aligned}\mathbb{P}(1 \text{ chosen}) &= \sum_{n=2}^{4} \mathbb{P}\Big(1 \text{ chosen on } 1^{st} \text{ trial } \& \ n \text{ chosen on } 2^{nd} \text{ trial})\\ &\quad + \mathbb{P}\Big(n \text{ chosen on } 1^{st} \text{ trial } \& \ 1 \text{ chosen on } 2^{nd} \text{ trial})\\ &= \sum_{n=2}^{4}\left(\pi_1 \frac{\pi_n}{1-\pi_1} + \pi_n \frac{\pi_1}{1-\pi_n}\right). \end{aligned} \tag{2.5}$$

Thus, the sampling probability is the working probability. It is the probability used to actually choose the samples, while the inclusion probability is a descriptive probability that indicates the likelihood of a ball being included in the final selected subset. Observe that the inclusion probabilities depend on the sampling scheme, while the sampling probabilities do not. When considering *uniform sampling without replacement*, the inclusion probability is a multiple of the sampling probability. For example, sampling B numbers from $\{1, 2, \cdots, N\}$ with sampling probabilities $1/N$, the inclusion probability is found to be $\mathbb{P}(n \in \mathcal{B}) = B/N$. Note further that while the sampling probabilities sum to 1 over all the sampling space, the inclusion probabilities $\mathbb{P}(n \in \mathcal{B})$ sum to B. In our derivations, we will be relying frequently on the inclusion probabilities.

Next, we consider a total number of K agents. At each iteration i of the algorithm, a subset of agents $\mathcal{L}_i$ of size L is chosen randomly *without replacement.* We denote the probability that agent k is included in the sample by Lp_k [76], i.e.,

$$p_k \triangleq \frac{\mathbb{P}\Big(k \in \mathcal{L}_i\Big)}{L}. \tag{2.6}$$

In addition, each sampled agent k will run a mini-batch SGD by sampling B_k data points $\mathcal{B}_{k,i}$ *without replacement* from its local data. We denote the probability of inclusion of data point n by $B_k p_n^{(k)}$, i.e.,

$$p_n^{(k)} \triangleq \frac{\mathbb{P}\Big(n \in \mathcal{B}_{k,i}\Big)}{B_k}. \tag{2.7}$$

We refer to p_k and $p_n^{(k)}$ as the *normalized inclusion probabilities*. They sum to 1 over the sampling space; p_k sums to 1 over all agents and $p_n^{(k)}$ over the data at each agent.

2.2 Algorithm Derivation

The goal of the federated learning algorithm is to approximate the centralized solution w^o while dealing with the constraint of distributed data. The goal is achieved by using an unbiased estimate of the gradient of the risk function, $\frac{1}{K}\sum_{k=1}^{K} \nabla_{w^{\mathsf{T}}} J_k(w)$. As explained in [59] for the case of uniform sampling, if we assume each agent k runs E_k epochs per iteration i (with each epoch using B_k samples in $\mathcal{B}_{k,i,e}$), then we can construct an unbiased estimate for the true gradient by considering the following estimator:

$$\frac{1}{L}\sum_{k\in\mathcal{L}_i} \frac{1}{E_k B_k}\sum_{e=1}^{E_k}\sum_{b\in\mathcal{B}_{k,i,e}} \nabla_{w^{\mathsf{T}}} Q_k(w;\boldsymbol{x}_{k,b}), \tag{2.8}$$

as opposed to the original estimator from [11], where the main difference is the scaling by the epoch size E_k. This correction is important for the performance of the averaged model. Since the number of epochs E_k can be non-uniform across the agents, then, without correction, agents with large epoch sizes will bias the solution by driving it towards their local model and away from w^o.

Expression (2.8) is still not sufficient for our purposes in this chapter, since agents and data are allowed to be sampled *non-uniformly without replacement*. In this case, we need to adjust (2.8) by including the *inclusion probabilities* [71]. They are necessary to ensure the estimate is unbiased, as will later be seen in Lemma 2.1. The local estimate of the gradient at agent k becomes $\frac{1}{Kp_k}\widehat{\nabla_{w^{\mathsf{T}}} J_k}(w)$, with:

$$\widehat{\nabla_{w^{\mathsf{T}}} J_k}(w) \triangleq \frac{1}{E_k B_k}\sum_{e=1}^{E_k}\sum_{b\in\mathcal{B}_{k,i,e}} \frac{1}{N_k p_b^{(k)}} \nabla_{w^{\mathsf{T}}} Q_k(w;\boldsymbol{x}_{k,b}). \tag{2.9}$$

Motivated by (2.9), we can write down a stochastic gradient update at each agent k at epoch e, and at the central processor at iteration i:

$$\boldsymbol{w}_{k,e} = \boldsymbol{w}_{k,e-1} - \frac{\mu}{Kp_k E_k B_k} \sum_{b \in \mathcal{B}_{k,i,e}} \frac{1}{N_k p_b^{(k)}} \nabla_{w^\top} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,b}), \tag{2.10}$$

$$\boldsymbol{w}_i = \frac{1}{L} \sum_{k \in \mathcal{L}_i} \boldsymbol{w}_{k,E_k}, \tag{2.11}$$

where at each iteration i, step (2.10) is repeated for $e = 1, 2, \cdots, E_k$. We arrive at the Algorithm 2.1, which we refer to as Importance Sampling Federated Averaging (ISFedAvg).

Algorithm 2.1: (Importance Sampling Federated Averaging)

```
initialize $w_0$;
for each iteration $i = 1, 2, \cdots$ do
  Select the set of participating agents $\mathcal{L}_i$ by sampling $L$ times from $\{1, \ldots, K\}$
  without replacement according to the sampling probabilities $\pi_k$.
  for each agent $k \in \mathcal{L}_i$ do
    initialize $\boldsymbol{w}_{k,0} = \boldsymbol{w}_{i-1}$
    for each epoch $e = 1, 2, \cdots E_k$ do
      Find indices of the mini-batch sample $\mathcal{B}_{k,i,e}$ by sampling $B_k$ times from
      $\{1, \ldots, N_k\}$ without replacement according to the sampling probabilities $\pi_n^{(k)}$.
      $\boldsymbol{g} = \frac{1}{B_k} \sum_{b \in \mathcal{B}_{k,i,e}} \frac{1}{N_k p_b^{(k)}} \nabla_{w^\top} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,b})$
      $\boldsymbol{w}_{k,e} = \boldsymbol{w}_{k,e-1} - \mu \frac{1}{E_k K p_k} \boldsymbol{g}$
    end for
  end for
  $\boldsymbol{w}_i = \frac{1}{L} \sum_{k \in \mathcal{L}_i} \boldsymbol{w}_{k,E_k}$
end for
```

2.3 Convergence Analysis

2.3.1 Modeling Conditions

To facilitate the analysis of the algorithm, we list some common assumptions on the nature of the local risk functions and their respective minimizers. Specifically, we assume convex loss functions with Lipschitz continuous gradients.

Assumption 2.1 (**Convexity and smoothness**). *The functions $J_k(\cdot)$ are $\nu-$strongly convex, and $Q_k(\cdot;x_{k,n})$ are convex, namely:*

$$J_k(w_2) \geq J_k(w_1) + \nabla_{w^\intercal} J_k(w_1)(w_2 - w_1) + \frac{\nu}{2}\|w_2 - w_1\|^2, \tag{2.12}$$

$$Q_k(w_2;x_{k,n}) \geq Q_k(w_1;x_{k,n}) + \nabla_{w^\intercal} Q_k(w_1;x_{k,n})(w_2 - w_1). \tag{2.13}$$

Also, the functions $Q_k(\cdot;x_{k,n})$ have $\delta-$Lipschitz gradients:

$$\|\nabla_{w^\intercal} Q_k(w_2;x_{k,n}) - \nabla_{w^\intercal} Q_k(w_1;x_{k,n})\| \leq \delta\|w_2 - w_1\|. \tag{2.14}$$

We further assume that the individual minimizers:

$$w_k^o = \underset{w\in\mathbb{R}^M}{\text{argmin}}\, J_k(w), \tag{2.15}$$

do not drift too far away from w^o. Such an assumption is viable, since in the case when the local models varry significantly among the different agents, collaboration is non-sensical. A multi-task formulation would make more sense in that case. This assumption does not reduce the non-iid assumption in federated learning, but instead bounds its degree. If we consider the example of text prediction on mobile phones, users using different languages will have differing models, and collaboration amongst them is not optimal. However, users using the same language, still have non-iid data, but most probably, their local models are relatively close and collaboration is advocated.

Assumption 2.2 (**Model drifts**). *The distance of each local model w_k^o to the global model w^o is uniformly bounded over the data, $\|w_k^o - w^o\| \leq \xi$.*

2.3.2 Error Recursion

Iterating the local update (2.10) over multiple epochs and combining according to (2.11), we obtain the following update for the centralized iterate:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu\frac{1}{L}\sum_{k\in\mathcal{L}_i}\frac{1}{Kp_kE_kB_k}\sum_{e=1}^{E_k}\sum_{b\in\mathcal{B}_{k,i,e}}\frac{1}{N_kp_b^{(k)}}\nabla_{w^\intercal}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,b}). \tag{2.16}$$

To simplify the notation, we introduce the error terms:

$$\boldsymbol{s}_i \triangleq \frac{1}{L}\sum_{\ell\in\mathcal{L}_i}\frac{1}{Kp_\ell}\widehat{\nabla_{w^\mathsf{T}} J_\ell}(\boldsymbol{w}_{i-1}) - \frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}), \tag{2.17}$$

$$\boldsymbol{q}_i \triangleq \frac{1}{L}\sum_{\ell\in\mathcal{L}_i}\frac{1}{Kp_\ell E_\ell B_\ell}\sum_{e=1}^{E_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{1}{N_\ell p_b^{(\ell)}}\Big(\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{\ell,e-1};\boldsymbol{x}_{\ell,b}) - \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{\ell,b})\Big). \tag{2.18}$$

The first error term $\boldsymbol{s}_i$, which we call *gradient error*, captures the error from approximating the true gradient by using subsets of agents and data; while the second error term $\boldsymbol{q}_i$, which we call *incremental error*, captures the error resulting from the incremental implementation, where at each epoch during one iteration, the gradient is calculated at the local iterate $w_{k,e-1}$. Note that this second error evaluates the loss function at the local and global iterates. As we will show later, the incremental error will play a minor role, and the dominant factor will be the gradient error. Before establishing the main result in Theorem 2.1 on the convergence of ISFedAvg algorithm, we present preliminary results that will lead to it. Thus, to show the convergence of the algorithm, we must assure the gradient noise $\boldsymbol{s}_i$ has zero mean and bounded variance, and the incremental noise $\boldsymbol{q}_i$ has bounded variance. Furthermore, since we split the noise due to the stochastic gradient into incremental and gradient noise, we can split the analysis into that of the centralized steps and the local epochs. By proving that both the centralized and local steps converge, we show the global algorithm converges too.

Replacing the two error terms (2.17) and (2.18) into recursion (2.16) and subtracting w^o from both sides of the equation, we get the following error recursion:

$$\widetilde{\boldsymbol{w}}_i = \widetilde{\boldsymbol{w}}_{i-1} + \mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) + \mu\boldsymbol{s}_i + \mu\boldsymbol{q}_i, \tag{2.19}$$

where $\widetilde{\boldsymbol{w}}_i = w^o - \boldsymbol{w}_i$. To bound the ℓ_2-norm of the error, we split it into two terms, centralized and incremental, using Jensen's inequality with some constant $\alpha \in (0,1)$ to be defined later:

$$\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \frac{1}{\alpha}\left\|\widetilde{\boldsymbol{w}}_{i-1} + \mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) + \mu\boldsymbol{s}_i\right\|^2 + \frac{1}{1-\alpha}\mu^2\|\boldsymbol{q}_i\|^2. \tag{2.20}$$

We start with the first term that represents the centralized solution. We need to show that it converges. To do so, we start with the gradient noise, and we establish in Lemma 2.1 that it remains bounded. We bound the gradient noise $\boldsymbol{s}_i$ under two constructions: sampling with replacement, and sampling without replacement.

Lemma 2.1 (**Estimation of moments of the gradient noise**). *The gradient noise defined in* (2.17) *has zero mean* $\mathbb{E}\{\boldsymbol{s}_i|\boldsymbol{w}_{i-1}\} = 0$, *with bounded variance, regardless of the sampling scheme. More specifically, sampling agents and data with replacement, results in the following bound:*

$$\mathbb{E}\{\|\boldsymbol{s}_i\|^2|\boldsymbol{w}_{i-1}\} \leq \beta_s^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_s^2, \tag{2.21}$$

where $\widetilde{\boldsymbol{w}}_{i-1} = w^o - \boldsymbol{w}_{i-1}$ *and the constants:*

$$\beta_s^2 \triangleq \frac{3\delta^2}{L} + \frac{1}{LK^2}\sum_{k=1}^{K}\frac{1}{p_k}\left(\beta_{s,k}^2 + 3\delta^2\right), \tag{2.22}$$

$$\sigma_s^2 \triangleq \frac{1}{LK^2}\sum_{k=1}^{K}\frac{1}{p_k}\left\{\sigma_{s,k}^2 + \left(3 + \frac{6}{E_kB_k}\right)\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2\right\}, \tag{2.23}$$

$$\beta_{s,k}^2 \triangleq \frac{3\delta^2}{E_kB_k}\left(1 + \frac{1}{N_k^2}\sum_{n=1}^{N_k}\frac{1}{p_n^{(k)}}\right), \tag{2.24}$$

$$\sigma_{s,k}^2 \triangleq \frac{6}{E_kB_kN_k^2}\sum_{n=1}^{N_k}\frac{1}{p_n^{(k)}}\|\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})\|^2. \tag{2.25}$$

On the other hand, sampling agents and data without replacement results in the same bound but without the scaling by L *in the constants* β_s^2 *and* σ_s^2.

Proof. Proof in Appendix 2.B. □

The term $\sigma_{s,k}^2$ in the bound captures what we call *data variability*. It is controlled by the mini-batch size B_k; as the mini-batch increases the effect of this term is reduced. The $\|\nabla_{w^\mathsf{T}}J_k(w^o)\|$ term quantifies the suboptimality of the global model locally; we call its effect *model variability*. It is reduced when the data and agents are more heterogeneous. From Assumption 2.2, we can bound it uniformly. Both these terms capture the inherent differences in the distribution of the data. The data variability is a direct measure of it, while the model variablity is an indirect measure.

Now that we have identified the mean and variance of the gradient noise, we can proceed to establish the important conclusion that the following centralized solution:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu\frac{1}{L}\sum_{\ell\in\mathcal{L}_i}\widehat{\nabla_{w^\mathsf{T}}J_\ell}(\boldsymbol{w}_{i-1}), \tag{2.26}$$

converges exponentially to an $O(\mu)-$neighbourhood of the optimizer. In this implementation, the center processor aggregates the approximate gradients of the selected agents. We will subsequently call upon this result to examine the convergence behavior of the proposed federated learning solution.

Lemma 2.2 (**Mean-square error convergence of centralized solution**)**.** *Consider the centralized recursion* (2.26) *where the cost functions satisfy Assumption 2.1, and where the first and second order moments of the gradient noise process satisfy the conditions in Lemma 2.1. Also, the samples are chosen without replacement. For step-size values satisfying* $\mu < 2\nu/(\delta^2+\beta_s^2)$*, it holds that* $\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2$ *converges exponentially fast according to the recursion* (2.27)*, where* $\lambda = 1-2\mu\nu+\mu^2(\delta^2+\beta_s^2)\in[0,1)$ *:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \lambda\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \mu^2\sigma_s^2. \tag{2.27}$$

It follows from (2.27) *that, for sufficiently small step-sizes:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \lambda^i\mathbb{E}\|\widetilde{\boldsymbol{w}}_0\|^2 + \frac{1-\lambda^i}{1-\lambda}\mu^2\sigma_s^2. \tag{2.28}$$

Proof. see Appendix 2.C. □

We next bound the incremental noise $\boldsymbol{q}_i$. To do so, we introduce the local terms:

$$\boldsymbol{q}_{k,i,e} \triangleq \frac{1}{Kp_k}\left(\frac{1}{B_k}\sum_{b\in\mathcal{B}_{k,i,e}}\frac{1}{N_kp_b^{(k)}}\nabla_{w^\intercal}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,b}) - \nabla_{w^\intercal}J_k(\boldsymbol{w}_{k,e-1})\right). \tag{2.29}$$

We show in the next lemma that the local gradient noise $\boldsymbol{q}_{k,i,e}$ has zero mean and bounded variance. This result is useful for showing that the local SGD steps converge in the mean-square error sense towards their local models w_k^o.

Lemma 2.3 (**Estimation of moments of the local gradient noise**)**.** *The local gradient noise defined in* (2.29) *has zero mean* $\mathbb{E}\left\{\boldsymbol{q}_{k,i,e}\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\}=0$*, and bounded variance, regardless of the sampling scheme:*

$$\mathbb{E}\left\{\|\boldsymbol{q}_{k,i,e}\|^2\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\} \leq \frac{E_k}{K^2p_k^2}\beta_{s,k}^2\|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \frac{1}{K^2p_k^2}\sigma_{q,k}^2, \tag{2.30}$$

where $\mathcal{F}_{e-1}=\{w_{k,0},w_{k,1},\cdots,w_{k,e-1}\}$ *is the filtration describing all sources of randomness due to the previous iterates,* $\widetilde{\boldsymbol{w}}_{k,e}=w_k^o-\boldsymbol{w}_{k,e}$*, and the constants are as defined in* (2.24) *and:*

$$\sigma_{q,k}^2 = \frac{3}{B_k N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}} \|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2. \tag{2.31}$$

Proof. Proof in Appendix 2.D. □

Now that we have shown that the local gradient noise of the incremental step has bounded variance, we can study the mean square deviation of the local SGD.

Lemma 2.4 (**Mean-square error convergence of local incremental step**). *For every agent k, consider the local stochastic gradient recursion* (2.10) *where the cost function is subject to Assumption 2.1, and where the first and second order moments of the gradient noise process satisfy the conditions in Lemma 2.3. For step-size values satisfying:*

$$\mu < \frac{2\nu}{\delta^2 + \frac{E_k}{K^2 p_k^2}\beta_{s,k}^2}, \tag{2.32}$$

it holds that $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,e}\|^2$ converges exponentially fast according to the recursion:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,e}\|^2 \leq \lambda_k \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \mu^2\sigma_{q,k}^2, \tag{2.33}$$

where:

$$\lambda_k = 1 - 2\nu\mu + \mu^2\left(\delta^2 + \frac{E_k}{K^2 p_k^2}\beta_{s,k}^2\right) \in [0,1). \tag{2.34}$$

It follows from (2.33) *that, for sufficiently small step-sizes:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,e}\|^2 \leq \lambda_k^e \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,0}\|^2 + \frac{1-\lambda_k^e}{1-\lambda_k}\mu^2\sigma_{q,k}^2. \tag{2.35}$$

Proof. Proof in Appendix 2.E. □

We can finally bound the incremental noise in the following lemma.

Lemma 2.5 (**Estimation of the second order moment of incremental noise**). *The incremental noise defined in* (2.18) *has bounded variance:*

$$\mathbb{E}\|\boldsymbol{q}_i\|^2 \leq O(\mu)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + O(\mu)\xi^2 + O(\mu^2)\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2, \tag{2.36}$$

where the $O(\cdot)$ terms depend on epoch sizes, local convergence rates, total number of data samples, number of agents, Lipschitz constant, and data and agent normalized inclusion probabilities. Thus, $\mathbb{E}\|\boldsymbol{q}_i\|^2 = O(\mu)$.

Proof. Proof in Appendix 2.F. □

We observe an average data variability term across agents $\sigma_{q,k}^2$, and a model variability term ξ^2. However, the effect of the latter dominates since it is multiplied by an $O(\mu)$[1] term as opposed to $O(\mu^2)$. Furthermore, the variance of the incremental noise is non-zero, however it is bounded by the step-size.

2.3.3 Main Theorem

Now that we have bounded each term of (2.20) and using Lemmas 2.2 and 2.5, we find:

$$\begin{aligned}\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 &\leq \frac{1}{\alpha}\left(\lambda\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \mu^2\sigma_s^2\right) + \frac{O(\mu^3)}{1-\alpha}\left(\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \xi^2\right) + \frac{O(\mu^4)}{1-\alpha}\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2 \\ &= \lambda'\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \frac{\mu^2\sigma_s^2}{\alpha} + \frac{O(\mu^3)}{1-\alpha}\xi^2 + \frac{O(\mu^4)}{1-\alpha}\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2, \end{aligned} \tag{2.37}$$

where:

$$\begin{aligned}\lambda' &\triangleq \frac{\lambda}{\alpha} + \frac{O(\mu^3)}{1-\alpha} \\ &= \frac{1-2\mu\nu+\mu^2(\delta^2+\beta_s^2)}{\alpha} + \frac{O(\mu^3)}{1-\alpha} \\ &= O\left(\frac{1}{\alpha}\right) + O\left(\frac{\mu}{\alpha}\right) + O\left(\frac{\mu^2}{\alpha}\right) + O\left(\frac{\mu^3}{1-\alpha}\right). \end{aligned} \tag{2.38}$$

[1] The $O(\mu)$ notation replaces a constant multiplying μ, i.e., $O(\mu) = c\mu$ for some $c \in \mathbb{R}^+$.

Applying this bound recursively we obtain:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq (\lambda')^i\,\mathbb{E}\|\widetilde{\boldsymbol{w}}_0\|^2 + \frac{1-(\lambda')^i}{1-\lambda'}\left(\frac{\mu^2\sigma_s^2}{\alpha} + \frac{O(\mu^3)}{1-\alpha}\xi^2 + \frac{O(\mu^4)}{1-\alpha}\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2\right), \tag{2.39}$$

and if we were to repeat the algorithm infinitly many times, i.e., taking limit $i \to \infty$, we would get the following bound:

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \frac{1}{1-\lambda'}\left(\frac{\mu^2\sigma_s^2}{\alpha} + \frac{O(\mu^3)}{1-\alpha}\xi^2 + \frac{O(\mu^4)}{1-\alpha}\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2\right), \tag{2.40}$$

for $\lambda' < 1$, which for $\alpha = \sqrt{\lambda}$ is achieved when:

$$\mu < \min\left\{\frac{2\nu}{\delta^2+\beta_s^2}, \frac{2\nu}{\delta^2+\frac{E_k}{K^2p_k^2}\beta_{s,k}^2}\right\}, \tag{2.41}$$

$$O(\mu^3) < (1-\sqrt{\lambda})^2. \tag{2.42}$$

Thus, since $\alpha = O(1)$, $1-\alpha = O(\mu)$, and $1-\lambda' = O(\mu)$:

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq O(\mu)\sigma_s^2 + O(\mu)\xi^2 + O(\mu^2)\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2. \tag{2.43}$$

The result is summarized in the theorem.

Theorem 2.1 (Mean-square error convergence of FL under importance sampling). *Consider the iterates $\boldsymbol{w}_i$ generated by the importance sampling federated averaging algorithm. For sufficiently small step-size μ, it holds that the mean-square error converges exponentially fast:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \lambda'\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + O(\mu^2)\left(\sigma_s^2+\xi^2\right) + O(\mu^3)\frac{1}{K}\sum_{k=1}^{K}\sigma_{q,k}^2, \tag{2.44}$$

where $\lambda' = 1 - O(\mu) + O(\mu^2) \in [0,1)$.

2.4 Importance Sampling

Due to the heterogeneity of nodes, which arise from their data and computational capabilities, it is important to guide the algorithm based on the potential contribution that each agent can have on the overall performance. By allowing asynchronicity, i.e., different epoch sizes among agents, we can take advantage of the varying computational capabilities. From [71], we know that the choice of samples at each iteration affects the solution. Therefore, instead of choosing the samples uniformly, we consider importance sampling where samples are chosen according to some distribution to be determined. A similar scheme can be enforced on the participating agents. In what follows, we show that using importance sampling enhances the overall performance.

2.4.1 Agent Level: Importance Sampling of Data

Every agent k at each epoch must select a mini-batch of data based on the normalized inclusion probabilities $p_n^{(k)}$. To find the optimal probabilities, we minimize the bound on the variance of the local gradient noise $\sigma_{s,k}^2$, namely:

$$\left\{p_n^{(k),o}\right\}_{n=1}^{N_k} = \underset{\sum_{n=1}^{N_k} p_n^{(k)}=1}{\operatorname{argmin}} \quad \sigma_{s,k}^2. \tag{2.45}$$

We solve the problem for both cases when sampling is done with replacement and without replacement. The results are the same for both sampling schemes.

Lemma 2.6 (Optimal local data inclusion probabilities). *The optimal local data normalized inclusion probabilities are given by:*

$$p_n^{(k),o} \triangleq \frac{\|\nabla_{w^\intercal} Q_k(w^o; x_{k,n})\|}{\sum_{m=1}^{N_k} \|\nabla_{w^\intercal} Q_k(w^o; x_{k,m})\|}. \tag{2.46}$$

Proof. We first introduce a Lagrange multiplier and then calculate the probabilities by setting the derivatives with respect to $p_n^{(k)}$ and the Lagrange multiplier to 0. □

As seen in Lemma 2.6, more weight is given to a data point that has a greater gradient norm, thus increasing its chances of being sampled and resulting in a faster convergence rate. In addition, we observe that the more homogeneous the data is the more uniform the inclusion probability is.

2.4.2 Cloud Level: Importance Sampling of Agents

At each iteration, the cloud must select a subset of agents to participate. The agents are selected in accordance with the normalized inclusion probabilities p_k. To find the optimal probabilities, we minimize the bound on the variance of the gradient noise σ_s^2, namely:

$$\{p_k^o\}_{k=1}^{K} = \underset{\sum_{k=1}^{K} p_k=1}{\operatorname{argmin}} \quad \sigma_s^2. \tag{2.47}$$

The following result holds for sampling with and without replacement, since the gradient noise only differ by a multiplicative factor.

Lemma 2.7 (**Optimal agent inclusion probabilities for sampling with replacement**)**.** *The optimal agent normalized inclusion probabilities are given by:*

$$p_k^o \triangleq \frac{\sqrt{\sigma_{s,k}^2 + \alpha_k \|\nabla_{w^\mathsf{T}} J_k(w^o)\|^2}}{\sum_{\ell=1}^K \sqrt{\sigma_{s,\ell}^2 + \alpha_\ell \|\nabla_{w^\mathsf{T}} J_\ell(w^o)\|^2}}. \tag{2.48}$$

where:

$$\alpha_k = \left(3 + \frac{6}{E_k B_k}\right). \tag{2.49}$$

Proof. The proof follows similarly to that of Lemma 2.6. □

We observe that the normalized inclusion probabilities will be closer to a uniform distribution the more the data and model variability terms are similar across agents.

2.4.3 Practical Issues

In the previous subsections, we focused on finding the optimal inclusion probabilities for the agents and data. However, several practical issues arise. The first is that all probabilities are calculated based on the optimal model w^o, which we do not have access to. To overcome this issue, we estimate the probabilities at each iteration by calculating them according to the current model $\boldsymbol{w}_{i-1}$. The current model is the best estimate of the true model, and as we perform more iterations of the algorithm, we improve the estimate of the model and thus, in turn, improve the estimate of the probabilities. Thus,

$$\widehat{\boldsymbol{p}}_n^{(k),o} = \frac{\|\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1}; x_{k,n})\|}{\sum_{m=1}^{N_k} \|\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1}; x_m)\|}, \tag{2.50}$$

$$\widehat{\boldsymbol{p}}_k^o = \frac{\sqrt{\sigma_{s,k}^2 + \alpha_k \|\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1})\|^2}}{\sum_{\ell=1}^K \sqrt{\sigma_{s,\ell}^2 + \alpha_\ell \|\nabla_{w^\mathsf{T}} J_\ell(\boldsymbol{w}_{i-1})\|^2}}. \tag{2.51}$$

In addition, since calculating the true gradient of the local loss function is costly, we replace it with the mini-batch approximation when calculating p_k^o:

$$\widehat{\boldsymbol{p}}_k^o = \frac{\sqrt{\sigma^2_{s,k} + \alpha_k \left\|\widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{i-1})\right\|^2}}{\sum_{\ell=1}^{K} \sqrt{\sigma^2_{s,\ell} + \alpha_\ell \left\|\widehat{\nabla_{w^\mathsf{T}} J_\ell}(\boldsymbol{w}_{i-1})\right\|^2}}. \tag{2.52}$$

Furthermore, every agent has access to all of its data and consequently to all of the gradients. However, the cloud does not have access to the gradients of all agents, and in turn cannot calculate the denominator of p_k. Instead, we propose the following solution: at iteration 0, all probabilities are set to $p_k = \frac{1}{K}$; then, during the i^{th} iteration, after the participating agents $\ell \in \mathcal{L}_i$ send the cloud the norm of their stochastic gradients $\|\widehat{\nabla_{w^\mathsf{T}} P_\ell}(\boldsymbol{w}_{i-1})\|$, the probabilities are updated as follows:

$$\widehat{\boldsymbol{p}}_k^o = \frac{\sqrt{\sigma^2_{s,k} + \alpha_k \left\|\widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{i-1})\right\|^2}}{\sum_{\ell \in \mathcal{L}_i} \sqrt{\sigma^2_{s,\ell} + \alpha_\ell \left\|\widehat{\nabla_{w^\mathsf{T}} J_\ell}(\boldsymbol{w}_{i-1})\right\|^2}} \left(1 - \sum_{\ell \in \mathcal{L}_i^c} \widehat{\boldsymbol{p}}_\ell^o\right), \tag{2.53}$$

where the multiplicative factor follows from ensuring all the probabilities $\widehat{p}_k^o$ sum to 1. Similarly for the local probabilities, since we are implementing mini-batch SGD, we only update the probabilities of the data points that were sampled:

$$\widehat{\boldsymbol{p}}_n^{(k),o} = \frac{\|\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1}; x_{k,n})\|}{\sum_{b \in \mathcal{B}_{k,i,e}} \|\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1}; x_b)\|} \left(1 - \sum_{b \in \mathcal{B}^c_{k,i,e}} \widehat{\boldsymbol{p}}_b^{(k),o}\right). \tag{2.54}$$

Finally, the last problem arises when sampling without replacement. We have found the optimal inclusion probabilities and not the optimal sampling probabilities, and moving from the former to the latter is not trivial. Thus, we rely on the literature under sampling without replacement with unequal probabilities. Multiple sampling schemes exist such that the sampling probabilities do not need to be calculated explicitly. In general, there are multiple non-uniform sampling without replacement schemes that guarantee the same inclusion probabilities. We choose to implement the sampling scheme proposed in [77], which ensures the inclusion probabilities are p_k^o and $p_n^{(k),o}$ for the agents and data, respectively. More explicitly, we first calculate the *progressive totals* of the inclusion probabilities:

$$\Pi_k = \sum_{\ell=1}^{k} L p_k^o, \tag{2.55}$$

for $k = 1, 2, \cdots, K$, and we set $\Pi_0 = 0$. Then, we select uniformly at random a *uniform variate* $d \in [0, 1)$. Then, we select the L agents that satisfy $\Pi_{k-1} \leq q + \ell < \Pi_k$, for some $\ell = 0, 1, \cdots, L-1$, i.e., for every $q+\ell$, the agent satisfying the previous condition is selected. The same is done for the samples at the agents. To further understand the scheme, we include the following example: consider we have a set of 6 elements, and we wish to select 2. The inclusion probabilities are given by: $\{15/200, 81/200, 26/200, 42/200, 20/200, 16/200\}$. We calculate the progerssing totals to get $\Pi = \{0.15, 0.96, 1.22, 1.64, 1.84, 2\}$. If we choose $d = 0.57$, then for $\ell = 0, 1$, we have $d + \ell = 0.57, 1.57$. Thus, the first and fourth elements are chosen.

2.5 Privacy Analysis

We next show that it is preferable for the FL algorithm to share gradient information, as opposed to model iterates, in order to enhance privacy. For the sake of simplicity we assume agents and data points are sampled uniformly. Thus, if we were to introduce differential privacy to federated learning, then a random Laplacian noise $\boldsymbol{g}_{k,i}$ should be added to each model by the client before aggregation by the server, and the new privatized aggregation step becomes:

$$\boldsymbol{w}_i = \frac{1}{L} \sum_{k \in \mathcal{L}_i} \boldsymbol{w}_{k,i} + \boldsymbol{g}_{k,i}. \tag{2.56}$$

However, if we were to study the MSE convergence of the privatized algorithm, we would notice a new $O(\mu^{-1})\sigma_g^2$ term in the bound of Theorem 3.1 in Chapter 3. Thus, we now describe an alternative implementation that shares gradients as opposed to weight estimates. Note first that the FL algorithm can be expressed in a single step taken from the server's perspective:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{L} \sum_{k \in \mathcal{L}_i} \frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}). \tag{2.57}$$

This suggests that instead of every agent sharing its final model $\boldsymbol{w}_{k,i}$, they could share the total update:

$$\frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}). \tag{2.58}$$

The server then aggregates the updates from all participating agents and updates the previous model $\boldsymbol{w}_{i-1}$. In this case, if we were to privatize this new version of the algorithm, we would add random noise to the updates which are then scaled by the step-size:

$$\boldsymbol{\psi}_{k,i-1} = \frac{1}{E_k} \sum_{e=1}^{E_k} \widehat{\nabla_{w^\mathsf{T}} J_k}(\boldsymbol{w}_{k,e-1}), \tag{2.59}$$

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \frac{1}{L} \sum_{k \in \mathcal{L}_i} \left(\boldsymbol{\psi}_{k,i-1} + \boldsymbol{g}_{k,i} \right). \tag{2.60}$$

We show in the following theorem the effect of the added noise to the new FL algorithm.

It turns out the noise introduces an $O(\mu)$ error instead of $O(\mu^{-1})$.

Theorem 2.2 (**MSE convergence of privatized FL**). *Under assumptions 2.1 and 2.2, the privatized FL algorithm* (2.59)−(2.60) *converges exponentially fast for a small enough step-size to a neighbourhood of the optimal model:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \lambda\, \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + O(\mu^2)\sigma_s^2 + O(\mu^2)\xi^2 + \frac{\mu^2}{L}\sigma_g^2 + O(\mu^3). \tag{2.61}$$

where $\lambda = \sqrt{1 - 2\nu\mu + (\beta_s^2 + \delta^2)\mu^2} + O(\mu^2) \in (0,1)$*. Then, in the limit:*

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{1,i}\|^2 \leq O(\mu)(\sigma_s^2 + \xi^2 + \sigma_g^2) + O(\mu^2). \tag{2.62}$$

Proof. See Appendix 2.G. □

Thus, sharing the updates instead of the models is more advantageous since the effect of the added noise on the performance is reduced. The $O(\mu)$ factor allows us to increase the value of the noise variance while ensuring the model utility does not deteriorate significantly. Therefore, to guarantee an $\epsilon(i)-$differentially private algorithm, we let the added noise be a zero-mean Laplacian random variable with σ_g^2 variance. To show this, we first bound the sensitivity of the algorithm by assuming without loss of generality that agent 1 replaces its original dataset by $\{x'_{1,n}\}$ and thus resulting in new model trajectories $\boldsymbol{w}'_i$. For some constants B and B' it can be shown that the sensitivity is bounded as follows:

$$\begin{aligned} \Delta(i) &\overset{\Delta}{=} \|\boldsymbol{w}_i - \boldsymbol{w}'_i\| \\ &\leq \|\widetilde{\boldsymbol{w}}_i\| + \|\widetilde{\boldsymbol{w}}'_i\| + \|w^o - w'^o\| \\ &\leq B + B' + \|w^o - w'^o\|, \end{aligned} \tag{2.63}$$

with high probability given by:

$$\begin{aligned} \mathbb{P}\left(\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\|\right) \geq &\left(1 - \frac{\lambda^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_0\|^2 + O(\mu)}{B^2}\right) \\ &\times \left(1 - \frac{\lambda'^i \mathbb{E}\|\widetilde{\boldsymbol{w}}'_0\|^2 + O(\mu)}{B'^2}\right). \end{aligned} \tag{2.64}$$

Using Markov's inequality and Theorem 2.2, we can bound the probability that the errors are unbounded:

$$\mathbb{P}(\|\widetilde{\boldsymbol{w}}_i\| \geq B) \leq \frac{\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2}{B^2} \leq \frac{\lambda^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_0\|^2 + O(\mu)}{B^2}, \tag{2.65}$$

$$\mathbb{P}(\|\widetilde{\boldsymbol{w}}_i'\| \geq B') \leq \frac{\mathbb{E}\|\widetilde{\boldsymbol{w}}_i'\|^2}{B'^2} \leq \frac{\lambda'^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_0'\|^2 + O(\mu)}{B'^2}, \tag{2.66}$$

and then conclude (2.64). Now that we have a bound on the sensitivity, we can show $\epsilon(i)-$differential privacy by first starting with the condition in the definition of differential privacy.

Definition 2.1 **($\epsilon(i)-$Differential privacy).** *We say that the algorithm given in* (2.59)–(2.60) *is $\epsilon(i)-$differentially private at time i if the following condition holds on the joint distribution $f(\cdot)$:*

$$\frac{f\left(\left\{\{\boldsymbol{\psi}_{k,j-1} + \boldsymbol{g}_{k,j}\}_{k\in\mathcal{L}_i}\right\}_{j=1}^{i}\right)}{f\left(\left\{\left\{\boldsymbol{\psi}'_{k,j-1} + \boldsymbol{g}_{k,j}\right\}_{k\in\mathcal{L}_i}\right\}_{j=1}^{i}\right)} \leq e^{\epsilon(i)}. \tag{2.67}$$

We can bound the ratio of the joint distributions by first using Bayes' rule and the independence of the added noise to get a product of Laplacian distributions, and then using triangle inequality and the bound on the sensitivity:

$$\begin{aligned}
\frac{f\left(\left\{\{\boldsymbol{\psi}_{k,j-1} + \boldsymbol{g}_{k,j}\}_{k\in\mathcal{L}_i}\right\}_{j=1}^{i}\right)}{f\left(\left\{\left\{\boldsymbol{\psi}'_{k,j-1} + \boldsymbol{g}_{k,j}\right\}_{k\in\mathcal{L}_i}\right\}_{j=1}^{i}\right)} &= \prod_{j=1}^{i} \frac{\exp\left(-\frac{\sqrt{2}}{\sigma_g}\|\boldsymbol{\psi}_{k,j-1} + \boldsymbol{g}_{k,j}\|\right)}{\exp\left(-\frac{\sqrt{2}}{\sigma_g}\|\boldsymbol{\psi}'_{k,j-1} + \boldsymbol{g}_{k,j}\|\right)} \\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g}\sum_{j=1}^{i}\|\boldsymbol{\psi}_{k,i-1} - \boldsymbol{\psi}'_{k,i-1}\|\right) \\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g}\sum_{j=1}^{i}\Delta(j)\right) \\
&= \exp\left(\frac{\sqrt{2}}{\sigma_g}(B + B' + \|w^o - w'^o\|)i\right).
\end{aligned} \tag{2.68}$$

Thus, the algorithm is $\epsilon(i)-$differentially private for:

$$\epsilon(i) = \frac{\sqrt{2}}{\sigma_g}(B + B' + \|w^o - w'^o\|)i, \tag{2.69}$$

and with high probability.

2.6 Experimental Results

To illustrate the theoretical results, we devise two experiments. The first consists of simulated data with quadratic risk functions, and the second consists of a real dataset with logistic risk functions. We further study the effect of some architectural and algorithmic constants.

2.6.1 Regression

We first validate the theory on a regression problem. We consider $K = 300$ agents, for which we generate $N_k = 100$ data points for each agent k as follow: Let $\boldsymbol{u}_{k,n}$ denote an independent streaming sequence of two-dimensional random vectors with zero mean and covariance matrix $R_{u_k} = \mathbb{E}\, \boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathsf{T}}$. Let $\boldsymbol{d}_k(n)$ denote a streaming sequence of random variables that have zero mean and variance $\sigma_{d_k}^2 = \mathbb{E}\, \boldsymbol{d}_k^2(n)$. Let $r_{d_k u_k} = \mathbb{E}\, \boldsymbol{d}_k(n)\boldsymbol{u}_{k,n}$ be the cross-variance vector. The data $\{\boldsymbol{d}_k(n), \boldsymbol{u}_{k,n}\}$ are related by the following linear regression model:

$$\boldsymbol{d}_k(n) = \boldsymbol{u}_{k,n}^T w^\star + \boldsymbol{v}_k(n), \tag{2.70}$$

for some randomly generated parameter vector $w^\star$ and where $\boldsymbol{v}_k(n)$ is a zero mean white noise process with variance $\sigma_{v_k}^2 = \mathbb{E}\boldsymbol{v}_k^2(n)$, independent of $\boldsymbol{u}_{k,n}$. The local risk is given by:

$$J_k(w) = \frac{1}{N_k}\sum_{n=1}^{N_k} \|\boldsymbol{d}_k(n) - \boldsymbol{u}_{k,n}^{\mathsf{T}} w\|^2 + \rho\|w\|^2, \tag{2.71}$$

and so the loss function becomes:

$$Q_k(w; \boldsymbol{d}_k(n), \boldsymbol{u}_{k,n}) = \|\boldsymbol{d}_k(n) - \boldsymbol{u}_{k,n}^{\mathsf{T}} w\|^2 + \rho\|w\|^2. \tag{2.72}$$

We set $\rho = 0.001$, while the batch sizes B_k and the epoch sizes E_k are chosen uniformly at random from the range $[1, 10]$ and $[1, 5]$, respectively. During each iteration, there are $L = 6$ active agents. To test the performance of the algorithm, we calculate at each iteration the mean-square deviation (MSD) of the parameter vector $\boldsymbol{w}_i$ with respect to the true model w^o:

$$\text{MSD}_i = \|\boldsymbol{w}_i - w^o\|^2. \tag{2.73}$$

The optimization problem has the closed form expression:

$$w^o = \left(\widehat{R}_u + \rho I\right)^{-1} \widehat{R}_u w^\star + \left(\widehat{R}_u + \rho I\right)^{-1} \widehat{r}_{uv}, \tag{2.74}$$

where:

$$\widehat{R}_u \triangleq \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}\boldsymbol{u}_{k,n}^{\mathsf{T}}\boldsymbol{u}_{k,n}, \tag{2.75}$$

$$\widehat{r}_{uv} \triangleq \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{n=1}^{N_k}\boldsymbol{v}_k(n)\boldsymbol{u}_{k,n}. \tag{2.76}$$

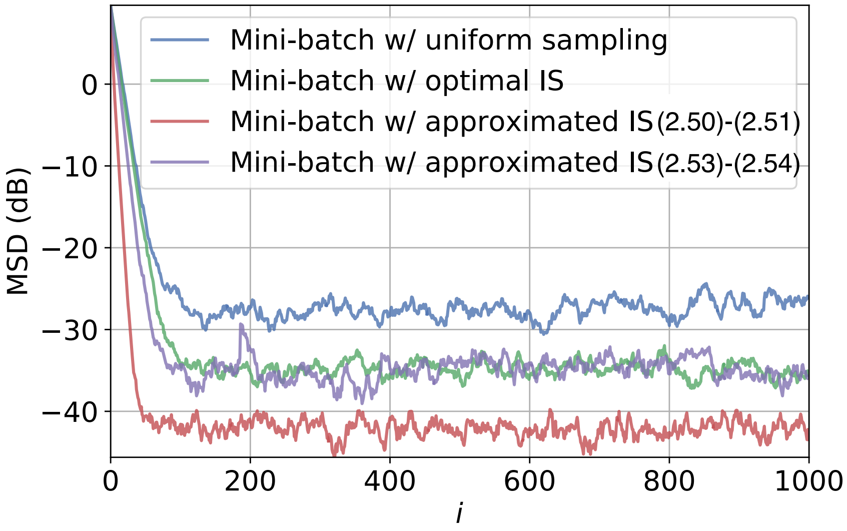
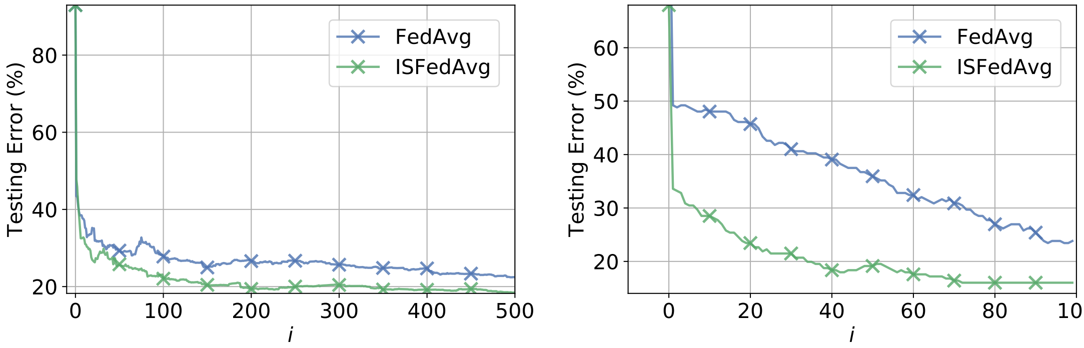


Figure 2.1 – MSD plots of the regression problem: blue curve is the standard mini-batch implementation, green curve is the importance sampling implementation with the true probabilities, red curve is the importance sampling implementation with approximate probabilities (2.50)–(2.51), purple curve is the importance sampling implementation with approximate probabilities (2.53)–(2.54).

We run four tests: we first run the standard FedAvg algorithm where the mini-batches are chosen uniformly with replacement. We then run Algorithm 2.1, once with the optimal probabilities $p_n^{(k)}$ and p_k in (2.46)–(2.48), once with the approximate probabilities (2.50)–(2.51), and once with (2.53)–(2.54). We implement the sampling scheme from [77]. We set the step-size $\mu = 0.01$. Each test is repeated 100 times, and the resulting MSD is averaged. We get the curves as shown in Figure 2.1. We see that the importance sampling scheme does better than the standard sampling algorithm. This comes as no surprise, since the probabilities were chosen to minimize the bound on the MSD. The importance sampling scheme (green curve) improved the MSD bound by 13.1 dB compared to the standard federated learning scheme (blue curve), i.e., it resulted in a lower gradient noise variance. In addition, both schemes converge exponentially fast, namely in the transient state the effect of the previous error is decreasing due to the multiplication by the convergence rate λ' (see first term in Theorem 2.1). In the steady state, the MSD plateaus at a constant that is mainly dependent on the gradient noise variance σ_s^2 and the model variability ξ^2. Furthermore, we observe that approximate probabilities do not degrade the performance of the algorithm. Our proposed approximate solution (2.53)–(2.54) (purple curve) performs just as well as using the true probabilities. In fact, we observe that the approximate probabilities converge quadratically to the true ones. Similarly, the approximate probabilities (2.51)–(2.50) do not degrade the overall performance. They, in fact, outperform the other solutions and converge faster (red curve). This is not surprising, since, at each iteration, we are attributing higher probabilities to agents and data points that have greater gradients. We are increasing their chances of being selected and thus taking steeper steps towards the true model.

2.6.2 Classification

We next study the theory in a classification context. We consider the ijcnn1 dataset [78]. The dataset consists of 35000 training samples and 91701 testing samples of $M = 22$ attributes. We distribute the data randomly in a non-IID fashion to $K = 100$ agents. Each agent receives a random number N_k of data points, where N_k ranges from 79 to 688. We run the two algorithms FedAvg and ISFedAvg. We set $\mu = 0.25$, $\rho = 0.0001$, $L = 10$, and B_k and E_k chosen as before. We also consider the Avazu click through dataset [79] that consists of 10 days worth of data points to study a click prediction system of online adds. We split 5101 data points each of 10^6 features to $K = 20$ agents, each receiving between $N_k = 235$ to $N_k = 780$ data points. Since the data is IID, we add Guassian noise to the data points of each agent. The noise is non-IID accross the agents. We set $\mu = 0.1$, $\rho = 0.01$, $L = 3$, $B_k \in [1, 20]$ and $E_k \in [1, 10]$. We plot the testing error in Figure 2.2. We observe that importance sampling improves the testing error from 22.45% to 18.46% for ijcnn1 and from 22.26% to 14.45%. This is because importance sampling is more sample efficient.



(a) IJCNN1 Dataset

(b) Avazu Dataset

Figure 2.2 – Testing error plots of the classification problem.

2.6.3 Effect of Parameters

We return to the regression problem to study the effect of the number of sampled agents L, the epoch size E_k, and the non-iid factor of the data captured by ξ. We study their effect in the regression setting since it is a more controlled environment that will allow us to limit the effect of unknown factors.

We first look at the effect of the number of sampled agents L per iteration. We varry $L \in \{6, 30, 60\}$ and compare the difference between FedAvg using uniform sampling versus importance sampling. We plot the results in Figure 2.3. We choose to not plot the MSD in the log domain since in the linear domain the effect is clearer. We observe that as the value of L increases the improved performance of ISFedAvg decreases. Thus, for large values of L, ISFedAvg is not much more advantageous than FedAvg. We also observe that as L increases both algorithms perform better. This is not surprising, since as we increase L we improve our estimate of the true gradient taken over all the K agents, i.e., decreasing the gradient noise $\boldsymbol{s}_i$.

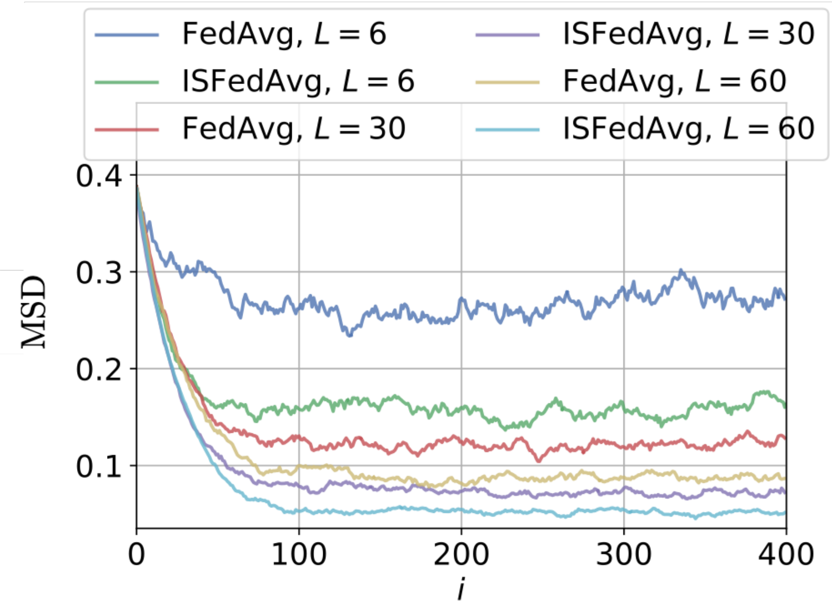


Figure 2.3 – MSD plots of the regression problem with varying L.

We next study the effect of varrying the epoch size E_k. We run three experiments for different range values of E_k. During each of these experiments we compare FedAvg with ISFedAvg. The corresponding plots are found in Figure 2.4. We observe that as E_k increases, the perfomance of FedAvg and ISFedAvg does not change much. This is not surprising, since in our modified version of FedAvg we normalize the gradient by E_k. If we were to repeat the same experiments with the standard FedAvg algorithm originally proposed by [11], increasing E_k would deteriorate the performance.

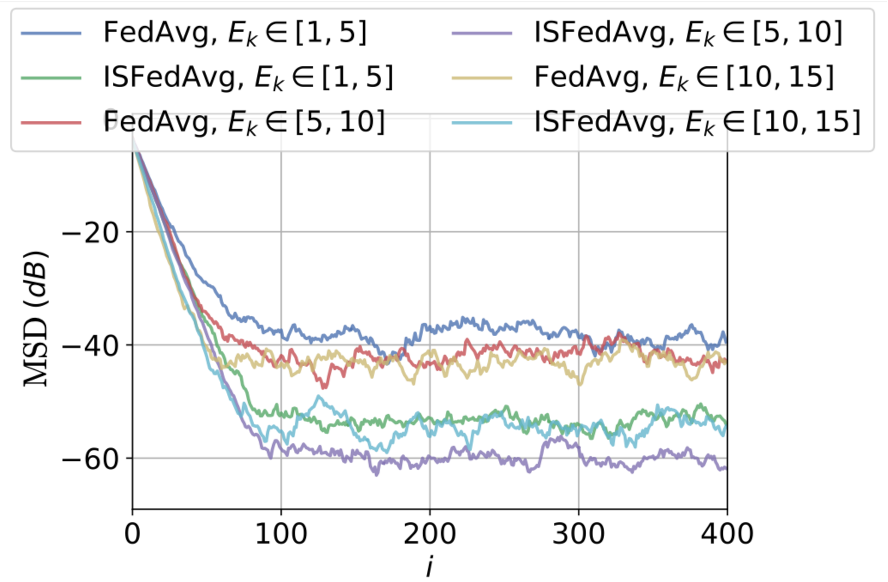
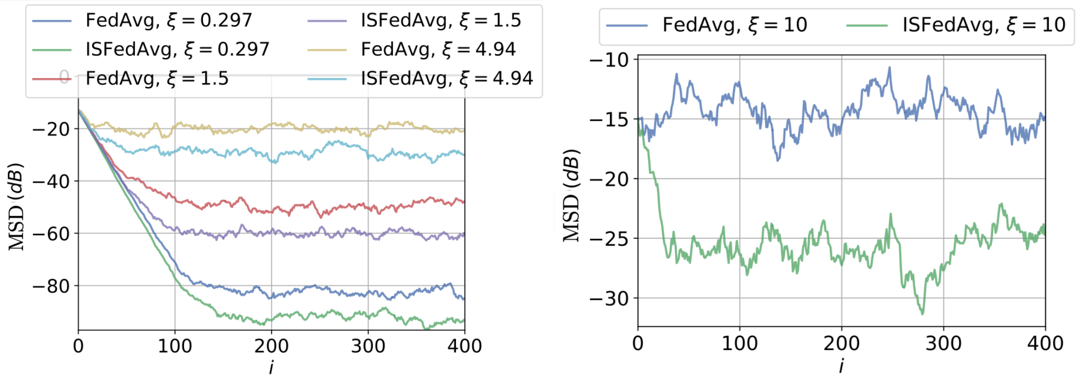


Figure 2.4 – MSD plots of the regression problem with varying E_k.

Finally, to study the effectc of the non-iid date, we vary the variability among the data distribution between the different agents. We control that by the covariance of the added noise $\{\boldsymbol{v}_k\}_{k=1}^{K}$. To make the data more non-identical we increase the covariance and thus resulting in a higher value of ξ. We run four experiments for four different values of the covariance matrix, and we approximate the corresponding $\xi \in \{0.297, 1.5, 4.94, 10\}$. Observing the MSD plots in Figure 2.5a, we see that as ξ increases, i.e., the data is more non-IID, the performance of both algorithms worsens. However, ISFedAvg always does better than FedAvg. In addition, we would like to point out by increasing ξ even more we run the risk of both algorithms diverging. As seen in Figure 2.5b, we can conclude that there exists a range of values of ξ for which FedAvg diverges while ISFedAvg still manages to converge. Thus, FedAvg is more sensitive to the values of ξ.



(a) Varying ξ.

(b) Diverging ξ.

Figure 2.5 – MSD plots of the regression problem with varying ξ.

2.6.4 Effect of Privatization

We assume we have $K = 1000$ agents of which we choose $L = 30$ at a time. We generate non-iid datasets of varying size for each agent as in the previous section. We allow each agent to run varying epochs $E_k \in [1, 10]$ during an iteration of the algorithm. We set the step-size $\mu = 0.2$, $\rho = 0.007$ and $\sigma_g^2 = 0.02$. We compare three algorithms: the standard FL algorithm, the privatized FL algorithm with sharing of models, and the privatized FL algorithm with sharing of updates. We plot the average MSD curves after repeating the experiment 100 times. As expected, the effect of the added noise is worse when models are shared (Figure 2.6 *yellow curve*) than when updates are shared (Figure 2.6 *red curve*).

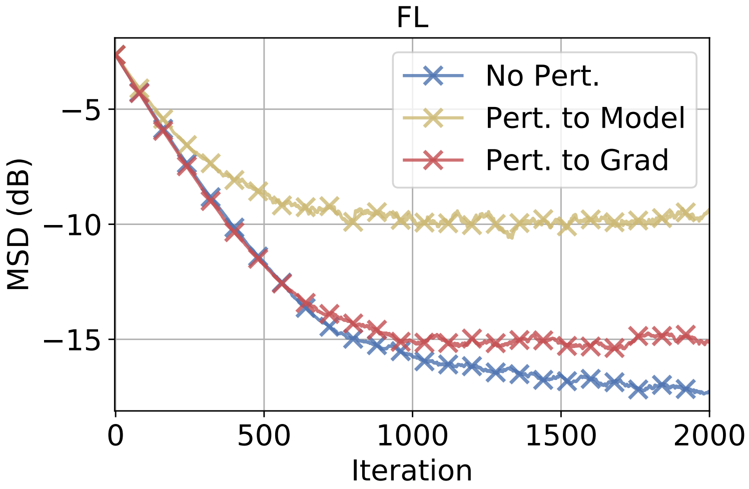
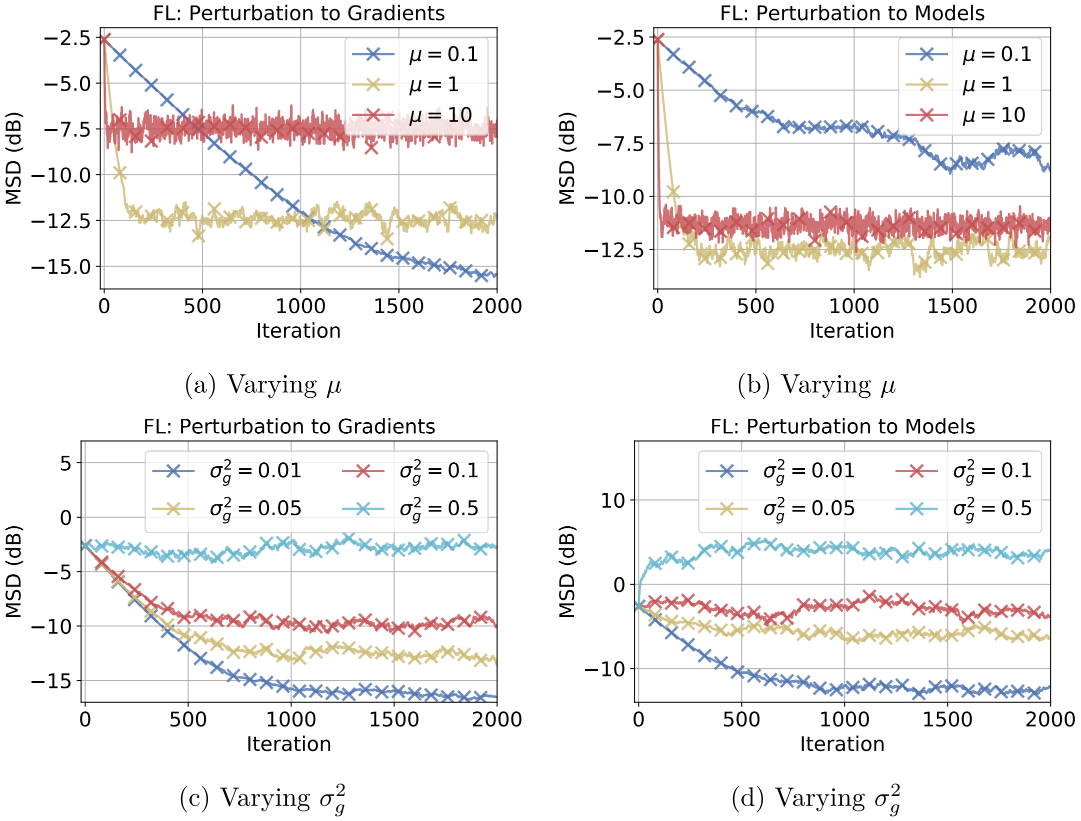


Figure 2.6 – MSD plots of privatized FL.

We next study the effect of the step-size on the MSD of the privatized FL algorithm. We expect that as μ is increased the MSD increases for the FL algorithm when updates are shared. While, when models are shared, since the gradient noise variance is tuned by μ and the added noise variance by μ^{-1}, we expect to observe a trade-off. On one hand, as μ is increased the effect of the gradient noise is increased while that of the added noise is diminished. On the other hand, as μ is decreased, the effect of the added noise overpowers that of the gradient noise. Indeed, we observe this phenomenon in (a) and (b) of Figure 2.7.

Finally, we study the effect of the variance of the added noise. We fix the step-size at $\mu = 0.2$ and vary the noise variance $\sigma_g^2 = \{0.01, 0.05, 0.1, 0.5\}$. In the two cases, as we increase σ_g^2 the performance diminishes ((c), (d) of Figure 2.7). However, the larger values of the added noise variance affect the perturbed models more than the perturbed gradients. The algorithm diverges for lower values of σ_g^2 in the case when models are shared as opposed to when gradients are shared. Thus, sharing updates can handle larger



(a) Varying μ

(b) Varying μ

(c) Varying σ_g^2

(d) Varying σ_g^2

Figure 2.7 – MSD plots of privatized FL with varying step-size and variance of added noise.

values of σ_g^2 before the algorithm diverges. In addition, since the variance is tuned by the step-size, we can always find a suitable μ to decrease its effect.

2.7 Conclusion

The work presented in this chapter incorporates two levels of importance sampling into the operation of federated learning: one for selecting agents and the other for selecting data batches at the agents. Optimal dynamical choices for the sampling probabilities are derived, and a detailed convergence analysis is performed. We also provided approximate expressions for the optimal sampling policies and illustrate the theoretical findings and the performance enhancement by means of simulations. Finally, we introduced privatization to federated learning by masking the messages. We showed that perturbing the updates instead of the models does not hinder performance. We dropped the assumption on the bounded gradients that rarely holds and showed the privatized federated learning algorithm is differentially private with high probability.

2.A Result on the Variance of the Mini-batch Estimate

Let $\{\mathcal{S} = \boldsymbol{x}_n \in \mathbb{R}^M\}_{n=1}^N$ denote a set of N *independent* random variables, each with mean $\mathbb{E}\boldsymbol{x}_n = \overline{x}_n$ and variance $\sigma_n^2 = \mathbb{E}\|\boldsymbol{x}_n - \mathbb{E}\boldsymbol{x}_n\|^2$. We consider the problem of estimating the expected value of the sample mean[2]:

$$\overline{x} \triangleq \mathbb{E}\left(\frac{1}{N}\sum_{n=1}^N \boldsymbol{x}_n\right) \tag{2.77}$$

We consider two estimators for $\overline{x}$, both constructed by considering a mini-batch of samples, where $\boldsymbol{x}_b^{\rm r}$ is sampled from $\mathcal{S}$ *with replacement* and $\boldsymbol{x}_b^{\rm nr}$ *without replacement.* Let p_n be the normalized inclusion probability of $\boldsymbol{x}_n$. We then define the two estimators:

$$\widehat{\boldsymbol{x}}^{\rm r} \triangleq \frac{1}{B}\sum_{b=1}^B \frac{1}{Np_b}\boldsymbol{x}_b^{\rm r}, \tag{2.78}$$

$$\widehat{\boldsymbol{x}}^{\rm nr} \triangleq \frac{1}{B}\sum_{b=1}^B \frac{1}{Np_b}\boldsymbol{x}_b^{\rm nr}. \tag{2.79}$$

Both estimators are unbiased, and it holds that:

$$\mathbb{E}\|\widehat{\boldsymbol{x}}^{\rm r} - \overline{x}\|^2 = \frac{1}{B}\sum_{n=1}^N p_n\left(\frac{1}{N^2p_n^2}\sigma_n^2 + \left\|\frac{1}{Np_n}\overline{x}_n - \overline{x}\right\|^2\right), \tag{2.80}$$

$$\mathbb{E}\|\widehat{\boldsymbol{x}}^{\rm nr} - \overline{x}\|^2 \le \sum_{n=1}^N p_n\left(\frac{1}{N^2p_n^2}\sigma_n^2 + \left\|\frac{1}{Np_n}\overline{x}_n - \overline{x}\right\|^2\right). \tag{2.81}$$

Proof. We begin with the *with-replacement* setting. The randomness of the samples introduces some intricacies that need to be accounted for in the notation. For the mean, we have:

$$\begin{aligned}\mathbb{E}\widehat{\boldsymbol{x}}^{\rm r} &= \frac{1}{B}\sum_{b=1}^B \mathbb{E}\left(\frac{1}{Np_b}\boldsymbol{x}_b^{\rm r}\right) = \frac{1}{B}\sum_{b=1}^B \mathbb{E}\left\{\mathbb{E}\left\{\frac{1}{Np_b}\boldsymbol{x}_b^{\rm r}\middle|\mathcal{S}\right\}\right\} \\ &= \frac{1}{B}\sum_{b=1}^B \mathbb{E}\left\{\sum_{n=1}^N p_n\frac{1}{Np_n}\boldsymbol{x}_n\right\} = \frac{1}{B}\sum_{b=1}^B \overline{x} = \overline{x}.\end{aligned} \tag{2.82}$$

[2] We introduce the following auxiliary result that is a slight variation of known results [80].

For the variance we find:

$$\begin{aligned}
&\mathbb{E}\|\widehat{\boldsymbol{x}}^{\mathrm{r}}-\overline{x}\|^2\\
&=\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B}\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2\\
&=\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right)\right\|^2\\
&=\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\|\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2+\frac{1}{B^2}\sum_{b_1\neq b_2}\mathbb{E}\left\{\left(\frac{1}{Np_{b_1}}\boldsymbol{x}_{b_1}-\overline{x}\right)^{\mathsf{T}}\left(\frac{1}{Np_{b_2}}\boldsymbol{x}_{b_2}-\overline{x}\right)\right\}\\
&\overset{(a)}{=}\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\|\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2+\frac{1}{B^2}\sum_{b_1\neq b_2}\mathbb{E}\left\{\frac{1}{Np_{b_1}}\boldsymbol{x}_{b_1}-\overline{x}\right\}\mathbb{E}\left\{\frac{1}{Np_{b_2}}\boldsymbol{x}_{b_2}-\overline{x}\right\}\\
&\overset{(b)}{=}\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\|\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2, \qquad (2.83)
\end{aligned}$$

where (a) is a result of the fact that the elements of $\mathcal{S}$ are independent and $\boldsymbol{x}_b^{\mathrm{r}}$ is sampled from $\mathcal{S}$ independently, and hence $\boldsymbol{x}_{b_1}$ and $\boldsymbol{x}_{b_2}$ are independent. Step (b) then follows from:

$$\mathbb{E}\left(\frac{1}{Np_b}\boldsymbol{x}_b\right)=\mathbb{E}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\right)=\overline{x}. \qquad (2.84)$$

Then:

$$\begin{aligned}
\mathbb{E}\|\widehat{\boldsymbol{x}}^{\mathrm{r}}-\overline{x}\|^2&=\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\|\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2\\
&=\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\{\mathbb{E}\left\|\frac{1}{Np_b}\boldsymbol{x}_b^{\mathrm{r}}-\overline{x}\right\|^2\middle|\mathcal{S}\right\}\\
&=\frac{1}{B^2}\sum_{b=1}^{B}\mathbb{E}\left\{\sum_{n=1}^{N}p_n\left\|\frac{1}{Np_n}\boldsymbol{x}_n-\overline{x}\right\|^2\right\}\\
&=\frac{1}{B^2}\sum_{b=1}^{B}\sum_{n=1}^{N}p_n\mathbb{E}\left\|\frac{1}{Np_n}\boldsymbol{x}_n-\overline{x}\right\|^2\\
&=\frac{1}{B}\sum_{n=1}^{N}p_n\mathbb{E}\left\|\frac{1}{Np_n}\boldsymbol{x}_n-\frac{1}{Np_n}\overline{x}_n+\frac{1}{Np_n}\overline{x}_n-\overline{x}\right\|^2\\
&=\frac{1}{B}\sum_{n=1}^{N}p_n\left(\mathbb{E}\left\|\frac{1}{Np_n}\boldsymbol{x}_n-\frac{1}{Np_n}\overline{x}_n\right\|^2+\left\|\frac{1}{Np_n}\overline{x}_n-\overline{x}\right\|^2\right)\\
&=\frac{1}{B}\sum_{n=1}^{N}p_n\left(\frac{1}{N^2p_n^2}\sigma_n^2+\left\|\frac{1}{Np_n}\overline{x}_n-\overline{x}\right\|^2\right). \qquad (2.85)
\end{aligned}$$

We now proceed to study the efficiency of the *without replacement* mini-batch mean.

The fact that the $\boldsymbol{x}_b$ are sampled from $\mathcal{S}$ without replacement causes pairs $\boldsymbol{x}_{b_1}, \boldsymbol{x}_{b_2}$ to no longer be independent. We denote the set of points sampled from $\mathcal{S}$ *without replacement* by $\mathcal{B}^{\mathrm{nr}}$ and introduce the activation function by:

$$\mathbb{I}_n \triangleq \begin{cases} 1, \text{ if } \boldsymbol{x}_n \in \mathcal{B}^{\mathrm{nr}}, \\ 0, \text{ if } \boldsymbol{x}_n \notin \mathcal{B}^{\mathrm{nr}}. \end{cases} \tag{2.86}$$

Then, the estimator $\widehat{\boldsymbol{x}}^{\mathrm{nr}}$ can be written equivalently as:

$$\widehat{\boldsymbol{x}}^{\mathrm{nr}} = \frac{1}{B} \sum_{n=1}^{N} \mathbb{I}_n \frac{1}{Np_n} \boldsymbol{x}_n. \tag{2.87}$$

For the mean, we have:

$$\begin{aligned} \mathbb{E}\widehat{\boldsymbol{x}}^{\mathrm{nr}} &= \frac{1}{B} \sum_{n=1}^{N} \mathbb{E}\left\{\mathbb{I}_n \frac{1}{Np_n} \boldsymbol{x}_n\right\} = \frac{1}{B} \sum_{n=1}^{N} \mathbb{E}\mathbb{I}_n \mathbb{E} \frac{1}{Np_n} \boldsymbol{x}_n \\ &= \frac{1}{B} \sum_{n=1}^{N} Bp_n \frac{1}{Np_n} \overline{x}_n = \frac{1}{N} \sum_{n=1}^{N} \overline{x}_n = \overline{x}. \end{aligned} \tag{2.88}$$

For the variance, we have:

$$\begin{aligned} \mathbb{E}\left\|\widehat{\boldsymbol{x}}^{\mathrm{nr}} - \overline{x}\right\|^2 &= \mathbb{E}\left\| \frac{1}{B} \sum_{n=1}^{N} \mathbb{I}_n \left(\frac{1}{Np_n} \boldsymbol{x}_n - \overline{x} \right) \right\|^2 \\ &= \frac{1}{B^2} \sum_{n=1}^{N} \mathbb{E}\left\| \mathbb{I}_n \left(\frac{1}{Np_n} \boldsymbol{x}_n - \overline{x} \right) \right\|^2 \\ &\quad + \frac{1}{B^2} \sum_{n_1 \neq n_2} \mathbb{E}\left\{ \mathbb{I}_{n_1} \left(\frac{1}{Np_{n_1}} \boldsymbol{x}_{n_1} - \overline{x} \right) \mathbb{I}_{n_2} \left(\frac{1}{Np_{n_2}} \boldsymbol{x}_{n_2} - \overline{x} \right) \right\}. \end{aligned} \tag{2.89}$$

We begin with:

$$\begin{aligned} \mathbb{E}\left\| \mathbb{I}_n \left(\frac{1}{Np_n} \boldsymbol{x}_n - \overline{x} \right) \right\|^2 &= \mathbb{E}\left\{ \left\| \mathbb{I}_n \left(\frac{1}{Np_n} \boldsymbol{x}_n - \overline{x} \right) \right\|^2 \middle| \mathbb{I}_n = 1 \right\} \mathbb{P}\left(\mathbb{I}_n = 1\right) \\ &\quad + \mathbb{E}\left\{ \left\| \mathbb{I}_n \left(\frac{1}{Np_n} \boldsymbol{x}_n - \overline{x} \right) \right\|^2 \middle| \mathbb{I}_n = 0 \right\} \mathbb{P}\left(\mathbb{I}_n = 0\right) \\ &= Bp_n \left(\mathbb{E}\left\| \frac{1}{Np_n} \boldsymbol{x}_n - \frac{1}{Np_n} \overline{x}_n + \frac{1}{Np_n} \overline{x}_n - \overline{x} \right\|^2 \right) \\ &= Bp_n \left(\frac{1}{N^2 p_n^2} \mathbb{E}\|\boldsymbol{x}_n - \overline{x}_n\|^2 + \left\| \frac{1}{Np_n} \overline{x}_n - \overline{x} \right\|^2 \right) \\ &= Bp_n \left(\frac{1}{N^2 p_n^2} \sigma_n^2 + \left\| \frac{1}{Np_n} \overline{x}_n - \overline{x} \right\|^2 \right). \end{aligned} \tag{2.90}$$

For the cross-term we have:

$$\begin{aligned}
&\mathbb{E}\left\{\mathbb{I}_{n_1}\left(\frac{1}{Np_{n_1}}\boldsymbol{x}_{n_1}-\overline{x}\right)\mathbb{I}_{n_2}\left(\frac{1}{Np_{n_2}}\boldsymbol{x}_{n_2}-\overline{x}\right)\right\}\\
&=\ \mathbb{E}\left\{\left(\frac{1}{Np_{n_1}}\boldsymbol{x}_{n_1}-\overline{x}\right)\left(\frac{1}{Np_{n_2}}\boldsymbol{x}_{n_2}-\overline{x}\right)\middle|\mathbb{I}_{n_1}=1,\mathbb{I}_{n_2}=1\right\}\mathbb{P}\left(\mathbb{I}_{n_1}=1,\mathbb{I}_{n_2}=1\right)\\
&=\ \mathbb{P}\left(\mathbb{I}_{n_2}=1,\mathbb{I}_{n_1}=1\right)\left(\frac{1}{Np_{n_1}}\mathbb{E}\boldsymbol{x}_{n_1}-\overline{x}\right)\left(\frac{1}{Np_{n_2}}\mathbb{E}\boldsymbol{x}_{n_2}-\overline{x}\right)\\
&=\ \mathbb{P}\left(\mathbb{I}_{n_2}=1,\mathbb{I}_{n_1}=1\right)\left(\frac{1}{Np_{n_1}}\overline{x}_{n_1}-\overline{x}\right)\left(\frac{1}{Np_{n_2}}\overline{x}_{n_2}-\overline{x}\right). \qquad (2.91)
\end{aligned}$$

We then get the desired result. □

2.B Proof of Lemma 2.1

We start with the sampling *with-replacement* construction. We have K agents from which we sample L. Thus, N and B in the previous Appendix 2.A are K and L, respectively. Let also:

$$\boldsymbol{x}_k=\widehat{\nabla_{w^\top}J_k}(\boldsymbol{w}_{i-1}), \qquad (2.92)$$

$$\overline{x}_k=\nabla_{w^\top}J_k(\boldsymbol{w}_{i-1}), \qquad (2.93)$$

$$\overline{x}=\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\top}J_k(\boldsymbol{w}_{i-1}). \qquad (2.94)$$

Then σ_k^2, which quantifies the second order moment of the local gradient noise, becomes:

$$\begin{aligned}
\sigma_k^2&=\mathbb{E}\left\{\left\|\widehat{\nabla_{w^\top}J_k}(\boldsymbol{w}_{i-1})-\nabla_{w^\top}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\
&=\frac{1}{E_k^2B_k^2}\sum_{e=1}^{E_k}\sum_{b\in\mathcal{B}_{k,i,e}}\mathbb{E}\left\{\left\|\frac{1}{N_kp_b^{(k)}}\nabla_{w^\top}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b})-\nabla_{w^\top}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\
&=\frac{1}{E_kB_k^2}\sum_{b\in\mathcal{B}_{k,i,e}}\mathbb{E}\left\{\left\|\frac{1}{N_kp_b^{(k)}}\nabla_{w^\top}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{k,b})\frac{1}{N_kp_b^{(k)}}\nabla_{w^\top}Q_k(w^o;\boldsymbol{x}_{k,b})\right.\right.\\
&\qquad\left.\left.+\frac{1}{N_kp_b^{(k)}}\nabla_{w^\top}Q_k(w^o;\boldsymbol{x}_{k,b})-\nabla_{w^\top}J_k(w^o)+\nabla_{w^\top}J_k(w^o)-\nabla_{w^\top}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\leq} \frac{3}{E_k B_k^2} \sum_{b\in\mathcal{B}_{k,i,e}} \Bigg\{ \sum_{n=1}^{N_k} p_n^{(k)} \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{i-1}; x_{k,n}) - \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(w^o; x_{k,n}) \right\|^2 \\
&\quad + \sum_{n=1}^{N_k} p_n^{(k)} \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(w^o; x_{k,n}) - \nabla_{w^\mathsf{T}} J_k(w^o) \right\|^2 \\
&\quad + \|\nabla_{w^\mathsf{T}} J_k(w^o) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1})\|^2 \Bigg\} \\
&\overset{(b)}{\leq} \frac{3}{E_k B_k^2} \sum_{b\in\mathcal{B}_{k,i,e}} \Bigg\{ \left(1 + \sum_{n=1}^{N_k} \frac{1}{N_k^2 p_n^{(k)}}\right) \delta^2 \|\widetilde{\boldsymbol{w}}_{i-1}\|^2 \\
&\quad + \sum_{n=1}^{N_k} p_n^{(k)} \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(w^o; x_{k,n}) - \nabla_{w^\mathsf{T}} J_k(w^o) \right\|^2 \Bigg\} \\
&\overset{(c)}{\leq} \frac{3\delta^2}{E_k B_k} \left(1 + \frac{1}{N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}}\right) \|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \frac{6}{E_k B_k N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}} \|\nabla_{w^\mathsf{T}} Q_k(w^o; x_{k,n})\|^2 \\
&\quad + \frac{6}{E_k B_k} \|\nabla_{w^\mathsf{T}} J_k(w^o)\|^2 \\
&= \beta_{s,k}^2 \|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_{s,k}^2 + \frac{6}{E_k B_k} \|\nabla_{w^\mathsf{T}} J_k(w^o)\|^2 ,
\end{aligned}
\tag{2.95}
$$

where (a) and (c) follow from using Jensen's inequality, and (b) follows from using the $\delta-$Lipschitz property of the gradients. Thus, using Appendix 2.A, we bound the stochastic noise variance as follows:

$$
\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2 | \boldsymbol{w}_{i-1}\right\} = \frac{1}{L} \sum_{k=1}^{K} p_k \left\{ \frac{1}{K^2 p_k^2} \sigma_k^2 + \left\| \frac{1}{K p_k} \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) - \frac{1}{K} \sum_{\ell=1}^{K} \nabla_{w^\mathsf{T}} J_\ell(\boldsymbol{w}_{i-1}) \right\|^2 \right\}. \tag{2.96}
$$

We focus on the second term since the first term has already been bounded. Using first Jensen's inequality to split into three terms and then Lipschitz condition of the gradients, we get:

$$
\begin{aligned}
&\left\| \frac{1}{K p_k} \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) - \frac{1}{K} \sum_{\ell=1}^{K} \nabla_{w^\mathsf{T}} J_\ell(\boldsymbol{w}_{i-1}) \right\|^2 \\
&= \frac{1}{K^2} \Bigg\| \frac{1}{p_k} \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) - \frac{1}{p_k} \nabla_{w^\mathsf{T}} J_k(w^o) + \frac{1}{p_k} \nabla_{w^\mathsf{T}} J_k(w^o) + \sum_{\ell=1}^{K} \nabla_{w^\mathsf{T}} J_\ell(w^o) \\
&\quad - \sum_{\ell=1}^{K} \nabla_{w^\mathsf{T}} J_\ell(\boldsymbol{w}_{i-1}) \Bigg\|^2 \\
&\leq 3\delta^2 \left(1 + \frac{1}{K^2 p_k^2}\right) \|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \frac{3}{K^2 p_k^2} \|\nabla_{w^\mathsf{T}} J_k(w^o)\|^2 .
\end{aligned}
\tag{2.97}
$$

Then, putting things together, we get:

$$\begin{aligned}\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2|\boldsymbol{w}_{i-1}\right\} &\leq \left(\frac{3\delta^2}{L}+\frac{1}{LK^2}\sum_{k=1}^{K}\frac{1}{p_k}\left(\beta_{s,k}^2+3\delta^2\right)\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\\ &\quad+\frac{1}{LK^2}\sum_{k=1}^{K}\frac{1}{p_k}\left\{\sigma_{s,k}^2+\left(3+\frac{6}{E_kB_k}\right)\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2\right\}\\ &=\beta_s^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_s^2. \end{aligned} \tag{2.98}$$

Next, we move to the sampling *without replacement* construction. The variance σ_k^2 becomes:

$$\begin{aligned}\sigma_k^2 &= \mathbb{E}\left\{\left\|\widehat{\nabla_{w^\mathsf{T}}J_k}(\boldsymbol{w}_{i-1})-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\ &=\mathbb{E}\left\{\left\|\frac{1}{E_kB_k}\sum_{e=1}^{E_k}\sum_{n=1}^{N_k}\mathbb{I}_n\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_n)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\ &=\frac{1}{E_kB_k^2}\sum_{n=1}^{N_k}\mathbb{E}\left\{\left\|\mathbb{I}_n\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_n)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\ &\quad+\frac{1}{E_kB_k^2}\sum_{n_1\neq n_2}\mathbb{E}\left\{\mathbb{I}_{n_1}\left(\frac{1}{N_kp_{n_1}^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{n_1})-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right)\right.\\ &\quad\left.\times\,\mathbb{I}_{n_2}\left(\frac{1}{N_kp_{n_2}^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};\boldsymbol{x}_{n_2})-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right)\middle|\boldsymbol{w}_{i-1}\right\}. \end{aligned} \tag{2.99}$$

Starting with the first term, we use Jensen's inequality in (a) and (c) and the Lipschitz condition in (b) to get:

$$\begin{aligned}&\frac{1}{E_kB_k^2}\sum_{n=1}^{N_k}\mathbb{P}(\mathbb{I}_n=1)\mathbb{E}\left\{\left\|\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};x_{k,n})-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1},\mathbb{I}_n=1\right\}\\ &=\frac{1}{E_kB_k}\sum_{n=1}^{N_k}p_n^{(k)}\left\|\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};x_{k,n})-\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})\right.\\ &\quad\left.+\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})-\nabla_{w^\mathsf{T}}J_k(w^o)+\nabla_{w^\mathsf{T}}J_k(w^o)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\le} \frac{3}{E_kB_k}\sum_{n=1}^{N_k}\left\{\frac{1}{N_k^2 p_n^{(k)}}\|\nabla_{w^\mathsf{T}}Q_k(\boldsymbol{w}_{i-1};x_{k,n})-\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})\|^2\right.\\
&\quad\left.+p_n^{(k)}\left\|\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})-\nabla_{w^\mathsf{T}}J_k(w^o)\right\|^2\right\}\\
&\quad+\frac{3}{E_kB_k}\left\|\nabla_{w^\mathsf{T}}J_k(w^o)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2,\\
&\overset{(b)}{\le} \frac{3\delta^2}{E_kB_k}\left(1+\frac{1}{N_k^2}\sum_{n=1}^{N_k}\frac{1}{p_n^{(k)}}\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\\
&\quad+\frac{3}{E_kB_k}\sum_{n=1}^{N_k}p_n^{(k)}\left\|\frac{1}{N_kp_n^{(k)}}\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})-\nabla_{w^\mathsf{T}}J_k(w^o)\right\|^2\\
&\overset{(c)}{\le} \frac{3\delta^2}{E_kB_k}\left(1+\frac{1}{N_k^2}\sum_{n=1}^{N_k}\frac{1}{p_n^{(k)}}\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\\
&\quad+\frac{6}{E_kB_k}\sum_{n=1}^{N_k}\frac{1}{N_k^2p_n^{(k)}}\|\nabla_{w^\mathsf{T}}Q_k(w^o;x_{k,n})\|^2+\frac{6}{E_kB_k}\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2\\
&=\beta_{s,k}^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_{s,k}^2+\frac{6}{E_kB_k}\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2,
\end{aligned}
\tag{2.100}
$$

The cross-term reduces to 0 by first conditioning over $\mathbb{I}_{n_1}=1,\mathbb{I}_{n_2}=1$ and then splitting the expectation. Each of the two terms are zero. Thus, putting everything together, we get:

$$
\sigma_k^2\le\beta_{s,k}^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_{s,k}^2+\frac{6}{E_kB_k}\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2. \tag{2.101}
$$

Next, to bound the second order moment of the gradient noise, we use (2.81):

$$
\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2|\boldsymbol{w}_{i-1}\right\}\le\sum_{k=1}^K p_k\left(\frac{1}{K^2p_k^2}\sigma_k^2+\left\|\frac{1}{Kp_k}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})-\frac{1}{K}\sum_{\ell=1}^K\nabla_{w^\mathsf{T}}J_\ell(\boldsymbol{w}_{i-1})\right\|^2\right). \tag{2.102}
$$

The second term is of the same form as for sampling with replacement, and thus can be bounded similarly:

$$
\begin{aligned}
\mathbb{E}\{\|\boldsymbol{s}_i\|^2|\boldsymbol{w}_{i-1}\}&\le\sum_{k=1}^K p_k\left\{\frac{\beta_{s,k}^2}{K^2p_k^2}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\frac{1}{K^2p_k^2}\sigma_{s,k}^2+\frac{6}{K^2p_k^2E_kB_k}\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2\right.\\
&\quad\left.+\frac{3}{K^2p_k^2}\|\nabla_{w^\mathsf{T}}J_k(w^o)\|^2+3\delta^2\left(1+\frac{1}{K^2p_k^2}\right)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right\}\\
&=\beta_s^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\sigma_s^2.
\end{aligned}
\tag{2.103}
$$

2.C Proof of Lemma 2.2

We first note the following result by using $\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(w^o)=0$:

$$\begin{aligned}
&\left\|\widetilde{\boldsymbol{w}}_{i-1}+\mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\\
&=\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\mu^2\left\|\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}P_k(w^o)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2+2\mu\widetilde{\boldsymbol{w}}_{i-1}^\mathsf{T}\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\\
&\overset{(a)}{\leq}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\mu^2\frac{1}{K}\sum_{k=1}^{K}\|\nabla_{w^\mathsf{T}}J_k(w^o)-\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\|^2+2\mu\widetilde{\boldsymbol{w}}_{i-1}^\mathsf{T}\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\\
&\overset{(b)}{\leq}(1+\mu^2\delta^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+2\mu\widetilde{\boldsymbol{w}}_{i-1}^\mathsf{T}\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\\
&\overset{(c)}{\leq}(1+\mu^2\delta^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+2\mu\frac{1}{K}\sum_{k=1}^{K}\left(J_k(w^o)-J_k(\boldsymbol{w}_{i-1})-\frac{\nu}{2}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\right)\\
&\overset{(d)}{\leq}(1+\mu^2\delta^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2-2\mu\frac{1}{K}\sum_{k=1}^{K}\nu\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\\
&=(1-2\mu\nu+\mu^2\delta^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2,
\end{aligned}\tag{2.104}$$

where (a) follows from Jensen's inequality, (b) from the Lipschitz condition, and (c) and (d) from strong convexity.

Returning to the main expression:

$$\widetilde{\boldsymbol{w}}_{i-1}+\mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})+\mu\boldsymbol{s}_i,\tag{2.105}$$

and taking conditional expectations, we obtain:

$$\begin{aligned}
&\mathbb{E}\left\{\left\|\widetilde{\boldsymbol{w}}_{i-1}+\mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})+\mu\boldsymbol{s}_i\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\
&\overset{(a)}{=}\mathbb{E}\left\{\left\|\widetilde{\boldsymbol{w}}_{i-1}+\mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}}J_k(\boldsymbol{w}_{i-1})\right\|^2\middle|\boldsymbol{w}_{i-1}\right\}+\mu^2\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2\middle|\boldsymbol{w}_{i-1}\right\}\\
&\overset{(b)}{\leq}(1-2\mu\nu+\mu^2\delta^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\mu^2\left(\beta_s^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\eta_s\|\widetilde{\boldsymbol{w}}_{i-1}\|+\sigma_s^2\right),
\end{aligned}\tag{2.106}$$

where the cross-term in (a) is zero because of the zero mean property of the gradient noise, and (b) follows from (2.104) and using the bound on the second order moment of the gradient noise.

Next, taking expectation again to remove the conditioning we get:

$$\mathbb{E}\left\|\widetilde{\boldsymbol{w}}_{i-1}+\mu\frac{1}{K}\sum_{k=1}^{K}\nabla_{w^{\mathsf{T}}}J_k(\boldsymbol{w}_{i-1})+\mu\boldsymbol{s}_i\right\|^2 \leq \left(1-2\mu\nu+\mu^2(\delta^2+\beta_s^2)\right)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \mu^2\eta_s\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|+\mu^2\sigma_s^2. \tag{2.107}$$

2.D Proof of Lemma 2.3

To show the mean is zero, it is enough to calculate the mean of the approximate gradient. We start with the sampling with replacement scheme where the samples are chosen independently from each other:

$$\begin{aligned}
&\mathbb{E}\left\{\frac{1}{B_k}\sum_{b\in\mathcal{B}_{k,i,e}}\frac{1}{N_kp_b^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,b})\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\}\\
&=\frac{1}{B_k}\sum_{b\in\mathcal{B}_{k,i,e}}\mathbb{E}\left\{\frac{1}{N_kp_b^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,b})\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\}\\
&=\frac{1}{N_k}\sum_{n=1}^{N_k}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,n}).
\end{aligned}\tag{2.108}$$

As for the sampling without replacement scheme, since the samples are now dependent, we introduce the indicator function $\mathbb{I}_n$ and the derivation goes as follows:

$$\begin{aligned}
&\mathbb{E}\left\{\frac{1}{B_k}\sum_{b\in\mathcal{B}_{k,i,e}}\frac{1}{N_kp_b^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,b})\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\}\\
&=\mathbb{E}\left\{\frac{1}{B_k}\sum_{n=1}^{N_k}\frac{\mathbb{I}_n}{N_kp_n^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,n})\middle|\mathcal{F}_{e-1},\mathcal{L}_i\right\}\\
&=\frac{1}{B_k}\sum_{n=1}^{N_k}\frac{\mathbb{P}(\mathbb{I}_n=1)}{N_kp_n^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,n})\\
&=\frac{1}{N_k}\sum_{n=1}^{N_k}\nabla_{w^{\mathsf{T}}}Q_k(\boldsymbol{w}_{k,e-1};\boldsymbol{x}_{k,n}).
\end{aligned}\tag{2.109}$$

Next, to bound the second order moment, we start with an intermediate step and bound the second order moment of the individual gradient noise of one sample. The derivation below holds regardless of the sampling scheme. By adding and subtracting $\frac{1}{N_kp_n^{(k)}}\nabla_{w^{\mathsf{T}}}Q_k(w_k^o;\boldsymbol{x}_{k,n})$, adding $\nabla_{w^{\mathsf{T}}}J_k(w_k^o)=0$, and then using Jensen's inequality and

Lipschitz condition, we get:

$$\begin{aligned}
&\left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,n}) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \right\|^2 \\
&\leq 3 \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,n}) - \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(w_k^o; \boldsymbol{x}_{k,n}) \right\|^2 \\
&\quad + 3 \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(w_k^o; \boldsymbol{x}_{k,n}) \right\|^2 + 3\delta^2 \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2.
\end{aligned} \tag{2.110}$$

Then, taking the conditional expectation and using the Lipschitz property, we get:

$$\begin{aligned}
&\mathbb{E}\left\{ \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,n}) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \right\|^2 \middle| \mathcal{F}_{e-1}, \mathcal{L}_i \right\} \\
&\leq \sum_{n=1}^{N_k} \frac{3 p_n^{(k)}}{N_k^2 \left(p_n^{(k)}\right)^2} \left(\|\nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; x_{k,n}) - \nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2 + \|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2 \right) \\
&\quad + 3\delta^2 \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 \\
&\leq \sum_{n=1}^{N_k} \frac{3}{N_k^2 p_n^{(k)}} \left(\delta^2 \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2 \right) + 3\delta^2 \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2.
\end{aligned} \tag{2.111}$$

Now going back to calculating the second order moment of the local incremental gradient noise, we first start with the sampling with replacement. Using the fact that the samples are independent we get:

$$\begin{aligned}
&\mathbb{E}\left\{ \|\boldsymbol{q}_{k,i,e}\|^2 \middle| \mathcal{F}_{e-1}, \mathcal{L}_i \right\} \\
&= \frac{1}{K^2 p_k^2 B_k^2} \sum_{b \in \mathcal{B}_{k,i,e}} \mathbb{E}\left\{ \left\| \frac{1}{N_k p_b^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,b}) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \right\|^2 \middle| \mathcal{F}_{e-1}, \mathcal{L}_i \right\} \\
&\leq \frac{3\delta^2}{K^2 p_k^2 B_k} \left(1 + \frac{1}{N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}} \right) \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \frac{3}{K^2 p_k^2 B_k N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}} \|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2.
\end{aligned} \tag{2.112}$$

As for the sampling without replacement, we also introduce the indicator function and write out the square of sums. The cross-terms disappear since each term has zero mean. The derivation then follows similarly to that of the sampling with replacement. More formally:

$$
\begin{aligned}
&\mathbb{E}\left\{\|\boldsymbol{q}_{k,i,e}\|^2 \middle| \mathcal{F}_{e-1}, \mathcal{L}_i\right\} \\
&= \frac{1}{K^2 p_k^2 B_k^2} \sum_{n=1}^{N_k} \mathbb{P}(\mathbb{I}_n = 1) \\
&\quad \times \mathbb{E}\left\{\left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; \boldsymbol{x}_{k,n}) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \right\|^2 \middle| \mathbb{I}_n = 1, \mathcal{F}_{e-1}, \mathcal{L}_i \right\} \\
&= \frac{1}{K^2 p_k^2 B_k} \sum_{n=1}^{N_k} p_n^{(k)} \left\| \frac{1}{N_k p_n^{(k)}} \nabla_{w^\mathsf{T}} Q_k(\boldsymbol{w}_{k,e-1}; x_{k,n}) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \right\|^2 \\
&\leq \frac{3\delta^2}{K^2 p_k^2 B_k} \left(1 + \frac{1}{N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}}\right) \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \frac{3}{K^2 p_k^2 B_k N_k^2} \sum_{n=1}^{N_k} \frac{1}{p_n^{(k)}} \|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2.
\end{aligned}
\tag{2.113}
$$

2.E Proof of Lemma 2.4

We subtract w_k^o from both sides of (2.10) and use (2.29) to get:

$$
\widetilde{\boldsymbol{w}}_{k,e} = \widetilde{\boldsymbol{w}}_{k,e-1} + \mu \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) + \mu \boldsymbol{q}_{k,i,e}. \tag{2.114}
$$

We bound the first two terms and use the fact that $\nabla_{w^\mathsf{T}} P_k(w_k^o) = 0$, Lipschitz condition, and the convexity of the cost function:

$$
\begin{aligned}
&\|\widetilde{\boldsymbol{w}}_{k,e-1} + \mu \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1})\|^2 \\
&= \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + 2\mu \widetilde{\boldsymbol{w}}_{k,e-1}^\mathsf{T} \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) + \mu^2 \|\nabla_{w^\mathsf{T}} J_k(w_k^o) - \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1})\|^2 \\
&\leq (1 + \mu^2 \delta^2) \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + 2\mu \widetilde{\boldsymbol{w}}_{k,e-1}^\mathsf{T} \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1}) \\
&\leq (1 - 2\nu\mu + \mu^2 \delta^2) \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2.
\end{aligned}
\tag{2.115}
$$

Returning to (2.114), squaring both sides, conditioning on the filtration $\mathcal{F}_{e-1}$, and taking expectations we obtain:

$$
\begin{aligned}
\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{k,e}\|^2 \middle| \mathcal{F}_{e-1}\right\} &\overset{(a)}{=} \mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{k,e-1} + \mu \nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{k,e-1})\|^2 \middle| \mathcal{F}_{e-1}\right\} + \mu^2 \mathbb{E}\left\{\|\boldsymbol{q}_{k,i,e}^2\|^2 \middle| \mathcal{F}_{e-1}\right\} \\
&\leq \left(1 - 2\nu\mu + \mu^2 \left(\delta^2 + \frac{E_k}{K^2 p_k^2} \beta_{s,k}^2\right)\right) \|\widetilde{\boldsymbol{w}}_{k,e-1}\|^2 + \mu^2 \frac{1}{K^2 p_k^2} \sigma_{q,k}^2,
\end{aligned}
\tag{2.116}
$$

where the cross term in (a) is zero because of the zero mean property of the local incremental gradient noise. Taking expectations on both sides again removes the condition on the filtration and leads to the desired result. By further iterating recursion (2.33) we obtain:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,e}\|^2 \leq \lambda_k^e \mathbb{E}\|\widetilde{\boldsymbol{w}}_{k,0}\|^2 + \frac{1-\lambda_k^e}{1-\lambda_k}\mu^2\sigma_{q,k}^2. \tag{2.117}$$

2.F Proof of Lemma 2.5

First, using Jensen's inequality (a) and Lipschitz continuity (b), we obtain:

$$\begin{aligned}
\|\boldsymbol{q}_i\|^2 &\overset{(a)}{\leq} \frac{1}{L}\sum_{\ell\in\mathcal{L}_i}\frac{1}{K^2p_\ell^2E_\ell B_\ell}\sum_{e=1}^{E_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{\|\nabla_{w^\mathsf{T}}Q_\ell(\boldsymbol{w}_{\ell,e-1};\boldsymbol{x}_{\ell,b})-\nabla_{w^\mathsf{T}}Q_\ell(\boldsymbol{w}_{i-1};\boldsymbol{x}_{\ell,b})\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\\
&\overset{(b)}{\leq} \frac{\delta^2}{L}\sum_{\ell\in\mathcal{L}_i}\frac{1}{K^2p_\ell^2E_\ell B_\ell}\sum_{e=1}^{E_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}.
\end{aligned} \tag{2.118}$$

Next, we focus on $\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2$, and by applying Jensen's inequality in (a) and (b) and Lipschitz condition in (c) we obtain:

$$\begin{aligned}
&\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2\\
&=\mu^2\left\|\frac{1}{E_\ell B_\ell}\sum_{f=0}^{e-2}\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{1}{N_\ell p_b^{(\ell)}}\nabla_{w^\mathsf{T}}Q_\ell(\boldsymbol{w}_{\ell,f};\boldsymbol{x}_{\ell,b})\right\|^2\\
&\overset{(a)}{\leq}\frac{\mu^2}{E_\ell B_\ell}\sum_{f=0}^{e-2}\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{1}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\|\nabla_{w^\mathsf{T}}Q_\ell(\boldsymbol{w}_{\ell,f};\boldsymbol{x}_{\ell,b})-\nabla_{w^\mathsf{T}}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})+\nabla_{w^\mathsf{T}}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})\|^2\\
&\overset{(b)}{\leq}\frac{2\mu^2}{E_\ell B_\ell}\sum_{f=0}^{e-2}\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{1}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\Big(\|\nabla_{w^\mathsf{T}}Q_\ell(\boldsymbol{w}_{\ell,f};\boldsymbol{x}_{\ell,b})-\nabla_{w^\mathsf{T}}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})\|^2\\
&\quad+\|\nabla_{w^\mathsf{T}}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})\|^2\Big)\\
&\overset{(c)}{\leq}\frac{2\mu^2}{E_\ell B_\ell}\sum_{f=0}^{e-2}\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{1}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\Big(\delta^2\|\widetilde{\boldsymbol{w}}_{\ell,f}\|^2+\|\nabla_{w^\mathsf{T}}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})\|^2\Big).
\end{aligned}$$

Then, taking the expectation given the previous filtration $\mathcal{F}_{e-2}$ and the participating agents $\mathcal{L}_i$, we see that:

$$\begin{aligned}
&\mathbb{E}\left\{\frac{1}{B_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{F}_{e-2},\mathcal{L}_i\right\}\\
&\le \mathbb{E}\left\{\frac{1}{B_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{1}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{F}_{e-2},\mathcal{L}_i\right\}\left(\frac{2\mu^2\delta^2}{E_\ell B_\ell}\sum_{f=0}^{e-2}\|\widetilde{\boldsymbol{w}}_{\ell,f}\|^2\mathbb{E}\left\{\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{1}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{F}_{e-2},\mathcal{L}_i\right\}\right.\\
&\quad\left.+\frac{2\mu^2}{E_\ell B_\ell}\sum_{f=0}^{e-2}\mathbb{E}\left\{\sum_{b\in\mathcal{B}_{\ell,i,f}}\frac{\|\nabla_{w^\intercal}Q_\ell(w_\ell^o;\boldsymbol{x}_{\ell,b})\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{F}_{e-2},\mathcal{L}_i\right\}\right)\\
&=\left(\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\right)\left(\frac{2\mu^2\delta^2}{E_\ell}\sum_{f=0}^{e-2}\|\widetilde{\boldsymbol{w}}_{\ell,f}\|^2\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}+\frac{2\mu^2(e-1)}{3E_\ell}\sigma_{q,\ell}^2\right).
\end{aligned}\tag{2.119}$$

Then, taking expectation again over the filtration, we obtain:

$$\begin{aligned}
&\mathbb{E}\left\{\frac{1}{B_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{L}_i\right\}\\
&\le\frac{1}{B_\ell}\left(\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\right)^2\frac{2\mu^2\delta^2}{E_\ell}\sum_{f=0}^{e-1}\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{\ell,f}\|^2\middle|\mathcal{L}_i\right\}+\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\frac{2\mu^2(e-1)}{3E_\ell}\sigma_{q,\ell}^2\\
&\overset{(a)}{\le}\frac{2\mu^2\delta^2}{E_\ell B_\ell}\left(\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\right)^2\sum_{f=0}^{e-1}\left(\lambda_\ell^f\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{\ell,0}\|^2\middle|\mathcal{L}_i\right\}+\frac{\mu^2}{K^2p_\ell^2}\frac{1-\lambda_\ell^f}{1-\lambda_\ell}\sigma_{q,\ell}^2\right)\\
&\quad+\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\frac{2\mu^2(e-1)}{3E_\ell}\sigma_{q,\ell}^2\\
&=\frac{2\mu^2\delta^2}{E_\ell B_\ell}\left(\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\right)^2\left(\frac{1-\lambda_\ell^e}{1-\lambda_\ell}\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{\ell,0}\|^2\middle|\mathcal{L}_i\right\}+\frac{\mu^2}{K^2p_k^2}\frac{e(1-\lambda_\ell)-1+\lambda_\ell^e}{(1-\lambda_\ell)^2}\sigma_{q,\ell}^2\right)\\
&\quad+\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\frac{2\mu^2(e-1)}{3E_\ell}\sigma_{q,\ell}^2\\
&\overset{(b)}{\le}\frac{2\mu^2\delta^2}{E_\ell B_\ell}\left(\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\right)^2\left(2\frac{1-\lambda_\ell^e}{1-\lambda_\ell}\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\middle|\mathcal{L}_i\right\}+2\frac{1-\lambda_\ell^e}{1-\lambda_\ell}\mathbb{E}\left\{\|w^o-w_\ell^o\|^2\middle|\mathcal{L}_i\right\}\right.\\
&\quad\left.+\frac{\mu^2}{K^2p_k^2}\frac{e(1-\lambda_\ell)-1+\lambda_\ell^e}{(1-\lambda_\ell)^2}\sigma_{q,\ell}^2\right)+\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2p_n^{(\ell)}}\frac{2\mu^2(e-1)}{3E_\ell}\sigma_{q,\ell}^2,
\end{aligned}\tag{2.120}$$

where we used Lemma 2.4 in (a), and in (b) we added and subtracted w^o and used Jensen's inequality. Then, summing over e results in:

$$\begin{aligned}
&\frac{1}{E_\ell}\sum_{e=1}^{E_\ell}\mathbb{E}\left\{\frac{1}{B_\ell}\sum_{b\in\mathcal{B}_{\ell,i,e}}\frac{\|\boldsymbol{w}_{i-1}-\boldsymbol{w}_{\ell,e-1}\|^2}{N_\ell^2\left(p_b^{(\ell)}\right)^2}\middle|\mathcal{L}_i\right\}\\
&\leq \frac{2\mu^2\delta^2}{E_\ell^2 B_\ell}\left(\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2 p_n^{(\ell)}}\right)^2\left(2\frac{(E_\ell+1)(1-\lambda_\ell)-1+\lambda_\ell^{E_\ell+1}}{(1-\lambda_\ell)^2}\left(\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\middle|\mathcal{L}_i\right\}\right.\right.\\
&\quad\left.\left.+\mathbb{E}\left\{\|w^o-w_\ell^o\|^2\middle|\mathcal{L}_i\right\}\right)+\frac{E_\ell(E_\ell+1)(1-\lambda_\ell)^2-2E_\ell(1-\lambda_\ell)+2\lambda_\ell-2\lambda_\ell^{E_\ell+1}}{(1-\lambda_\ell)^3}\frac{\mu^2}{K^2p_k^2}\sigma_{q,\ell}^2\right)\\
&\quad+\frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2 p_n^{(\ell)}}\frac{E_\ell(E_\ell-1)\mu^2}{3E_\ell^2}\sigma_{q,\ell}^2. \qquad (2.121)
\end{aligned}$$

Taking the expectation of (2.118) given the choice of the agents and plugging the above expression, we get:

$$\begin{aligned}
\mathbb{E}\left\{\|\boldsymbol{q}_i\|^2\middle|\mathcal{L}_i\right\}\leq\frac{\delta^2}{L}\sum_{\ell\in\mathcal{L}_i}\frac{1}{K^2p_\ell^2}\Big(&a\mu^2\mathbb{E}\left\{\|\widetilde{\boldsymbol{w}}_{i-1}\|^2\middle|\mathcal{L}_i\right\}+a\mu^2\mathbb{E}\left\{\|w^o-w_\ell^o\|^2\middle|\mathcal{L}_i\right\}\\
&+(b\mu^4+c\mu^2)\sigma_{q,\ell}^2\Big), \qquad (2.122)
\end{aligned}$$

where we introduced constants a, b, c to make the notation simpler:

$$a \triangleq \frac{4}{E_\ell^2 B_\ell}\left(\sum_{n=1}^{N}\frac{1}{N_\ell^2 p_n^{(\ell)}}\right)^2\frac{(E_\ell+1)(1-\lambda_\ell)-1+\lambda_\ell^{E_\ell+1}}{(1-\lambda_\ell)^2}, \qquad (2.123)$$

$$b \triangleq \frac{2}{E_\ell^2 B_\ell}\left(\sum_{n=1}^{N}\frac{1}{N_\ell^2 p_n^{(\ell)}}\right)^2\frac{E_\ell(E_\ell+1)(1-\lambda_\ell)^2-2E_\ell(1-\lambda_\ell)+2\lambda_\ell-2\lambda_\ell^{E_\ell+1}}{(1-\lambda_\ell)^3}, \qquad (2.124)$$

$$c \triangleq \frac{1}{B_\ell}\sum_{n=1}^{N_\ell}\frac{1}{N_\ell^2 p_n^{(\ell)}}\frac{E_\ell-1}{3E_\ell}. \qquad (2.125)$$

Then, taking again the expectation to remove the conditioning and using Assumption 2.2:

$$\mathbb{E}\|\boldsymbol{q}_i\|^2\leq\delta^2\sum_{k=1}^{K}\frac{1}{K^2p_k}\sum_{n=1}^{N_k}\frac{1}{N_k^2p_n^{(k)}}\left(a\mu^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+a\mu^2\xi^2+(b\mu^4+c\mu^2)\sigma_{q,k}^2\right). \qquad (2.126)$$

Further simplifying the notation gives us the desired result. Thus, since $a=O(\mu^{-1})$, $b=O(\mu^{-2})$ and $c=O(1)$, we get $\mathbb{E}\|\boldsymbol{q}_i\|^2=O(\mu)$.

2.G Proof of Theorem 2.2

We start by writing the error recursion:

$$\widetilde{\boldsymbol{w}}_i = \widetilde{\boldsymbol{w}}_{i-1} + \frac{\mu}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) + \mu\boldsymbol{s}_i + \mu\boldsymbol{q}_i + \frac{\mu}{L}\sum_{k\in\mathcal{L}_i}\boldsymbol{g}_{k,i}. \tag{2.127}$$

We have already shown that the gradient noise is zero-mean and has bounded second order-moment Lemma 2.1, while the incremental noise has bounded second order-moment Lemma 2.5:

$$\mathbb{E}\{\|\boldsymbol{s}_i\|^2|\mathcal{F}_{i-1}\} \leq \beta_s^2\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + \sigma_s^2, \tag{2.128}$$

$$\mathbb{E}\|\boldsymbol{q}_i\|^2 \leq O(\mu)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2 + O(\mu)\xi^2 + O(\mu^2)\sigma_q^2, \tag{2.129}$$

where the constants $\beta_s^2, \sigma_s^2, \sigma_q^2$ are given by:

$$\beta_s^2 = \frac{6\delta^2}{L}\left(1 + \frac{1}{K}\sum_{k=1}^{K}\frac{1}{E_k}\right), \tag{2.130}$$

$$\sigma_s^2 = \frac{1}{LK}\sum_{k=1}^{K}\left(\frac{12}{E_k}+3\right)\frac{1}{N_k}\sum_{n=1}^{N_k}\|\nabla_{w^\mathsf{T}} Q_k(w^o; x_{k,n})\|^2, \tag{2.131}$$

$$\sigma_q^2 = \frac{3}{K}\sum_{k=1}^{K}\sum_{n=1}^{N_k}\|\nabla_{w^\mathsf{T}} Q_k(w_k^o; x_{k,n})\|^2. \tag{2.132}$$

Taking the conditional mean of the ℓ_2-norm of the error, we can split the noise term from the rest and then apply Jensen's inequality with some constant $\alpha \in (0,1)$:

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_i\|^2|\mathcal{F}_{i-1},\mathcal{L}_i\} &= \mathbb{E}\left\{\left\|\widetilde{\boldsymbol{w}}_{i-1} + \frac{\mu}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1}) + \mu\boldsymbol{s}_i + \mu\boldsymbol{q}_i\right\|^2\middle|\mathcal{F}_{i-1},\mathcal{L}_i\right\} \\ &\quad + \frac{\mu^2}{L^2}\sum_{k\in\mathcal{L}_i 1}\mathbb{E}\|\boldsymbol{g}_{k,i}\|^2 \\ &\leq \frac{1}{\alpha}\left\|\widetilde{\boldsymbol{w}}_{i-1} + \frac{\mu}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1})\right\|^2 + \frac{\mu^2}{\alpha}\mathbb{E}\{\|\boldsymbol{s}_i\|^2|\mathcal{F}_{i-1},\mathcal{L}_i\} + \frac{\mu^2}{L}\sigma_g^2 \\ &\quad + \frac{\mu^2}{1-\alpha}\mathbb{E}\{\|\boldsymbol{q}_i\|^2|\mathcal{F}_{i-1},\mathcal{L}_i\}.\end{aligned} \tag{2.133}$$

Using strong convexity and Lipschitz continuity of the functions we can bound the first term as:

$$\left\|\widetilde{\boldsymbol{w}}_{i-1} + \frac{\mu}{K}\sum_{k=1}^{K}\nabla_{w^\mathsf{T}} J_k(\boldsymbol{w}_{i-1})\right\|^2 \leq (1-2\nu\mu+\delta^2\mu^2)\|\widetilde{\boldsymbol{w}}_{i-1}\|^2. \tag{2.134}$$

Then, taking the expectations again over the past models and the selected agents, and using the bound on the gradient noise and incremental noise:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \left(\frac{1-2\nu\mu+(\beta_{s,1}^2+\delta^2)\mu^2}{\alpha}+\frac{O(\mu^3)}{1-\alpha}\right)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2+\frac{\mu^2}{\alpha}\sigma_s^2+\frac{\mu^2}{L}\sigma_g^2. \\ +\frac{O(\mu^3)\xi^2+O(\mu^4)\sigma_q^2}{1-\alpha} \tag{2.135}$$

Then, recursively bounding the error with $\alpha=\sqrt{1-2\nu\mu+(\beta_s^2+\delta^2)\mu^2}$:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \lambda^i\mathbb{E}\|\widetilde{\boldsymbol{w}}_{1,0}\|^2+\frac{1-\lambda^i}{1-\lambda}\left(O(\mu^2)\sigma_s^2+O(\mu^2)\xi^2+\frac{\mu^2}{L}\sigma_g^2+O(\mu^3)\sigma_q^2\right), \tag{2.136}$$

and taking the limit of i:

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq O(\mu)(\sigma_s^2+\xi^2+\sigma_g^2)+O(\mu^2)\sigma_q^2. \tag{2.137}$$

3 Privatized Graph Federated Learning

In the previous chapter, we considered the federated learning setting and studied its convergence under importance sampling. We then perturbed the gradients to privatize the algorithm and showed that the added noise does not deteriorate performance. In this chapter, we adjust the original federated setting into a network of federated units. This new system is both distributed and decentralized; the network of servers is a distributed system, and each server with its clients forms a decentralized system. In this chapter, we drop importance sampling for simplicity and use standard uniform sampling. We focus on the privatization scheme between the servers and use the previously constructed privatization mechanism for the clients. The material in this chapter is based on the work in [81].

3.1 Introduction

Federated learning (FL) [11] is one particular distributed structure where training happens in collaboration between different clients and the server. Compared to a fully decentralized solution, communication occurs between the server and the clients (or agents), instead of directly between the agents themselves. Such a distributed architecture is not robust to communication failures and computational overloads, nor it is immune to privacy attacks when agents are required to share their local updates. In standard FL, millions of users can be connected to *one* server at a time. This means one server will need to be responsible for the communication with all clients with significant computational burden, thus rendering the system susceptible to communication failures. Furthermore, whether clients send their gradient updates or their local models, information about their data can be inferred from the exchanges and leaked [82–85]. Consider for instance the logistic risk; the gradient of the loss function is a constant multiple of the feature vector. Thus, even though the actual data samples are not sent to the server, information about them can still be inferred from the gradient updates or the models.

These considerations motivate us to propose a networked architecture for federated learning with privacy guarantees. In particular, we introduce the graph federated architecture, which consists of multiple servers, and we privatize the algorithm by ensuring the communication ocuring between the servers and the clients is secure. Graph homomorphic perturbations, which were initially introduced in [86], focus on the communication between servers. They are based on adding correlated noise to the messages sent between servers such that the noise cancels out if we were to take the average of all messages across all servers. As for the privatization between the clients and their servers, we share noisy updates as opposed to models. The two protocols make sure the effect of the added noise is reduced.

Other works have also contributed to addressing the same challenges we are considering in this work, albeit differently. For example, the work [87] introduces a hierarchical architecture, where it is assumed there are multiple servers connected in a tree structure. Such a solution still has one main server and faces the same robustness problem as FL. The graph federated learning architecture in this work (and which appeared in the earlier conference publication [88]) is a more general structure. While [89] has a similar architecture to the GFL architecture proposed earlier in [88], it nevertheless does not deal with privacy and employs different objective functions and a different learning algorithm based on the alternating direction method of multipliers. Likewise, a plethora of solutions exist that relate to privacy issues. These methods may be split into two sub-groups: those using random perturbations to ensure a certain level of differential privacy [46, 90–98], or those that rely on cryptographic methods [99–103]. Both have their advantages and disadvantages. While differential privacy is easy to implement, it hinders the performance of the algorithm by reducing the model utility. As for cryptographic methods, they are generally harder to implement since they require more computational and communication power [104, 105]. Furthremore, they restrict the number of participating users. Moving forward, we go ahead with the study of differentially private methods.

The main contribution in this chapter is three-fold. We introduce a new generalized and more realistic architecture for the federated setting where we now consider multiple servers connected by some graph structure. Furthermore, many earlier works have proposed adding Laplacian noise sources to the shared information among agents in order to ensure some level of privacy. However, these works have largely ignored the fact that these noises degrade the mean-square error (MSE) performance of the network from $O(\mu)$ down to $O(\mu^{-1})$, where μ is the small learning parameter. To resolve this issue, we define a new noise generation scheme that results in an $O(1)$ bound on the MSE while ensuring privacy. Although the work [98] proposed a noisy-distributed consensus strategy, this reference lacks a useful construction method for the perturbations. In this work, we devise a construction scheme. Moreover, we do not assume bounded gradients, as commonly assumed in previous works [90, 93, 94], since this condition does not actually hold in most situations in practice. Note, for instance, that even quadratic risks do not have bounded gradients. For this reason, we will not rely on this condition, and will

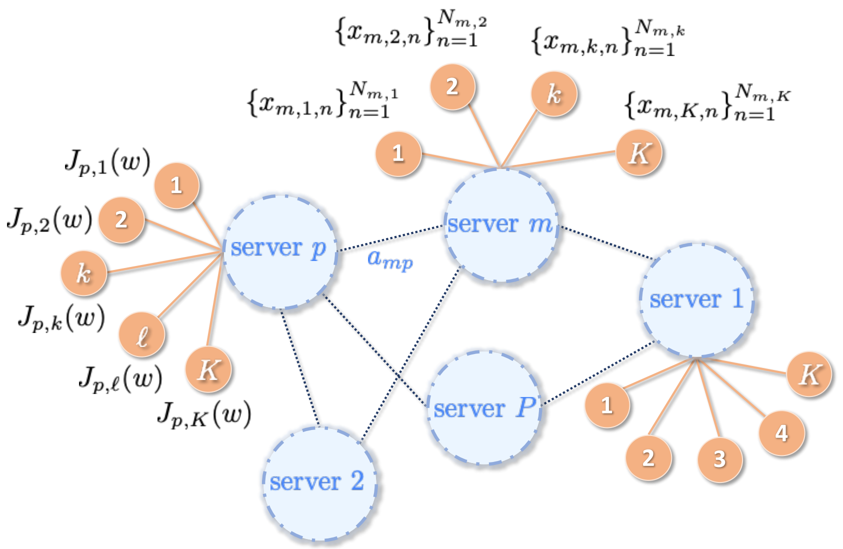


Figure 3.1 – The graph federated learning architecture.

instead be able to show that our noise construction is able to ensure differential privacy with high probability for most cases of interest.

3.2 Graph Federated Architecture

In the graph federated architecture, which we initially introduced in [88], we consider P federated units connected by a graph structure. Each federated unit consists of a server and a set of K agents. Thus, the overall architecture can be represented as a graph depicted in Figure 3.1. We denote the combination matrix connecting the servers by $A \in \mathbb{R}^{P\times P}$, and we write a_{mp} to refer to the elements of A. We assume each agent of every server has its own dataset $\{x_{p,k,n}\}_{n=1}^{N_{p,k}}$ that is non-iid when compared to the other agents. The subscript p refers to the federated unit, k to the agent, and n to the data sample.

With this architecture, we associate a convex optimization problem that will take into account the cost function at each federated unit. Thus, the optimization goal is to find the optimal global model w^o that minimizes an average empirical risk:

$$w^o \triangleq \underset{w\in\mathbb{R}^M}{\text{argmin}}\, \frac{1}{P}\sum_{p=1}^{P}\frac{1}{K}\sum_{k=1}^{K} J_{p,k}(w), \tag{3.1}$$

where each individual cost is an empirical risk defined over the local loss functions $Q_{p,k}(\cdot;\cdot)$:

$$J_{p,k}(w) \triangleq \frac{1}{N_{p,k}}\sum_{n=1}^{N_{p,k}} Q_{p,k}(w; x_{p,k,n}). \tag{3.2}$$

To solve problem (3.1) each federated unit p runs the standard federated averaging (FedAvg) algorithm [11]. An iteration i of the algorithm consists of the server p selecting a subset of L participating agents $\mathcal{L}_{p,i}$. Then, in parallel, each agent runs a series of stochastic gardient descent (SGD) steps. We call these local steps epochs, and denote an epoch by the letter e and the total number of epochs by $E_{p,k}$. The sampled data point at an agent k in the federated unit p during the e^{th} epoch of iteration i is denoted by b. Thus, during an iteration i, each participating agent $k \in \mathcal{L}_{p,i}$ updates the last model $\boldsymbol{w}_{p,i-1}$ and sends its new model $\boldsymbol{w}_{p,k,E_{p,k}}$ to the server after $E_{p,k}$ epochs. During a single epoch e, the agent updates its current local model $w_{p,k,e-1}$ by running a single SGD step. Thus, an agent repeats the following adaptation step for $e = 1, 2, \cdots, E_{p,k}$:

$$\boldsymbol{w}_{p,k,e} = \boldsymbol{w}_{p,k,e-1} - \frac{\mu}{E_{p,k}} \nabla_{w^\mathsf{T}} Q_{p,k}(\boldsymbol{w}_{p,k,e-1}; \boldsymbol{x}_{p,k,b}), \tag{3.3}$$

with $\boldsymbol{x}_{p,k,b}$ be the sampled data of agent k in federated unit p, and $\boldsymbol{w}_{p,k,0} = \boldsymbol{w}_{p,i-1}$. After all the participating agents $k \in \mathcal{L}_{p,i}$ run all their epochs, the server aggregates their final models $\boldsymbol{w}_{p,k,E_{p,k}}$, which we rename as $\boldsymbol{w}_{p,k,i}$ since it is the final local model at iteration i:

$$\boldsymbol{\psi}_{p,i} = \frac{1}{L} \sum_{k \in \mathcal{L}_{p,i}} \boldsymbol{w}_{p,k,i}. \tag{3.4}$$

Next, at the server level, these estimates are combined across neighbourhoods using a diffusion type strategy, where we first consider the previous steps (3.3) and (3.4) as the adaptation step and the following step as the combination step:

$$\boldsymbol{w}_{p,i} = \sum_{m \in \mathcal{N}_p} a_{pm} \boldsymbol{\psi}_{m,i}. \tag{3.5}$$

To introduce privacy, the models communicated at each round between the agents and the servers need to be encrypted in some way. We could either apply secure multiparty computation (SMC) tools, like secret sharing, or use differential privacy. We focus on differential privacy or masking tools that can be represented by added noise. Thus, we let agent 1 in federated unit 2 add a noise component $\boldsymbol{g}_{2,1,i}$ to its final model $\boldsymbol{w}_{2,1,i}$ at iteration i, and then let serever 2 add $\boldsymbol{g}_{12,i}$ to the message $\boldsymbol{\psi}_{2,i}$ it sends to server 1. More generally, we denote by $\boldsymbol{g}_{pm,i}$ the noise added to the message sent by server m to server p at iteration i. Similarly, we denote by $\boldsymbol{g}_{p,k,i}$ the noise added to the model sent by agent k to server p during the i^{th} iteration. We use unseparated subscripts pm for the inter-server noise components to point out their ability to be combined into a matrix structure. Contrarily, the agent-server noise components' subscripts are separated by a comma to highlight a hierarchical structure. Thus, the privatized algorithm can be written as a client update step (3.6), a server aggregation step (3.7), and a server combination step (3.8):

$$\boldsymbol{w}_{p,k,i} = \boldsymbol{w}_{p,i-1} - \frac{\mu}{E_{p,k}} \sum_{e=1}^{E_{p,k}} \nabla_{w^\top} Q_{p,k}(\boldsymbol{w}_{p,k,e-1}; \boldsymbol{x}_{p,k,b}), \tag{3.6}$$

$$\boldsymbol{\psi}_{p,i} = \frac{1}{L} \sum_{k\in\mathcal{L}_{p,i}} \boldsymbol{w}_{p,k,i} + \boldsymbol{g}_{p,k,i}, \tag{3.7}$$

$$\boldsymbol{w}_{p,i} = \sum_{m\in\mathcal{N}_p} a_{pm}(\boldsymbol{\psi}_{m,i} + \boldsymbol{g}_{pm,i}). \tag{3.8}$$

The client update step (3.6) follows from (3.3) by combining the multiple epochs for $e = 1, 2, \cdots, E_{p,k}$ into one update step, with $\boldsymbol{w}_{p,k,i} = \boldsymbol{w}_{p,k,E_{p,k}}$ and $\boldsymbol{w}_{p,k,0} = \boldsymbol{w}_{p,i-1}$, namely:

$$\begin{aligned}\boldsymbol{w}_{p,k,E_{p,k}} &= \boldsymbol{w}_{p,k,E_{p,k}-1} - \frac{\mu}{E_{p,k}} \nabla_{w^\top} Q_{p,k}(\boldsymbol{w}_{p,k,E_{p,k}-1}; \boldsymbol{x}_{p,k,b}) \\ &= \boldsymbol{w}_{p,k,E_{p,k},-2} - \frac{\mu}{E_{p,k}} \sum_{e=E_{p,k}-1}^{E_{p,k}} \nabla_{w^\top} Q_{p,k}(\boldsymbol{w}_{p,k,e-1}; \boldsymbol{x}_{p,k,b}) \\ &= \boldsymbol{w}_{p,k,0} - \frac{\mu}{E_{p,k}} \sum_{e=1}^{E_{p,k}} \nabla_{w^\top} Q_{p,k}(\boldsymbol{w}_{p,k,e-1}; \boldsymbol{x}_{p,k,b}). \end{aligned} \tag{3.9}$$

3.3 Performance Analysis

In this section, we show a list of results on the performance of the algorithm. We study the convergence of the privatized algorithm (3.6)–(3.8), and examine the effect of privatization on performance.

3.3.1 Modeling Conditions

To go forward with our analysis, we require certain reasonable assumptions on the graph structure and cost functions.

Assumption 3.1 (Combination matrix). *The combination matrix A describing the graph is symmetric and doubly-stochastic, i.e.:*

$$a_{pm} = a_{mp}, \quad \sum_{m=1}^{P} a_{mp} = 1. \tag{3.10}$$

Furthermore, the graph is strongly-connected and A satisfies:

$$\iota_2 \overset{\Delta}{=} \rho\left(A - \frac{1}{P}\mathbb{1}\mathbb{1}^{\mathsf{T}}\right) < 1. \tag{3.11}$$

Assumption 3.2 (Convexity and smoothness). *The empirical risks $J_{p,k}(\cdot)$ are $\nu-$strongly convex, and the loss functions $Q_{p,k}(\cdot;\cdot)$ are convex, namely for $\nu > 0$:*

$$J_{p,k}(w_2) \geq J_{p,k}(w_1) + \nabla_{w^{\mathsf{T}}} J_{p,k}(w_1)(w_2 - w_1) + \frac{\nu}{2}\|w_2 - w_1\|^2, \tag{3.12}$$

$$Q_{p,k}(w_2;\cdot) \geq Q_{p,k}(w_1;\cdot) + \nabla_{w^{\mathsf{T}}} Q_{p,k}(w_1;\cdot)(w_2 - w_1). \tag{3.13}$$

Furthermore, the loss functions have $\delta-$Lipschitz continuous gradients, meaning there exists $\delta > 0$ such that for any data point $x_{p,n}$:

$$\|\nabla_{w^{\mathsf{T}}} Q_{p,k}(w_2; x_{p,k,n}) - \nabla_{w^{\mathsf{T}}} Q_{p,k}(w_1; x_{p,k,n})\| \leq \delta\|w_2 - w_1\|. \tag{3.14}$$

We also require a bound on the difference between the global optimal model w^o and the local optimal models $w^o_{p,k}$ that optimize $J_{p,k}(\cdot)$. This assumption is used to bound the gradient noise and the incremental noise defined further ahead. It is not a restrictive assumption, and it imposes a condition on when collaboration is sensical among different agents. In other words, since the agents have non-iid data, sometimes their optimal models are too different and collaboration would hurt their individual performance. For example, when considering recommender systems, people in the same country are more likely to get the same movie recommended as opposed to accross different countries. This means, people of the same country might have different models but relatively close contrary to different countries.

Assumption 3.3 (Model drifts). *The distance of each local model $w^o_{p,k}$ to the global model w^o is uniformly bounded, i.e., there exists $\xi \geq 0$ such that $\|w^o - w^o_p\| \leq \xi$.*

3.3.2 Network Centroid Convergence

We study the convergence of the algorithm from the network centroid's $\boldsymbol{w}_{c,i}$ perspective:

$$\boldsymbol{w}_{c,i} \overset{\Delta}{=} \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{w}_{p,i}. \tag{3.15}$$

We write the central recursion as:

$$\begin{aligned}\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu\frac{1}{PL}\sum_{p=1}^{P}\sum_{k\in\mathcal{L}_{p,i}}\frac{1}{E_{p,k}}\sum_{e=1}^{E_{p,k}}\nabla_{w^\mathsf{T}}Q_{p,k}(\boldsymbol{w}_{p,k,e-1};\boldsymbol{x}_{p,k,b})\\ +\frac{1}{PL}\sum_{p=1}^{P}\sum_{k\in\mathcal{L}_{p,i}}\boldsymbol{g}_{p,k,i}+\frac{1}{P}\sum_{p,m=1}^{P}a_{pm}\boldsymbol{g}_{pm,i}.\end{aligned} \tag{3.16}$$

Next, we define the model error as $\widetilde{\boldsymbol{w}}_{c,i} \overset{\Delta}{=} w^o - \boldsymbol{w}_{c,i}$ and the average gradient noise:

$$\boldsymbol{s}_i \overset{\Delta}{=} \frac{1}{P}\sum_{p=1}^{P}\boldsymbol{s}_{p,i}, \tag{3.17}$$

with the per-unit gradient noise $\boldsymbol{s}_{p,i}$:

$$\boldsymbol{s}_{p,i} \overset{\Delta}{=} \widehat{\nabla_{w^\mathsf{T}}J_p}(\boldsymbol{w}_{p,i-1}) - \nabla_{w^\mathsf{T}}J_p(\boldsymbol{w}_{p,i-1}), \tag{3.18}$$

and:

$$\widehat{\nabla_{w^\mathsf{T}}J_p}(\cdot) \overset{\Delta}{=} \frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}}\frac{1}{E_{p,k}}\sum_{e=1}^{E_{p,k}}\nabla_{w^\mathsf{T}}Q_{p,k}(\cdot;\boldsymbol{x}_{p,k,b}). \tag{3.19}$$

We introduce the average incremental noise $\boldsymbol{q}_i$ and the local incremental noise $\boldsymbol{q}_{p,i}$, which capture the error introduced by the multiple local update steps:

$$\boldsymbol{q}_i \overset{\Delta}{=} \frac{1}{P}\sum_{p=1}^{P}\boldsymbol{q}_{p,i}, \tag{3.20}$$

$$\boldsymbol{q}_{p,i} \overset{\Delta}{=} \frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}}\frac{1}{E_{p,k}}\sum_{e=1}^{E_k}\Big(\nabla_{w^\mathsf{T}}Q_{p,k}(\boldsymbol{w}_{p,k,e-1};\boldsymbol{x}_{p,k,b}) - \nabla_{w^\mathsf{T}}Q(\boldsymbol{w}_{p,i-1};\boldsymbol{x}_{p,k,b})\Big) \tag{3.21}$$

We then arrive at the following error recursion:

$$\widetilde{\boldsymbol{w}}_{c,i} = \widetilde{\boldsymbol{w}}_{c,i-1} + \mu\frac{1}{P}\sum_{p=1}^{P}\nabla_{w^\mathsf{T}}J_p(\boldsymbol{w}_{p,i-1}) + \mu\boldsymbol{s}_i + \mu\boldsymbol{q}_i - \boldsymbol{g}_i, \tag{3.22}$$

where $\boldsymbol{g}_i$ is the total added noise at iteration i:

$$\boldsymbol{g}_i \overset{\Delta}{=} \frac{1}{PL}\sum_{p=1}^{P}\sum_{k\in\mathcal{L}_{p,i}}\boldsymbol{g}_{p,k,i} + \frac{1}{P}\sum_{p,m=1}^{P}a_{pm}\boldsymbol{g}_{pm,i} \tag{3.23}$$

We estimate the first and second-order moments of the gradient noise in the following lemma. To do so, we use the fact, shown in the previous chapter (Lemma 2.1 in Chapter 2), that the individual gradient noise is zero-mean with a bounded second order moment:

$$\mathbb{E}\left\{\|\boldsymbol{s}_{p,i}\|^2|\mathcal{F}_{i-1}\right\} \leq \beta_{s,p}^2\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \sigma_{s,p}^2, \tag{3.24}$$

where the constants are defined as:

$$\beta_{s,p}^2 \triangleq \frac{6\delta^2}{L}\left(1+\frac{1}{K}\sum_{k=1}^{K}\frac{1}{E_{p,k}}\right), \tag{3.25}$$

$$\sigma_{s,p}^2 \triangleq \frac{1}{LK}\sum_{k=1}^{K}\left(\frac{12}{E_{p,k}}+3\right)\frac{1}{N_{p,k}}\sum_{n=1}^{N_{p,k}}\|\nabla_{w^{\mathsf{T}}}Q_{p,k}(w^o;x_{p,k,n})\|^2, \tag{3.26}$$

and $\mathcal{F}_{i-1}$ is the filtration defined over the randomness introduced by all the past subsampling of the data for the calculation of the stochastic gradient. Using Assumption 3.3, we can guarantee that $\sigma_{s,p}^2$ is bounded by bounding:

$$\|\nabla_{w^{\mathsf{T}}}Q_{p,k}(w^o;x_{p,k,n})\|^2 \leq 2\|\nabla_{w^{\mathsf{T}}}Q_{p,k}(w_{p,k}^o;x_{p,k,n})\|^2 + 2\delta^2\xi^2. \tag{3.27}$$

Lemma 3.1 (**Estimation of moments of the gradient noise**). *The gradient noise defined in* (3.17) *is zero-mean and has a bounded second-order moment:*

$$\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2|\mathcal{F}_{i-1}\right\} \leq \beta_s^2\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \sigma_s^2 + \frac{2}{P}\sum_{p=1}^{P}\beta_{s,p}^2\|\boldsymbol{w}_{p,i-1}-\boldsymbol{w}_{c,i-1}\|^2 \tag{3.28}$$

where the constants β_s^2 and σ_s^2 are given by:

$$\beta_s^2 \triangleq \frac{2}{P}\sum_{p=1}^{P}\beta_{s,p}^2, \quad \sigma_s^2 \triangleq \frac{1}{P}\sum_{p=1}^{P}\sigma_{s,p}^2. \tag{3.29}$$

Proof. The result follows from applying the Jensen's inequality and the bounds on the per-unit gradient noise $\boldsymbol{s}_{p,i}$:

$$\begin{aligned}\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2|\mathcal{F}_{i-1}\right\} &\leq \frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\left\{\|\boldsymbol{s}_{p,i}\|^2|\mathcal{F}_{i-1}\right\}\\ &\leq \frac{1}{P}\sum_{p=1}^{P}\beta_{s,p}^2\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \sigma_{s,p}^2.\end{aligned} \tag{3.30}$$

□

The new term found in the bound of the gradient term is what we call the network disagreement:

$$\frac{1}{P}\sum_{p=1}^{P}\|\boldsymbol{w}_{p,i}-\boldsymbol{w}_{c,i}\|^2. \tag{3.31}$$

It captures the difference in the path taken by the individual models versus the network centroid. We bound this difference in Lemma 3.3. However, before doing so, we show that the second order moment of the incremental noise is on the order of $O(\mu)$. From Lemma 2.5 in Chapter 2, we can bound the individual incremental noise:

$$\mathbb{E}\|\boldsymbol{q}_{p,i}\|^2 \leq a\mu^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + a\mu^2\xi^2 + \frac{1}{K}\sum_{k=1}^{K}(b_k\mu^4 + c_k\mu^2)\sigma_{q,p,k}^2, \tag{3.32}$$

where the constants are given by:

$$a \triangleq \frac{4\delta^2}{K}\sum_{k=1}^{K}\frac{(E_{p,k}+1)(1-\lambda)-1+\lambda^{E_{p,k}+1}}{E_{p,k}^2(1-\lambda)^2}, \tag{3.33}$$

$$b_k \triangleq \frac{2E_{p,k}(E_{p,k}+1)(1-\lambda)^2-4E_{p,k}(1-\lambda)+4\lambda}{E_{p,k}^2(1-\lambda)^3} - \frac{2\lambda^{E_{p,k}+1}}{E_{p,k}^2(1-\lambda)^3}, \tag{3.34}$$

$$c_k \triangleq \frac{E_{p,k}-1}{3E_{p,k}}, \tag{3.35}$$

$$\lambda \triangleq 1-2\nu\mu+4\delta^2\mu^2, \tag{3.36}$$

$$\sigma_{q,p,k}^2 \triangleq 3\sum_{n=1}^{N_{p,k}}\|\nabla_{w^\mathsf{T}}Q_{p,k}(w_{p,k}^o;x_{p,k,n})\|^2. \tag{3.37}$$

The following result follows.

Lemma 3.2 (Estimation of second-order moment of the incremental noise). *The incremental noise defined in* (3.20) *has a bounded second-order moment:*

$$\mathbb{E}\|\boldsymbol{q}_i\|^2 \leq O(\mu)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + O(\mu)\xi^2 + O(\mu^2)\sigma_q^2 + \frac{O(\mu)}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1}-\boldsymbol{w}_{c,i-1}\|^2, \tag{3.38}$$

where the constant σ_q^2 is the average of $\sigma_{q,p,k}^2$:

$$\sigma_q^2 \triangleq \frac{1}{PK}\sum_{p=1}^{P}\sum_{k=1}^{K}(b_k\mu^4+c_k\mu^2)\sigma_{q,p,k}^2. \tag{3.39}$$

Proof. The result follows from applying the Jensen inequality and the bounds on the per-unit incremental noise $\boldsymbol{q}_{p,i}$:

$$\begin{aligned}\mathbb{E}\|\boldsymbol{q}_i\|^2 &\leq \frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{q}_{p,i}\|^2 \\ &\leq a\mu^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + a\mu^2\xi^2 + \frac{1}{K}(b_k\mu^4 + c_k\mu^2)\sigma^2_{q,p,k}. \qquad (3.40)\end{aligned}$$

Furthermore, $a = O(\mu^{-1}), b_k = O(\mu^{-1})$, and $c_k = O(1)$ reduce the expression to (3.38).

□

We now bound the network disagreement. To do so, we first introduce the eigendecomposition of $A = QHQ^{\mathsf{T}}$:

$$Q \overset{\Delta}{=} \begin{bmatrix}\frac{1}{\sqrt{P}}\mathbb{1} & Q_\theta\end{bmatrix}, \quad H \overset{\Delta}{=} \begin{bmatrix}1 & 0\\ 0 & H_\theta\end{bmatrix}, \qquad (3.41)$$

where H_θ is a diagonal matrix that includes the last $(P-1)$ eigenvalues of A and Q_θ their corresponding eigenvectors.

Lemma 3.3 (Network disagreement). *The average deviation from the centroid is bounded during each iteration i:*

$$\begin{aligned}\frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i} - \boldsymbol{w}_{c,i}\|^2 \leq \frac{\iota_2^i}{P}\mathbb{E}\|(Q_\epsilon\otimes I)\boldsymbol{\mathcal{W}}_0\|^2 + \frac{\iota_2^2}{P}\sum_{j'=0}^{i-1}\iota_2^{j'}\sum_{p=1}^{P}\Bigg\{\mu^2\Bigg(\frac{2\delta^2}{\iota_2(1-\iota_2)} + \beta^2_{s,p} \\ + O(\mu)\Bigg)\Bigg(\lambda_p^{j'}A^{j'}[p]\,col\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\}_{p=1}^{P} + \sum_{j=0}^{j'-1}\lambda_p^{j}A^{j}[p] \\ \times\, col\Big\{\mu^2\sigma^2_{s,p} + O(\mu^2)\xi^2 + O(\mu^3)\sigma^2_{q,p} + \sigma^2_{g,p}\Big\}_{p=1}^{P}\Bigg) \\ + \mu^2\frac{2\|\nabla_{w^{\mathsf{T}}}J_p(w^o)\|^2}{\iota_2(1-\iota_2)} + \mu^2\sigma^2_{s,p} + O(\mu^3)\xi^2 + O(\mu^4)\sigma^2_{q,p} + \frac{1}{\iota_2^2}\sigma^2_{g,p}\Bigg\}, \qquad (3.42)\end{aligned}$$

where $\boldsymbol{\mathcal{W}}_0 \overset{\Delta}{=} col\Big\{\boldsymbol{w}_{p,0}\Big\}_{p=1}^{P}$ *and*

$$\lambda_p \overset{\Delta}{=} \sqrt{1 - 2\nu\mu + \delta^2\mu^2} + \beta^2_{s,p}\mu^2 + O(\mu^2) \in (0,1). \qquad (3.43)$$

Then, in the limit:

$$\limsup_{i\to\infty} \frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i}-\boldsymbol{w}_{c,i}\|^2 \leq \frac{\iota_2^2}{P(1-\iota_2)}\sum_{p=1}^{P}\mu^2\sigma_{s,p}^2+\frac{1}{\iota_2^2}\sigma_{g,p}^2+O(\mu)\sigma_{g,p}^2+O(\mu^3). \tag{3.44}$$

Proof. See Appendix 3.B. □

Thus, from the above lemma, we see that the individual models gravitate to the centroid model with an error introduced due to the added privatization. The effect of the added noise overpowers that of the gradient and incremental noise, since the later is on the order of the step-size.

Then, using the above result, we can establish the convergence of the centroid model to a neighbourhood of the true optimal model w^o in the MSE sense.

Theorem 3.1 (Centroid MSE convergence). *Under Assumptions 3.1, 3.2 and 3.3, the network centroid converges to the optimal point w^o exponentially fast for a sufficiently small step-size μ:*

$$\begin{aligned}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq &\lambda_c\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2+\mu^2\sigma_s^2+O(\mu^2)\xi^2+O(\mu^3)\sigma_q^2+\mathbb{E}\|\boldsymbol{g}_i\|^2\\ &+\frac{O(\mu)}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1}-\boldsymbol{w}_{c,i-1}\|^2,\end{aligned} \tag{3.45}$$

where:

$$\lambda_c=\sqrt{1-2\nu\mu+\delta^2\mu^2}+\beta_s^2\mu^2+O(\mu^2)\in(0,1). \tag{3.46}$$

Then, letting i tend to infinity, we get:

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq \frac{\mu^2\sigma_s^2+O(\mu^2)\xi^2+O(\mu^3)\sigma_q^2+\mathbb{E}\|\boldsymbol{g}\|^2}{1-\lambda_c}+\sum_{p=1}^{P}O(1)\sigma_{g,p}^2+O(\mu). \tag{3.47}$$

Proof. See Appendix 3.C. □

The main term in the above bound is the variance of the added noise with a dominating factor of μ^{-1}, since:

$$1 - \lambda_c = 1 - \sqrt{1 - O(\mu) + O(\mu^2)} - O(\mu^2) = O(\mu) - O(\mu^2) = O(\mu) \tag{3.48}$$

which allows us to rewrite the bound as follows:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq O(\mu)\sigma_s^2 + O(\mu)\xi^2 + O(\mu^2)\sigma_q^2 + O(\mu^{-1})\mathbb{E}\|\boldsymbol{g}\|^2 + \sum_{p=1}^{P} O(1)\sigma_{g,p}^2 + O(\mu), \tag{3.49}$$

with $\mathbb{E}\|\boldsymbol{g}\|^2$ representing the variance of the total added noise, independent of time. While in general decreasing the step-size improves performance, the above result shows that this need not be the case with privatization. Thus, since the added noise impacts the model utility negatively, it is important to choose a privatization scheme that reduces the effect. In what follows, we look closely at such a scheme.

3.3.3 Graph Homomorphic Perturbations

We consider a specific privatization scheme and specialize the above results. The goal of the scheme is to remove the $O(\mu^{-1})$ term from the MSE bounds. Thus, we wish to cancel out the total added noise amongst servers, i.e.,

$$\sum_{p,m=1}^{P} a_{pm}\boldsymbol{g}_{pm,i} = 0. \tag{3.50}$$

To achieve this, we introduce graph homomorphic perturbations defined as follows [86]. We assume each server p draws a sample $\boldsymbol{g}_{p,i}$ independently from the Laplace distribution $Lap(0, \sigma_g/\sqrt{2})$ with variance σ_g^2. Server p then sets the noise $\boldsymbol{g}_{mp,i}$ added to the message sent to its neighbour m as:

$$\boldsymbol{g}_{mp,i} = \begin{cases} \boldsymbol{g}_{p,i}, & m \neq p \\ -\frac{1-a_{pp}}{a_{pp}}\boldsymbol{g}_{p,i}, & m = p \end{cases} \tag{3.51}$$

Thus, with such a scheme, the noise components proportional to $O(\mu^{-1})$ resulting from the noise added between the servers cancel out in the error recursions, and the remaining error introduced by the noise is controlled by the step-size. Thus, its effect can be mitigated by using a smaller step-size. In the next corollary, we show that if no noise is added amongst the clients and graph-homorphic perturbations are used amongst servers, then the error converges to $O(1)\sigma_g^2$.

Corollary 3.1 (Centroid MSE convergence under graph homomorphic perturbations). *Under Assumptions 3.1, 3.2 and 3.3, the network centroid with graph homomorphic perturbations converges to the optimal point w^o exponentially fast for a sufficiently small step-size μ:*

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq \lambda_c \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\sigma_s^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_q^2 + \frac{O(\mu)}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2. \tag{3.52}$$

Then, letting i tend to infinity, we get:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq \frac{\mu^2\sigma_s^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_q^2}{1-\lambda_c} + \sum_{p=1}^{P} O(1)\sigma_{g,p}^2 + O(\mu). \tag{3.53}$$

Proof. Starting from (3.47), and replacing $\mathbb{E}\|\boldsymbol{g}\|^2 = 0$ because $\boldsymbol{g}_i = 0$, we get the final result. □

3.4 Privacy Analysis

We study the privacy of the algorithm (3.6)–(3.8) in terms of differential privacy. We focus on graph homomorphic perturbations and show that the adopted scheme is differentially private. To do so, we first define what it means for an algorithm to be $\epsilon-$differentially private. Therefore, without loss of generality, assume agent 1 in federated unit 1 decides to not participate, and its data samples $x_{1,1}$ are replaced by a new set $x'_{1,1}$ with a different distribution. Then, with the new data, the algorithm takes a different path. We denote the new models by $\boldsymbol{w}'_{p,k,i}$. The idea behind differential privacy is that an outside observant should not be able to distinguish between the two trajectories $\boldsymbol{w}_{p,k,i}$ and $\boldsymbol{w}'_{p,k,i}$ and conclude whether agent one participated in the training. More formally, differential privacy is defined bellow.

Definition 3.1 ($\epsilon(i)-$Differential privacy). *We say that the algorithm given in (3.6)–(3.8) is $\epsilon(i)-$differentially private for server p at time i if the following condition holds on the joint distribution $f(\cdot)$:*

$$\frac{f\left(\left\{\{\boldsymbol{\psi}_{p,j} + \boldsymbol{g}_{pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=0}^{i}\right)}{f\left(\left\{\left\{\boldsymbol{\psi}'_{p,j} + \boldsymbol{g}_{pm,j}\right\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=0}^{i}\right)} \leq e^{\epsilon(i)}. \tag{3.54}$$

Thus, the above definition states that minimaly varried trajectories have comparable probabilities. In addition, the smaller the value of ϵ is, the higher the privacy guarantee will be. Thus, the goal will be to decrease ϵ as long as the model utility is not strongly affected.

Next, in order to show that the algorithm is differentially private, we require the sensitivity of the alogorithm to be bounded. The sensitivity at time i is defined as:

$$\Delta(i) = \|\boldsymbol{\mathcal{w}}_i - \boldsymbol{\mathcal{w}}_i'\|. \tag{3.55}$$

It measures the distance between the original and perturbed weight vectors. It is shown in Appendix 3.D that $\Delta(i)$ can be bounded as follows:

$$\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\|, \tag{3.56}$$

for constants B and B' chosen by the designer. Moreover, the above bound holds with high probability given by:

$$\begin{aligned}\mathbb{P}\left(\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\|\right) \geq &\left(1 - \frac{\lambda_{\max}^i \mathbb{E}\|\boldsymbol{\mathcal{w}}_0\|^2 + O(\mu) + O(\mu^{-1})}{B^2}\right) \\ &\times \left(1 - \frac{\lambda_{\max}'^i \mathbb{E}\|\boldsymbol{\mathcal{w}}_0'\|^2 + O(\mu) + O(\mu^{-1})}{B'^2}\right).\end{aligned} \tag{3.57}$$

This result shows that the sensitivity can be bounded with high probability, which in turn is dependent on the values chosen for B and B'. Larger values for these constants increase the probability, but nevertheless lead to a looser bound for privacy (as shown in Theorem 3.2). Therefore, the choice of B and B' needs to be balanced judiciously to ensure the desired level of privacy.

Using the bound on the sensitivity and from the definition of differential privacy, we can finally show that the algorithm is differentially private with high probability.

Theorem 3.2 (Privacy of GFL algorithm). *If the algorithm (3.6)–(3.8) adopts graph homomorphic perturbations, then it is $\epsilon(i)-$differentially private with high probability, at time i for a standard deviation of:*

$$\sigma_g = \frac{\sqrt{2}}{\epsilon(i)}(B + B' + \sqrt{P}\|w^o - w'^o\|)(i+1). \tag{3.58}$$

Proof. See Appendix 3.E. □

Thus, the above theorem suggests, if we wish the algorithm to be $\epsilon(i)-$differentially private, then we need to choose the noise variance accordingly. The larger the variance is, the more private the algorithm will be. However, the longer the algorithm is run, we will require a larger noise variance to keep the same level of privacy guarantee. Said differently, if we fix the added noise, then as time passes, the algorithm becomes less private, and more information is leaked. However, with graph-homomorphic perturbations, we can afford to increase the varianve since its effect is constant on the MSE, and thus decreases the leakage.

3.5 Experimental Results

We conduct a series of experiments to study the influence of privatization on the GFL algorithm. The aim of the experiments is to show the superior performance of graph homomorphic perturbations to random perturbations and perturbations to gradients versus models, and to study the effect of different parameters on the performance of the algorithm.

3.5.1 Regression

We first start by studying a regression problem on simulated data. We do so for the tractability of the problem. We consider the quadratic loss that has a closed form solution, i.e., a formal expression for the true model w^o is known, which makes the calculation of the mean square error feasible and more accurate.

Therefore, consider a streaming feature vector $\boldsymbol{u}_{p,k,n} \in \mathbb{R}^M$ with output variable $\boldsymbol{d}_{p,k}(n) \in \mathbb{R}$ given by:

$$\boldsymbol{d}_{p,k}(n) = \boldsymbol{u}_{p,k,n}^{\mathsf{T}} w^\star + \boldsymbol{v}_{p,k}(n), \tag{3.59}$$

where $w^\star \in \mathbb{R}^M$ is some generating model, and $\boldsymbol{v}_{p,k}(n)$ is some zero-mean Guassian random variable with $\sigma^2_{v_{p,k}}$ variance and independent of $\boldsymbol{u}_{p,k,n}$. Then, the optimal model that solves the following problem:

$$\min_w \frac{1}{P}\sum_{p=1}^{P}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_{p,k}}\sum_{n=1}^{N_{p,k}}\|\boldsymbol{d}_{p,k}(n) - \boldsymbol{u}_{p,k,n}^{\mathsf{T}} w\|^2 + \rho\|w\|^2 \tag{3.60}$$

is found to be:

$$w^o = (\widehat{R}_u + \rho I)^{-1}(\widehat{R}_u w^\star + \widehat{r}_{uv}), \tag{3.61}$$

where $\widehat{R}_u$ and $\widehat{r}_{uv}$ are defined as:

$$\widehat{R}_u \overset{\Delta}{=} \frac{1}{P}\sum_{p=1}^{P}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_{p,k}}\sum_{n=1}^{N_k}\boldsymbol{u}_{p,k,n}\boldsymbol{u}_{p,k,n}^{\mathsf{T}}, \tag{3.62}$$

$$\widehat{r}_{uv} \overset{\Delta}{=} \frac{1}{P}\sum_{p=1}^{P}\frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_{p,k}}\sum_{n=1}^{N_k}\boldsymbol{v}_{p,k}(n)\boldsymbol{u}_{p,k,n}. \tag{3.63}$$

We consider $P = 10$ units, each with $K = 100$ total agents. We assume, $N_{p,k} = 100$ for each agent. We randomly generate two-dimensional feature vectors $\boldsymbol{u}_{p,k}(n)$ from a Guassian random vector with zero-mean and a randomly generated covarinace matrix $R_{u_{p,k}}$. We then calculate the corresponding outputs according to (3.59). To make the data non-iid accross agents, we assume the covariance matrix $R_{u_{p,k}}$ is different for each agent, as well as the variance $\sigma^2_{v_{p,k}}$ of the added noise. When running the algorithm, we assume each unit samples at random $L = 11$ agents, and each agent runs $E_{p,k} \in [1, 10]$ epochs and uses a mini-batch of $B_{p,k} \in [5, 10]$ samples.

We compare three algorithms: the standard GFL algorithm, the privatized GFL algorithm with random perturbations, and the privatized GFL with homomorphic perturbations. We do not add noise between the clients and their server to focus on the effect of the perturbations between the servers. In the first set of simulations, we fix the step-size $\mu = 0.7$ and the regularization parameter $\rho = 0.1$. We fix the variance of the added noise for privatization in both schemes to $\sigma_g^2 = 0.1$. We then plot the mean-square deviation (MSD) at each time step for the centroid model:

$$\text{MSD}_i \overset{\Delta}{=} \|\boldsymbol{w}_{c,i} - w^o\|^2, \tag{3.64}$$

as seen in Figure 3.2. We observe that the privatized GFL with random perturbations has lower performance compared to the other two algorithms. While, using homomorphic perturbations does not result in such a decay in performance. Thus, our suggested scheme does a good job at tracking the performance of the original GFL algorithm, while not compromising with the privacy level.

We next study the extent of the effect of the noise on the model utility. Thus, we run a series of experiments with varying added noise $\sigma_g^2 = \{0.001, 0.01, 0.1, 1, 2, 10\}$ for the two privatized GFL algorithms. We plot the resulting MSD curves in Figure 3.3a. We obsereve for a fixed step-size, as we increase the variance, the MSD of the algorithm with random perturbations increases significantly as opposed to the algorithm with homomorphic perturbations. Thus, we conclude that the algorithm with random perturbtaions is more sensitive to the variance of the added noise. In fact, at some point, while using random perturbations, for some variance, the algorithm breaks down. While using graph homomorphic perturbations, delays that effect for much larger variance. In addition, as long as the step-size is small enough, we can always control the effect of the

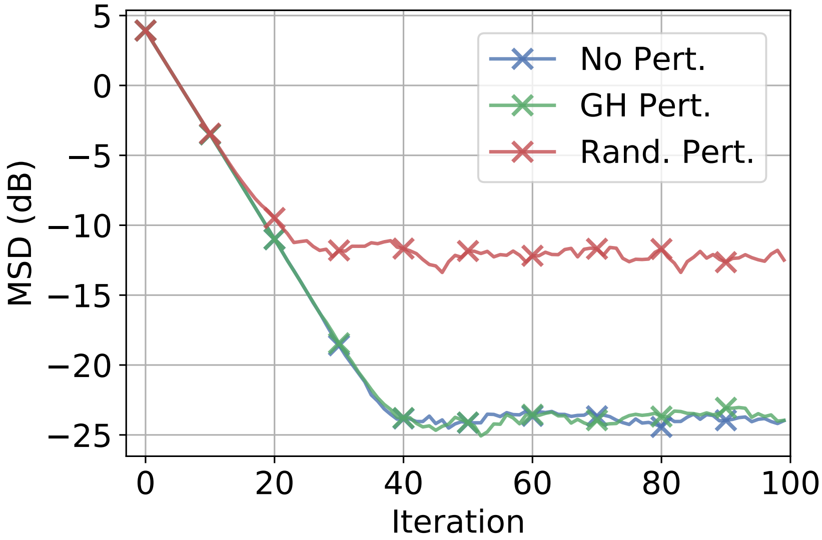


Figure 3.2 – Performance of GFL with no perturbations (blue), with graph homomorphic perturbations (green), and random perturbations (red).

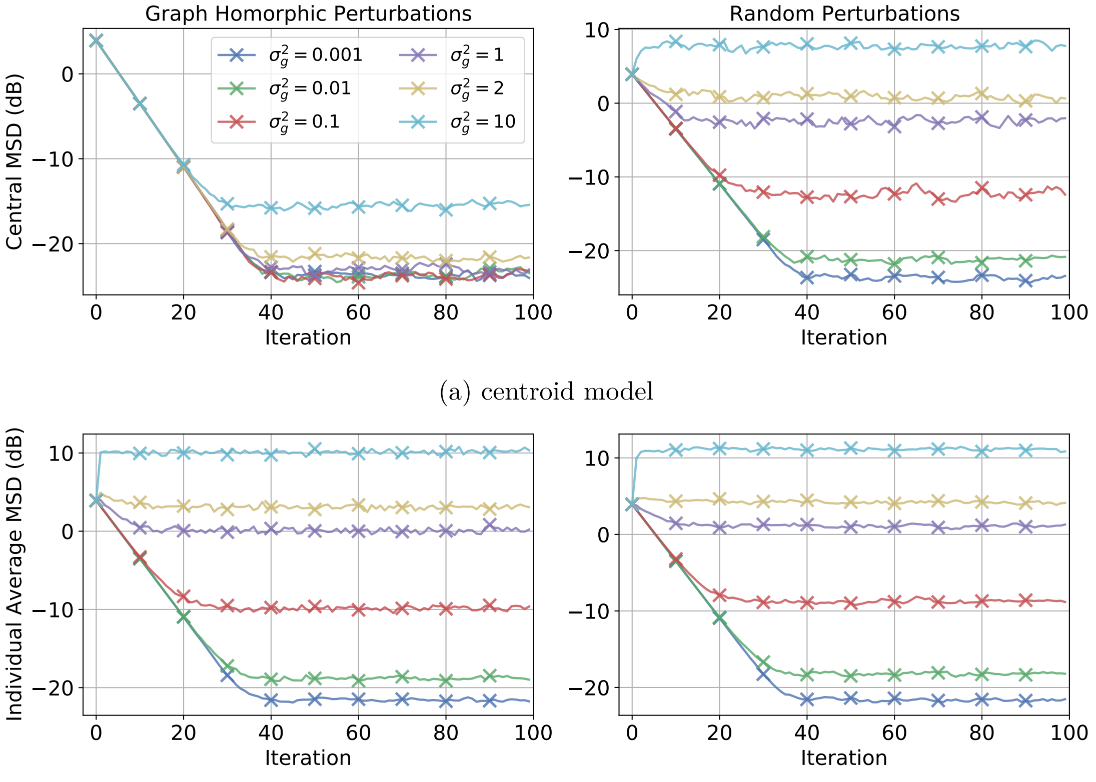
graph homomorphic perturbations.

However, if we were to look at the individual MSD for one federated unit, we would discover that the performance of the algorithm decays as the noise variance is increased. Nonetheless, it is not to the extent of random perturbations. We plot in Figure 3.3b the average individual MSD for the varying noise variance:

$$\text{MSD}_{\text{avg},i} \triangleq \frac{1}{P}\sum_{p=1}^{P}\|\boldsymbol{w}_{p,i} - w^o\|^2. \tag{3.65}$$

We observe that for a fixed noise variance, homomorphic perturbations results in a better performance. Furthermore, as we increase the noise variance, the network disagreement increases for both schemes. This comes as no surprise and is in accordance with Lemma 3.3. Furthermore, as previously mentioned, graph homomorphic perturbations have the added value of not being negatively affected by the decrease in the step-size. In addition, even though the improvement does not seem significant, the source of the error of the two schemes is different. Furthemore, the information of the true model is distributed in the network and can be retrieved by running at the end of the learning algorithm a consensus-type step. At that point, the local models no longer contain information about the local data, and thus agents can safely share their models. However, when random perturbations are used, reconstruction is not possible since the information has been lost in the netwrok due to the added perturbations.

We next fix the noise variance $\sigma_g^2 = 0.1$ and varying the step-size $\mu = \{0.1, 0.5, 1, 5\}$. According to Theorem 3.1, the MSD resulting from random perturbations includes an



(a) centroid model

(b) individual models

Figure 3.3 – Performance curves of privatized GFL with varying noise variance.

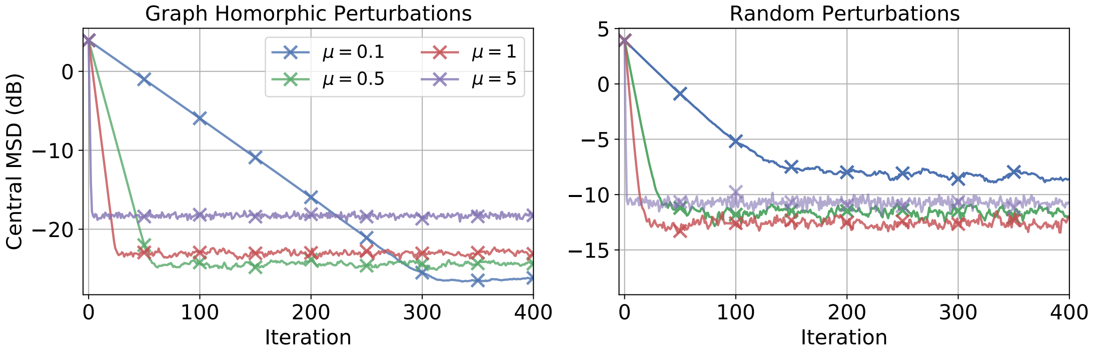


Figure 3.4 – Performance curves of privatized GFL with varying step-size.

$O(\mu^{-1})$ term, which is not the case when using graph homomorphic perturbations. Thus, we expect a decrease in the step-size will not significantly affect the privatized algorithm with graph homomorphic perturbations as opposed to random perturbations. Indeed, as seen in Figure 3.4, as μ is increased, the final MSD increases; this is probably due to the $O(\mu)\sigma_s^2$ term in the bound. In contrast, for significantly small or large μ, the performance of the privatized algorithm with random perturbations decreases. In addition, what we observe for both privacy schemes, is that the rate of convergence slows down as we decrease the step-size. Thus, there exists an optimal step-size that achieves a good compromise between a fast convergence and a low MSD.

3.5.2 Classification

We now focus on a classification problem applied to a dataset on click rate prediction of ads. We consider the Avazu click through dataset [79]. We split the 5101 data unequally among a total of 50 agents. We assume there are $P = 5$ units each with $K = 10$ agents. We add non-idd noise to the data at each agent to change their distributions. We again compare three algorithms: standard GFL, privatized GFL with homomorphic perturbations, and privatized GFL with random perturbations. We use a regularized logistic risk with regularization parameter $\rho = 0.03$. We set the step-size $\mu = 0.5$. We repeat the algorithms for multiple levels of privacy. We then settle on a noise variance $\sigma_g^2 = 0.6$ for which the privatized algorithm with random perturbations still converges. We plot in Figure 3.5 the testing error on a set of 256 clean samples that were not perturbed with noise to change their distributions. We use the centriod model learned during each iteration to calculate the corresponding testing error. We observe that the graph homorphic perturbations do not hinder the performance of the privatized model. As for random perturbations, they significantly reduce the utility of the learnt model.

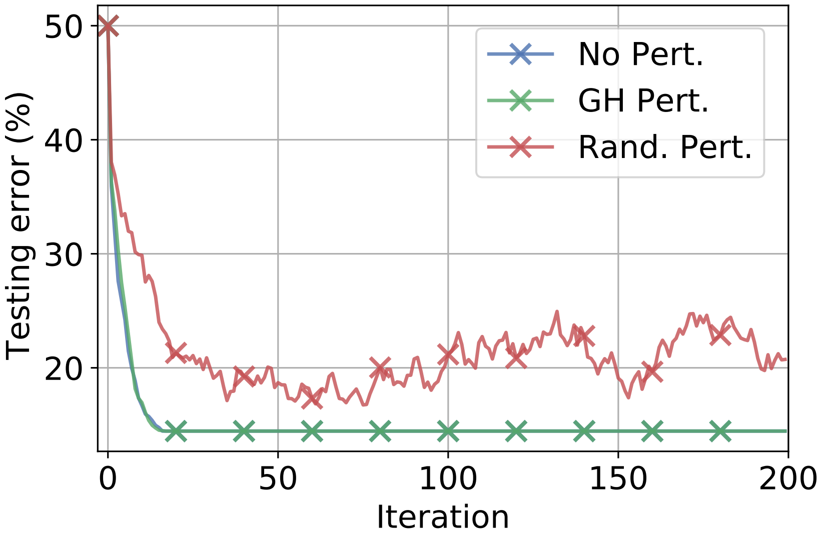


Figure 3.5 – Testing error of GFL with no perturbations (blue), with graph homomorphic perturbations (green), and random perturbations (red).

3.6 Conclusion

In this chapter, we introduced graph federated learning and implemented an algorithm that guarantees privacy of the data in a differential privacy sense. We showed general privatization based on adding random perturbations to updates in federated learning have a negative effect on the performance of the algorithm. Random perturbations drive the algorithm farther away from the true optimal model. However, we showed by adding graph homomorphic perturbations, which exploit the graph structure, performance can be recovered with guaranteed privacy. We also showed that using dependent perturbations does not result in the same trade-off between privacy and efficiency. Thus, we no longer have to choose what to prioritize, and instead, we can have both a highly privatized algorithm with a good model utility.

3.A Auxiliary Result on Individual MSE Performance

We first introduce the following theorem, which will be used to bound the network disagreement. We loosely bound the individual MSE for each federated unit. A tighter bound can be found, however, it is not needed.

Theorem 3.3 (Individual MSE convergence). *Under Assumptions 3.1, 3.2 and 3.3, the individual models converge to the optimal model w^o exponentially fast for a sufficiently small step-size:*

$$\begin{aligned}&\text{col}\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2\}_{p=1}^P \\ &\preceq \Lambda^i \text{col}\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\}_{p=1}^P + \sum_{j=0}^{i} \Lambda^j \text{col}\{\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2\}_{p=1}^P,\end{aligned} \tag{3.66}$$

where $\preceq$ is the elementwise comparison, Λ is a diagonal matrix with the p^{th} entry given by:

$$\lambda_p = \sqrt{1 - 2\nu\mu + \delta^2\mu^2} + \beta_{s,p}^2\mu^2 + O(\mu^2) \in (0,1), \tag{3.67}$$

$\sigma_{q,p}^2$ the average of $\sigma_{q,p,k}^2$, and $\sigma_{g,p}^2$ is the total variance introduced by the noise added at server p. Then, taking the limit of i to infinity:

$$\limsup_{i\to\infty} col\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2\Big\}_{p=1}^P \preceq (I-\Lambda)^{-1} col\Big\{\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2\Big\}_{p=1}^P. \tag{3.68}$$

Proof. Focusing on the error of a single server p, we can verify that:

$$
\begin{aligned}
&\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{p,i}\|^2|\mathcal{F}_{i-1}\}\\
&\overset{(a)}{=} \mathbb{E}\left\{\left\| \sum_{m\in\mathcal{N}_p} a_{pm}\Big(\widetilde{\boldsymbol{w}}_{m,i-1} + \mu\nabla_{w^\mathsf{T}} J_m(\boldsymbol{w}_{m,i-1}) + \mu\boldsymbol{q}_{m,i}\Big)\right\|^2 \middle|\mathcal{F}_{i-1}\right\}\\
&\quad + \mu^2\mathbb{E}\left\{\left\|\sum_{m\in\mathcal{N}_p} a_{pm}\boldsymbol{s}_{m,i}\right\|^2\middle|\mathcal{F}_{i-1}\right\} + \mathbb{E}\left\{\left\|\sum_{m\in\mathcal{N}_p}\frac{a_{pm}}{L}\sum_{k\in\mathcal{L}_{m,i}}\boldsymbol{g}_{m,k,i}\right\|^2\middle|\mathcal{F}_{i-1}\right\}\\
&\quad + \mathbb{E}\left\{\left\|\sum_{m\in\mathcal{N}_p} a_{pm}\boldsymbol{g}_{pm,i}\right\|^2\middle|\mathcal{F}_{i-1}\right\},\\
&\overset{(b)}{\leq} \sum_{m\in\mathcal{N}_p} a_{pm}\Bigg(\frac{1}{\alpha}\|\widetilde{\boldsymbol{w}}_{m,i-1} + \mu\nabla_{w^\mathsf{T}}J_m(\boldsymbol{w}_{m,i-1})\|^2 + \frac{\mu^2}{1-\alpha}\Big(O(\mu)\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 + O(\mu)\xi^2\\
&\quad + O(\mu^2)\sigma_{q,m}^2\Big) + \mu^2\left(\sigma_{s,m}^2 + \beta_{s,m}^2\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2\right) + \frac{1}{LK}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{g}_{m,k,i}\|^2 + \mathbb{E}\|\boldsymbol{g}_{pm,i}\|^2\Bigg),\\
&\overset{(c)}{\leq} \sum_{m\in\mathcal{N}_p} a_{pm}\Bigg(\left(\frac{1-2\nu\mu+\delta^2\mu^2}{\alpha} + \beta_{s,m}^2\mu^2 + \frac{O(\mu^3)}{1-\alpha}\right)\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 + \mu^2\sigma_{s,m}^2\\
&\quad + \frac{O(\mu^3)\xi^2 + O(\mu^4)\sigma_{q,m}^2}{1-\alpha} + \frac{1}{LK}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{g}_{m,k,i}\|^2 + \mathbb{E}\|\boldsymbol{g}_{pm,i}\|^2\Bigg),
\end{aligned}
\tag{3.69}
$$

where we define $\sigma_{q,m}^2$ to be the average of $\sigma_{q,m,k}^2$. Step (a) follows from independence of random variables and the zero-mean of the gradient noise and the added noise, (b) from Jensen's inequality and the bound on the gardient noise (3.24) and the incremental noise (3.38), (c) from ν-strong convexity and δ-Lipschtz continuity. Then, choosing:

$$
\alpha = \sqrt{1-2\nu\mu+\delta^2\mu^2} = 1 - O(\mu), \tag{3.70}
$$

and taking the expectation over the filtration, we get:

$$
\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2 \leq \sum_{m\in\mathcal{N}_p} a_{pm}\left(\lambda_m\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 + \mu^2\sigma_{s,m}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,m}^2 + \sigma_{g,m}^2\right), \tag{3.71}
$$

where we introduce the constants λ_m and $\sigma_{g,m}^2$, which is the total variance introduced by the noise added at server m :

$$
\lambda_m \triangleq \sqrt{1-2\nu\mu+\delta^2\mu^2} + \beta_{s,m}^2\mu^2 + O(\mu^2). \tag{3.72}
$$

Next, taking the column vector of every local MSE, we get the following bound in which we drop the indexing from the column vectors:

$$
\begin{aligned}
&\mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2\Big\} \\
&\preceq \Lambda A\,\mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2\Big\} + A\,\mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + \sigma_{g,p}^2 + O(\mu^2)\xi^2\Big\} + A\,\mathrm{col}\Big\{O(\mu^3)\sigma_{q,p}^2\Big\}, \\
&\preceq \Lambda^i A^i \mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\} + \sum_{j=0}^{i}\Lambda^j A^j \mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + \sigma_{g,p}^2\Big\} + \Lambda^j A^j \mathrm{col}\Big\{O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2\Big\}, \\
&\preceq \Lambda^i \mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\} + \sum_{j=0}^{i}\Lambda^j \mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + \sigma_{g,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2\Big\}, \qquad (3.73)
\end{aligned}
$$

where we define the diagonal matrix Λ with λ_p as entries on the diagonal. Then choosing μ small enough such that $\lambda_p < 1$ for every p, we know the limit of Λ^i as i goes to infinity is zero. Furthermore, if the eigenvalues of Λ are less than 1, which they are, then the geometric series converges to $(I-\Lambda)^{-1}$. Thus, we get the desired result.

□

3.B Proof of Lemma 3.3

Consider the aggregate model vector, i.e., $\mathcal{W}_i \overset{\Delta}{=} \mathrm{col}\Big\{\boldsymbol{w}_{p,i}\Big\}_{p=1}^{P}$, for which we write the model recursion as:

$$
\begin{aligned}
\mathcal{W}_i =&(A\otimes I)^{\mathsf{T}}\left(\mathcal{W}_{i-1} - \mu\mathrm{col}\Big\{\nabla_{w^{\mathsf{T}}}J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{s}_{p,i} + \boldsymbol{q}_{p,i}\Big\} + \mathrm{col}\left\{\frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}}\boldsymbol{g}_{p,k,i}\right\}\right) \\
&+ \mathrm{diag}\Big((A\otimes I)^{\mathsf{T}}\boldsymbol{\mathcal{G}}_i\Big), \qquad (3.74)
\end{aligned}
$$

where $\boldsymbol{\mathcal{G}}_i$ is a matrix whose entries are the noise $\boldsymbol{g}_{pm,i}$, and the diag$(\cdot)$ function extracts the diagonal entries of a matrix and transforms them into a column vector.

Since A is doubly-stochastic, then it admits an eigendecomposition of the form $A = QHQ^{\mathsf{T}}$, with the first eigenvalue equal to 1 and its corresponding eigenvector equal to $\mathbb{1}/\sqrt{P}$.

Next, we define the extended centroid model $\mathcal{W}_{c,i} \overset{\Delta}{=} \left(\frac{1}{P}\mathbb{1}\mathbb{1}^{\mathsf{T}}\otimes I\right)\mathcal{W}_i$, and write:

$$
\begin{aligned}
\mathcal{W}_i - \mathcal{W}_{c,i} &= \left(I - \frac{1}{P}\mathbb{1}\mathbb{1}^{\mathsf{T}}\otimes I\right)\mathcal{W}_i \\
&= \left((Q^{\mathsf{T}}\otimes I)(Q\otimes I) - \frac{1}{P}\mathbb{1}\mathbb{1}^{\mathsf{T}}\otimes I\right)\mathcal{W}_i \\
&= (Q_\epsilon^{\mathsf{T}}\otimes I)(Q_\epsilon\otimes I)\mathcal{W}_i
\end{aligned}
$$

$$= (Q_\epsilon^\mathsf{T} \otimes I)H_\epsilon(Q_\epsilon \otimes I)\Bigg(\boldsymbol{\mathcal{W}}_{i-1} - \mu\text{col}\Big\{\nabla_{w^\mathsf{T}} J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{s}_{p,i} + \boldsymbol{q}_{p,i}\Big\}$$

$$+(Q_\epsilon^\mathsf{T} \otimes I)(Q_\epsilon \otimes I)\text{diag}\Big((A \otimes I)^\mathsf{T}\boldsymbol{\mathcal{G}}_i\Big) + \text{col}\Big\{\frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}} \boldsymbol{g}_{p,k,i}\Big\}\Bigg). \tag{3.75}$$

Then, taking the conditional expectation given the past models of $\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_i\|^2$, we can split the gradient noise and the added privacy noise from the model and the true gradient. Taking again the expectation over the past data, and then using the sub-multiplicity property of the norm followed by Jensen's inequality, we have:

$$\begin{aligned}
&\mathbb{E}\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_i\|^2 \\
&\leq \|H_\epsilon\|^2\Bigg(\mathbb{E}\left\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_{i-1} - (Q_\epsilon \otimes I)\mu\text{col}\Big\{\nabla_{w^\mathsf{T}} J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{q}_{p,i}\Big\}\right\|^2 \\
&\quad +\mu^2\|Q_\epsilon \otimes I\|^2\sum_{p=1}^P \mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 + \|Q_\epsilon \otimes I\|^2\sum_{p=1}^P \mathbb{E}\left\|\frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}} \boldsymbol{g}_{p,k,i}\right\|^2\Bigg) \\
&\quad + \|Q_\epsilon \otimes I\|^2\mathbb{E}\|\text{diag}\Big((A \otimes I)^\mathsf{T}\boldsymbol{\mathcal{G}}_i\Big)\|^2 \\
&\leq \|H_\epsilon\|^2\Bigg(\frac{1}{\|H_\epsilon\|}\mathbb{E}\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_{i-1}\|^2 + \frac{\mu^2\|Q_\epsilon \otimes I\|^2}{1-\|H_\epsilon\|}\sum_{p=1}^P \mathbb{E}\|\nabla_{w^\mathsf{T}} J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{q}_{p,i}\|^2 \\
&\quad + \mu^2\|Q_\epsilon \otimes I\|^2\sum_{p=1}^P \mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 + \|Q_\epsilon \otimes I\|^2\sum_{p=1}^P \mathbb{E}\left\|\frac{1}{L}\sum_{k\in\mathcal{L}_{p,i}} \boldsymbol{g}_{k,p,i}\right\|^2\Bigg) \\
&\quad + \|Q_\epsilon \otimes I\|^2\mathbb{E}\|\text{diag}\Big((A \otimes I)^\mathsf{T}\boldsymbol{\mathcal{G}}_i\Big)\|^2.
\end{aligned} \tag{3.76}$$

Next, we focus on each individual term. Using Jensen for some constant α and then the Lipschitz condition and the bound on the incremental noise, we can bound the below norm as follows:

$$\begin{aligned}
\mathbb{E}\|\nabla_{w^\mathsf{T}} J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{q}_{p,i}\|^2 \leq& \frac{2}{\alpha}\Big(\delta^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \|\nabla_{w^\mathsf{T}} J_p(w^o)\|^2\Big) \\
&+ \frac{1}{1-\alpha}\Big(O(\mu)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + O(\mu)\xi^2 + O(\mu^2)\sigma_{q,p}^2\Big).
\end{aligned} \tag{3.77}$$

Using the bound on the gradient noise (3.24), we get another $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2$ term, which can be bounded by the result in Theorem 3.3. Thus, we write:

$$
\begin{aligned}
&\frac{1}{1-\|H_\epsilon\|}\mathbb{E}\|\nabla_{w^\mathsf{T}} J_p(\boldsymbol{w}_{p,i-1}) + \boldsymbol{q}_{p,i}\|^2 + \mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 \\
&\leq \left(\frac{2\delta^2}{\alpha(1-\|H_\epsilon\|)} + \beta_{s,p}^2 + \frac{O(\mu)}{(1-\alpha)(1-\|H_\epsilon\|)}\right)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \frac{2\|\nabla_{w^\mathsf{T}} J_p(w^o)\|^2}{\alpha(1-\|H_\epsilon\|)} + \sigma_{s,p}^2 \\
&\quad + \frac{O(\mu)\xi^2 + O(\mu^2)\sigma_{q,p}^2}{(1-\alpha)(1-\|H_\epsilon\|)} \\
&\leq \left(\frac{2\delta^2}{\alpha(1-\|H_\epsilon\|)} + \beta_{s,p}^2 + \frac{O(\mu)}{(1-\alpha)(1-\|H_\epsilon\|)}\right)\Bigg(\lambda_p^i A^i[p]\mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\} \\
&\quad + \sum_{j=0}^{i-1}\lambda_p^j A^j[p]\mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2\Big\}\Bigg) + \frac{2\|\nabla_{w^\mathsf{T}} J_p(w^o)\|^2}{\alpha(1-\|H_\epsilon\|)} + \sigma_{s,p}^2 \\
&\quad + \frac{O(\mu)\xi^2 + O(\mu^2)\sigma_{q,p}^2}{(1-\alpha)(1-\|H_\epsilon\|)}.
\end{aligned} \tag{3.78}
$$

The noise term can be witten in a more compact way, $\|Q_\epsilon \otimes I\|^2 \sum_{p=1}^{P} \sigma_{g,p}^2$. Thus, putting everything together, we get:

$$
\begin{aligned}
&\mathbb{E}\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_i\|^2 \\
&\leq \|H_\epsilon\|\mathbb{E}\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_{i-1}\|^2 \\
&\quad + \mu^2\|Q_\epsilon \otimes I\|^2\|H_\epsilon\|^2\sum_{p=1}^{P}\Bigg(\left(\frac{2\delta^2}{\alpha(1-\|H_\epsilon\|)} + \beta_{s,p}^2 + \frac{O(\mu)}{(1-\alpha)(1-\|H_\epsilon\|)}\right) \\
&\quad \times \Bigg(\lambda_p^i A^i[p]\mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\} + \sum_{j=0}^{i-1}\lambda_p^j A^j[p]\mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2\Big\}\Bigg) \\
&\quad + \frac{2\|\nabla_{w^\mathsf{T}} J_p(w^o)\|^2}{\alpha(1-\|H_\epsilon\|)} + \sigma_{s,p}^2 + \frac{O(\mu)\xi^2 + O(\mu^2)\sigma_{q,p}^2}{(1-\alpha)(1-\|H_\epsilon\|)}\Bigg) + \|Q_\epsilon \otimes I\|^2\sum_{p=1}^{P}\sigma_{g,p}^2 \\
&\leq \|H_\epsilon\|^i\mathbb{E}\|(Q_\epsilon \otimes I)\boldsymbol{\mathcal{W}}_0\|^2 \\
&\quad + \sum_{j'=0}^{i-1}\|H_\epsilon\|^{j'+2}\|Q_\epsilon \otimes I\|^2\Bigg\{\mu^2\sum_{p=1}^{P}\Bigg(\left(\frac{2\delta^2}{\alpha(1-\|H_\epsilon\|)} + \beta_{s,p}^2 + \frac{O(\mu)}{(1-\alpha)(1-\|H_\epsilon\|)}\right) \\
&\quad \times \Bigg(\lambda_p^{j'} A^{j'}[p]\mathrm{col}\Big\{\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2\Big\} + \sum_{j=0}^{j'-1}\lambda_p^j A^j[p]\mathrm{col}\Big\{\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2\Big\}\Bigg) \\
&\quad + \frac{2\|\nabla_{w^\mathsf{T}} J_p(w^o)\|^2}{\alpha(1-\|H_\epsilon\|)} + \sigma_{s,p}^2 + \frac{O(\mu)\xi^2 + O(\mu^2)\sigma_{q,p}^2}{(1-\alpha)(1-\|H_\epsilon\|)}\Bigg) + \frac{1}{\|H_\epsilon\|^2}\sum_{p=1}^{P}\sigma_{g,p}^2\Bigg\}.
\end{aligned} \tag{3.79}
$$

Going back to the network disagreement, it is bounded by the above bound multiplied by $\|Q_\epsilon^\mathsf{T}\otimes I\|^2/P$. If we were to drive i to infinity, since $\|H_\epsilon\| = \iota_2 < 1$, with ι_2 being the second eigenvalue of A, and choosing $\alpha = \iota_2$ we would have:

$$
\begin{aligned}
&\limsup_{i\to\infty}\frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i}-\boldsymbol{w}_{c,i}\|^2\\
&\leq \frac{\|Q_\epsilon\otimes I\|^4\iota_2^2}{P}\Bigg\{\mu^2\sum_{p=1}^{P}\Bigg(\bigg(\frac{2\delta^2}{\iota_2(1-\iota_2)}+\beta_{s,p}^2+\frac{O(\mu)}{(1-\iota_2)^2}\bigg)\sum_{j'=0}^{\infty}\iota_2^{j'}\sum_{j=0}^{j'-1}\lambda_p^jA^j[p]\\
&\quad\times\operatorname{col}\Big\{\mu^2\sigma_{s,p}^2+O(\mu^2)\xi^2+O(\mu^3)\sigma_{q,p}^2+\sigma_{g,p}^2\Big\}+\frac{2\|\nabla_{w^\mathsf{T}}J_p(w^o)\|^2}{\iota_2(1-\iota_2)^2}+\frac{\sigma_{s,p}^2}{1-\iota_2}\\
&\quad+\frac{O(\mu)\xi^2+O(\mu^2)\sigma_{q,p}^2}{(1-\iota_2)^3}\Bigg)+\frac{1}{(1-\iota_2)\iota_2^2}\sum_{p=1}^{P}\sigma_{g,p}^2\Bigg\}\\
&\leq \frac{\iota_2^2}{P}\Bigg\{\mu^2\sum_{p=1}^{P}\Bigg(\bigg(\frac{2\delta^2}{\iota_2(1-\iota_2)}+\beta_{s,p}^2+\frac{O(\mu)}{(1-\iota_2)^2}\bigg)\\
&\quad\times\sum_{m\in\mathcal{N}_p}\frac{\iota_2(\mu^2\sigma_{s,m}^2+O(\mu^2)\xi^2+O(\mu^3)\sigma_{q,m}^2+\sigma_{g,m}^2)}{1-\iota_2\lambda_p a_{pm}}+\frac{2\|\nabla_{w^\mathsf{T}}J_p(w^o)\|^2}{\iota_2(1-\iota_2)^2}\\
&\quad+\frac{\sigma_{s,p}^2}{1-\iota_2}+\frac{O(\mu)\xi^2+O(\mu^2)\sigma_{q,p}^2}{(1-\iota_2)^3}\Bigg)+\frac{1}{(1-\iota_2)\iota_2^2}\sum_{p=1}^{P}\sigma_{g,p}^2\Bigg\}\\
&= \frac{\iota_2^2}{P(1-\iota_2)}\sum_{p=1}^{P}\mu^2\sigma_{s,p}^2+\frac{1}{\iota_2^2}\sigma_{g,p}^2+O(\mu)\sigma_{g,p}^2+O(\mu^3).
\end{aligned}
\tag{3.80}
$$

3.C Proof of Theorem 3.1

First taking the conditional mean of the ℓ_2-norm of the centroid error given the past models, splits the mean into three independent terms: the centralized recursion, the gradient noise and the added noise. Then, taking the expectation again, we get:

$$
\begin{aligned}
&\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2\\
&= \mathbb{E}\bigg\|\widetilde{\boldsymbol{w}}_{c,i-1}+\mu\frac{1}{P}\sum_{p=1}^{P}\nabla_{w^\mathsf{T}}J_p(\boldsymbol{w}_{p,i-1})+\mu\boldsymbol{q}_i\bigg\|^2+\mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2+\mathbb{E}\|\boldsymbol{g}_{c,i}\|^2\\
&\overset{(a)}{\leq}\frac{1}{\alpha^2}\mathbb{E}\bigg\|\widetilde{\boldsymbol{w}}_{c,i-1}+\mu\frac{1}{P}\sum_{p=1}^{P}\nabla_{w^\mathsf{T}}J_p(\boldsymbol{w}_{c,i-1})\bigg\|^2+\frac{\mu^2}{1-\alpha}\mathbb{E}\|\boldsymbol{q}_i\|^2\\
&\quad+\frac{\delta^2\mu^2}{\alpha(1-\alpha)P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1}-\boldsymbol{w}_{c,i-1}\|^2+\mu^2\mathbb{E}\|\boldsymbol{s}_i\|^2+\mathbb{E}\|\boldsymbol{g}_{c,i}\|^2
\end{aligned}
$$

$$\begin{aligned}
&\overset{(b)}{\le} \left(\frac{1}{\alpha^2}(1-2\nu\mu+\delta^2\mu^2) + \beta_s^2\mu^2 + \frac{O(\mu^3)}{1-\alpha} \right) \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\sigma_s^2 + \mathbb{E}\|\boldsymbol{g}_{c,i}\|^2 \\
&\quad + \left(\frac{\delta^2}{\alpha(1-\alpha)} + \frac{O(\mu^3)}{1-\alpha} + \beta_{s,max}^2 \right) \frac{\mu^2}{P} \sum_{p=1}^{P} \mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2 + \frac{O(\mu^3)\xi^2 + O(\mu^4)\sigma_q^2}{1-\alpha},
\end{aligned} \tag{3.81}$$

where inequality (a) follows from Jensen with constant $\alpha \in (0,1)$ and Lipshcitz, and (b) from applying Lemma 3.1. Then, choosing:

$$\alpha = \sqrt[4]{1-2\nu\mu+\delta^2\mu^2} = 1 - O(\mu), \tag{3.82}$$

the bound becomes:

$$\begin{aligned}
\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \le & \lambda_c \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\sigma_s^2 + \mathbb{E}\|\boldsymbol{g}_{c,i}\|^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_q^2 \\
& + \frac{O(\mu)}{P} \sum_{p=1}^{P} \mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2.
\end{aligned} \tag{3.83}$$

Finally, using the result on the network disagreement, recusrively bounding the error, and taking the limit of i, we get the final result:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \le \frac{\mu^2\sigma_s^2 + \mathbb{E}\|\boldsymbol{g}_c\|^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_q^2}{1-\lambda_c} + \sum_{p=1}^{P} O(1)\sigma_{g,p}^2 + O(\mu). \tag{3.84}$$

3.D Secondary Result on the Extended Model Error

To show the sensitivity of the algorithm is bounded with high probability, we require a bound on $\mathbb{E}\|\widetilde{\mathcal{W}}_i\|^2$ and $\mathbb{E}\|\widetilde{\mathcal{W}}_i'\|^2$. From Theorem 3.3 we can bound the individual errors by:

$$\begin{aligned}
\mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2 &\le \lambda_p \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2 \\
&\le \lambda_{\max} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i-1}\|^2 + \mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2 \\
&\le \lambda_{\max}^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2 + \frac{1-\lambda_{\max}^i}{1-\lambda_{\max}} \left(\mu^2\sigma_{s,p}^2 + O(\mu^2)\xi^2 + O(\mu^3)\sigma_{q,p}^2 + \sigma_{g,p}^2 \right) \\
&\le \lambda_{\max}^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2 + O(\mu) + O(\mu^{-1}),
\end{aligned} \tag{3.85}$$

where $\lambda_{\max} = \max_p \lambda_p$. Then, $\mathbb{E}\|\widetilde{\mathcal{W}}_i\|^2$ can be bounded as follows:

$$\begin{aligned}\mathbb{E}\|\widetilde{\mathcal{W}}_i\|^2 &= \sum_{p=1}^{P} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,i}\|^2 \\ &\leq \sum_{p=1}^{P} \lambda_{\max}^i \mathbb{E}\|\widetilde{\boldsymbol{w}}_{p,0}\|^2 + O(\mu) + O(\mu^{-1}) \\ &= \lambda_{\max}^i \mathbb{E}\|\widetilde{\mathcal{W}}_0\|^2 + O(\mu) + O(\mu^{-1}). \end{aligned} \tag{3.86}$$

It follows that for some constants B and B', the probability that $\mathbb{E}\|\widetilde{\mathcal{W}}_i\|$ and $\mathbb{E}\|\widetilde{\mathcal{W}}'_i\|$ are unbounded can be bounded using Markov's inequality by:

$$\begin{aligned}\mathbb{P}(\|\widetilde{\mathcal{W}}_i\| \geq B) &\leq \frac{\mathbb{E}\|\widetilde{\mathcal{W}}_i\|^2}{B^2} \\ &\leq \frac{\lambda_{\max}^i \mathbb{E}\|\widetilde{\mathcal{W}}_0\|^2 + O(\mu) + O(\mu^{-1})}{B^2}, \end{aligned} \tag{3.87}$$

and similarly for $\mathbb{P}(\|\widetilde{\mathcal{W}}'_i\| \geq B')$.

3.E Proof of Theorem 3.2

To evaluate the probability distribution in Definition 3.1, we note that the randomness of the models $\boldsymbol{\psi}_{p,j}$ arises from the subsampling of the data for the calculation of the stochastic gradient at each iteration. Thus, given the subsampled dataset, the models are now deterministic and since the added noises $\boldsymbol{g}_{pm,j}$ are Laplacian random variables, the distribution of the added noise over the neighbourhood of agent p and over the iterations is given by:

$$\begin{aligned} f\left(\left\{\{\boldsymbol{\psi}_{p,j} + \boldsymbol{g}_{pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=0}^{i}\right) &= f(\boldsymbol{y}_0)f(\boldsymbol{y}_1|\boldsymbol{y}_0)\cdots f(\boldsymbol{y}_i|\boldsymbol{y}_0,\cdots,\boldsymbol{y}_{i-1}) \\ &= \prod_{j=0}^{i} \frac{1}{\sqrt{2}\sigma_g} \exp\left(-\frac{\sqrt{2}}{\sigma_g}\|\boldsymbol{\psi}_{p,j} + \boldsymbol{g}_{p,j}\|\right) \\ &= \frac{1}{\sqrt{2}\sigma_g} \exp\left(-\frac{\sqrt{2}}{\sigma_g}\sum_{j=0}^{i}\|\boldsymbol{\psi}_{p,j} + \boldsymbol{g}_{p,j}\|\right), \end{aligned} \tag{3.88}$$

where $\boldsymbol{y}_j = \{\boldsymbol{\psi}_{p,j} + \boldsymbol{g}_{pm,j}\}_{m \in \mathcal{N}_p \setminus \{p\}}$ and the ratio in Definition 3.1 is bounded with high probability:

$$\begin{aligned}
&\exp\left(-\frac{\sqrt{2}}{\sigma_g}\sum_{j=0}^{i}\Big(\|\boldsymbol{\psi}_{p,j}+\boldsymbol{g}_{p,j}\| - \|\boldsymbol{\psi}'_{p,j}+\boldsymbol{g}_{p,j}\|\Big)\right)\\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g}\sum_{j=0}^{i}\|\boldsymbol{\psi}_{p,j}-\boldsymbol{\psi}'_{p,j}\|\right)\\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g}\sum_{j=0}^{i}\Delta(j)\right)\\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g}\sum_{j=0}^{i}(B+B'+\sqrt{P}\|w^o - w'^o\|)\right)\\
&= \exp\left(\frac{\sqrt{2}}{\sigma_g}(B+B'+\sqrt{P}\|w^o - w'^o\|)(i+1)\right),
\end{aligned} \tag{3.89}$$

where the inequalities follow from the triangle inequality and the bound on the sensitivity of the algorithm.

4 Privacy in Decentralized Learning

In chapter 3 we examined a setting combining both decentralized and distributed learning. We suggested a privatization scheme for the decentralized layer that improves the effect of the added noise on the mean-square error from what was previously $O(\mu^{-1})$ to what is now $O(1)$. In this chapter, we focus on the more general decentralized setting and show how to further reduce the effect of the added noise for privacy down to $O(\mu)$. The material in this chapter is based on the work in [106].

4.1 Introduction

Decentralized learning and optimization strategies are relevant in many contexts in real-world problems, such as in the design of robotic swarms for rescue missions, or the design of cloud computing services, or the exchange of information over social networks. Even in scenarios where a centralized solution is possible, it is often preferable to rely on a decentralized implementation for various reasons. For instance, the centralized solution tends to have high maintenance costs and is sensitive to the failure of the central processor. Agents may also be reluctant to share their data with a remote central processor due to privacy and safety considerations. The amount of data available at each agent may be significant in size, which makes it difficult to regularly transmit large amounts of data between the dispersed agents and the central processor. Decentralized implementations offer an attractive and robust alternative. The architecture can tolerate the failure of individual agents since processing can continue to occur among the remaining agents. Also, agents are only required to share minimal processed information with their neighbours.

There exist several schemes for decentralized optimization, which have been studied extensively in the literature. Among these schemes we list the incremental strategy [107–114], consensus strategy [23, 35, 115–122], and diffusion strategy [9, 25, 123–128, 128, 129]. The incremental algorithm requires a renumbering of the agents over a cyclic path to cover the entire graph. This is usually a challenging task since the determination of an appropriate cycle is an NP-hard problem and, moreover, the failure of any edge along the path turns the solution moot. The consensus and diffusion strategies avoid the need for a circular path over the graph. They rely on the local sharing of information among neighbouring agents. One main difference between both classes of strategies is that consensus updates are asymmetrical, where the starting point for the gradient-descent step is different from the location where the gradient vector is evaluated — see expression (4.16). It was shown in several earlier works (see, e.g., [9, 25, 124]) that this asymmetry reduces the stability range of consensus implementations in comparison to diffusion solutions, especially in scenarios involving the need for continuous learning and adaptation.

Now, one key aspect of decentralized architectures is that they require agents to share information with their neighbours. This aspect raises an important privacy question about whether the information that is being shared over the edges in the graph can be intercepted. For instance, it is known that in algorithms that rely on gradient-descent updates, information leakage can occur through the sharing of the local gradients or the models that they estimate [82–85]. This can be problematic when the network is dealing with classified or sensitive data such as healthcare or financial data. In such cases, attackers may be able to recover certain elements of an individual's personal information. There is no question that it is useful to pursue decentralized strategies that guarantee a certain level of privacy.

There exists several useful works in the literature that address privacy questions for decentralized algorithms. These contributions rely mainly on two types of tools: differential privacy or cryptography. Cryptographic methods range from using secure aggregation to multiparty computation and homomorphic encryption [99–102, 130]. Although these methods do not hinder the performance of the learned model, they add significant computational and communication overhead.

On the other hand, differentially private methods mask the messages by adding some random noise [46, 86, 90–97, 131]. They are simple to implement, but they introduce errors into the learned model and reduce the overall utility of the network. One main reason for this degradation is that the noise is often added at will *without* accounting for the graph topology.

In this work, we focus on differential privacy since it is simpler to apply and more scalable. We explain how to adjust its application to match the graph topology, while ensuring privacy and performance guarantees. In particular, we examine the effect of two differentially private schemes: the traditional random perturbations scheme and a graph-homomorphic scheme. We establish the superiority of the latter over the former in the mean-square error (MSE) sense. We also devise a third scheme, called *local* graph-homormphic processing, which fully removes the degrading effect of the noise on performance. These results apply to a broad class of decentralized learning and optimization formulations.

4.2 Problem Setup

We consider a graph topology with P agents, labelled $p = 1, 2, \dots, P$, as illustrated in Figure 4.1. The objective is for the agents to approach the minimizer of an aggregate convex optimization problem of the form:

$$w^o \triangleq \underset{w\in\mathbb{R}^M}{\operatorname{argmin}} \frac{1}{P}\sum_{p=1}^{P}\left\{ J_p(w) \triangleq \frac{1}{N_p}\sum_{n=1}^{N_p} Q_p(w; x_{p,n})\right\}, \tag{4.1}$$

where the risk function $J_p(\cdot)$ is associated with the pth agent and is defined as an empirical average of the corresponding loss function $Q_p(\cdot;\cdot)$ evaluated at the local data $\{x_{p,n}\}_{n=1}^{N_p}$. We associate two non-negative weights a_{mp} and a_{pm} with the edge linking neighbouring agents m and p. In this notation, a_{mp} is the weight used by agent p to scale information arriving from m, and similarly for a_{pm}; it scales information from p toward m. The neighbourhood of an agent p is denoted by $\mathcal{N}_p$ and consists of all agents that are connected to p by an edge. We assume that $\mathcal{N}_p$ includes agent p as well.

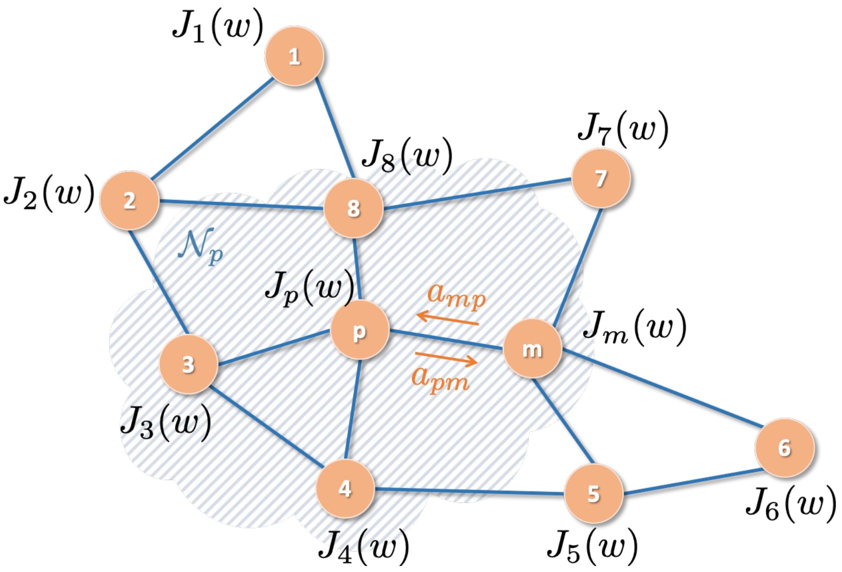


Figure 4.1 – Illustration of a network of agents.

4.2.1 Modeling Conditions

We assume the individual risk functions $J_p(w)$ are strongly convex and the loss functions $Q_p(w;\cdot)$ have Lipschitz continuous gradients and are twice differentiable. These conditions are common in the study of decentralized methods. Although the conditions can be relaxed and the results extended to broader scenarios (see, e.g., [9, 17, 25, 132, 133]), it is sufficient for the purposes of this work to illustrate the main ideas under these assumptions.

Assumption 4.1 (Convexity and smoothness). *The risks $J_p(\cdot)$ are $\nu-$strongly convex, and the losses $Q_p(\cdot;\cdot)$ are convex and twice differentiable, namely for some $\nu > 0$:*

$$J_p(w_2) \geq \; J_p(w_1) + \nabla_{w^\mathsf{T}} J_p(w_1)(w_2 - w_1) + \frac{\nu}{2}\|w_2 - w_1\|^2, \tag{4.2}$$

$$Q_p(w_2;\cdot) \geq \; Q_p(w_1;\cdot) + \nabla_{w^\mathsf{T}} Q_p(w_1;\cdot)(w_2 - w_1). \tag{4.3}$$

The loss functions have $\delta-$Lipschitz continuous gradients, meaning there exists $\delta > 0$ such that for any data point $x_{p,n}$:

$$\|\nabla_{w^\mathsf{T}} Q_p(w_2; x_{p,n}) - \nabla_{w^\mathsf{T}} Q_p(w_1; x_{p,n})\| \leq \delta \|w_2 - w_1\|. \tag{4.4}$$

Since we assume the loss functions are twice differentiable, then the above strong-convexity and Lipschitz continuity conditions are equivalent to (see [9, 17, 25]):

$$0 < \nu I \leq \nabla^2_{w^\mathsf{T}} J_p(w) \leq \delta I. \tag{4.5}$$

We further assume that the graph topology is strongly connected. This means that there exist paths linking any arbitrary pair of agents (m, p) in both directions and, moreover, at least one agent p in the network has a self-loop with $a_{pp} > 0$. In other words, at least one agent has some trust in its local information. The combination matrix $A = [a_{mp}]$ is usually left-stochastic meaning that its entries satisfy:

$$a_{mp} \geq 0, \qquad \sum_{m \in \mathcal{N}_p} a_{mp} = 1. \tag{4.6}$$

That is, the weights on edges connecting agents are nonnegative, and the entries on each column of A add up to one. The strong connectedness of the graph translates into guaranteeing that A is a primitive matrix. As a result, it follows from the Peron-Frobenius theorem [9] that A will have a single eigenvalue at one, while all other eigenvalues are strictly inside the unit circle. Moreover, an eigenvector q will exist with positive entries $\{q_p\}$ adding up to one and satisfying:

$$Aq = q, \quad q_p > 0, \quad \mathbb{1}^\mathsf{T} q = 1. \tag{4.7}$$

We refer to q as the Peron eigenvector of A. Furthermore, it holds that $\rho\left(A - q\mathbb{1}^\mathsf{T}\right) < 1$, where $\rho(\cdot)$ denotes the spectral radius of its matrix argument.

Next, let w^o denote the global minimizer for (4.1) and let w_p^o denote the local minimizer for $J_p(\cdot)$. We assume that the difference between these global and local models is bounded since, otherwise, collaboration would not be beneficial and one would instead follow a different optimization approach such as multi-task learning [134].

To clarify this point further, we consider a simple example involving a quadratic loss. Assume the data arriving at node p, denoted by $\boldsymbol{d}_p(n)$, is generated by some linear regression model under additive noise of the form:

$$\boldsymbol{d}_p(n) = \boldsymbol{u}_{p,n}^{\mathsf{T}} w^\star + \boldsymbol{o}_p(n), \tag{4.8}$$

where $\boldsymbol{u}_{p,n}$ is the feature vector and $w^\star$ is the model. We can seek to estimate $w^\star$ by solving:

$$\min_w \frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_p}\sum_{n=1}^{N_p}(\boldsymbol{d}_p(n) - \boldsymbol{u}_{p,n}^{\mathsf{T}} w)^2. \tag{4.9}$$

The global minimizer in this case is given by:

$$w^o = w^\star + \widehat{R}_u^{-1}\widehat{r}_{uo}, \tag{4.10}$$

where:

$$\widehat{R}_u \triangleq \frac{1}{P}\sum_{p=1}^{P}\left\{\widehat{R}_{p,u} \triangleq \frac{1}{N_p}\sum_{n=1}^{N_p}\boldsymbol{u}_{p,n}\boldsymbol{u}_{p,n}^{\mathsf{T}}\right\}, \tag{4.11}$$

$$\widehat{r}_{uo} \triangleq \frac{1}{P}\sum_{p=1}^{P}\left\{\widehat{r}_{p,uo} \triangleq \frac{1}{N_p}\sum_{n=1}^{N_p}\boldsymbol{o}_p(n)\boldsymbol{u}_{p,n}\right\}, \tag{4.12}$$

while the local minimizers of $J_p(w)$ are given by:

$$w_p^o = w^\star + \widehat{R}_{p,u}^{-1}\widehat{r}_{p,uo}. \tag{4.13}$$

Thus, the global model (4.10) can be written as a weighted average of the local models (4.13):

$$w^o = \frac{1}{P}\sum_{p=1}^{P}\widehat{R}_u^{-1}\widehat{R}_{p,u} w_p^o. \tag{4.14}$$

This implies that the global model is a mixture of the local models. Therefore, the bound imposed below on the model difference amounts to an assumption on how different the distributions of the data across the agents are. This condition is weaker than a uniform bound on the difference between the gradients of the cost functions, which is more commonly assumed in the literature (see [135, 136]).

Assumption 4.2 (**Model drifts**). *The distance of each local model w_p^o to the global model w^o is uniformly bounded, i.e., there exists $\xi \geq 0$ such that $\|w^o - w_p^o\| \leq \xi$.*

4.3 Decentralized Learning

4.3.1 Generalized Decentralized Learning

We focus on two main strategies: consensus and diffusion. The consensus strategy for solving (1) takes the form:

$$\boldsymbol{\psi}_{p,i-1} = \sum_{m \in \mathcal{N}_p} a_{mp} \boldsymbol{w}_{m,i-1}, \tag{4.15}$$

$$\boldsymbol{w}_{p,i} = \boldsymbol{\psi}_{p,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_p}(\boldsymbol{w}_{p,i-1}), \tag{4.16}$$

where $\widehat{\nabla_{w^\mathsf{T}} J_p}(\cdot)$ denotes a stochastic gradient *approximation* for the true gradient of $J_p(\cdot)$. Usually, the approximation is taken as the gradient of the loss function, namely, $\nabla_{w^\mathsf{T}} Q_p(\boldsymbol{w}_{p,i-1}, \boldsymbol{x}_{p,i})$. Here, the quantities $\{\boldsymbol{\psi}_{p,i}, \boldsymbol{w}_{p,i}\}$ denote estimates for w^o at node p at time i. Observe that the gradient vector in (4.16) is evaluated at the prior local model $\boldsymbol{w}_{p,i-1}$ and not at the intermediate model $\boldsymbol{\psi}_{p,i-1}$. The diffusion strategy, in turn, admits two related implementations known as combine-then-adapt (CTA) and adapt-then-combine (ATC). They differ by the order in which the calculations are performed with combination coming before adaptation in one case, and with the order reversed in the other case. The CTA diffusion strategy is described by:

$$\boldsymbol{\psi}_{p,i-1} = \sum_{m \in \mathcal{N}_p} a_{mp} \boldsymbol{w}_{m,i-1}, \tag{4.17}$$

$$\boldsymbol{w}_{p,i} = \boldsymbol{\psi}_{p,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_p}(\boldsymbol{\psi}_{p,i-1}). \tag{4.18}$$

Comparing with (4.15)–(4.16), observe now that the starting point in (4.18) for the gradient-descent step is the same as the point where the gradient vector is evaluated. Similarly, the ATC diffusion strategy is given by:

$$\boldsymbol{\psi}_{p,i} = \boldsymbol{w}_{p,i-1} - \mu \widehat{\nabla_{w^\mathsf{T}} J_p}(\boldsymbol{w}_{p,i-1}), \tag{4.19}$$

$$\boldsymbol{w}_{p,i} = \sum_{m \in \mathcal{N}_p} a_{mp} \boldsymbol{\psi}_{m,i}. \tag{4.20}$$

The above three algorithms can be combined into a single general description as follows [9]:

$$\boldsymbol{\phi}_{p,i-1} = \sum_{m\in\mathcal{N}_p} a_{1,mp}\boldsymbol{w}_{m,i-1}, \tag{4.21}$$

$$\boldsymbol{\psi}_{p,i} = \sum_{m\in\mathcal{N}_p} a_{0,mp}\boldsymbol{\phi}_{m,i-1} - \mu\widehat{\nabla_{w^\mathsf{T}} J_p}(\boldsymbol{\phi}_{p,i-1}), \tag{4.22}$$

$$\boldsymbol{w}_{p,i} = \sum_{m\in\mathcal{N}_p} a_{2,mp}\boldsymbol{\psi}_{m,i}. \tag{4.23}$$

where we are introducing three combination matrices, $\{A_0, A_1, A_2\}$. By setting $A_0 = A$ and $A_1 = A_2 = I$, we obtain consensus, while $A_1 = A$ and $A_0 = A_2 = I$ leads to CTA diffusion, and $A_2 = A$ and $A_0 = A_1 = I$ leads to ATC diffusion. Other choices are possible.

4.3.2 Privacy Learning

We now examine differentially private algorithms to safeguard the privacy of the information that is shared among the agents. For illustration purposes, assume the data $\{x_{1,n}\}$ at agent 1 is replaced by a different set $\{x'_{1,n}\}$. The algorithm will thus take a new trajectory, which we denote by $\{\boldsymbol{\phi}'_{p,i-1}, \boldsymbol{\psi}'_{p,i}, \boldsymbol{w}'_{p,i}\}$. In a private implementation, an external observer should be oblivious to this change at agent 1. Concretely, all we need to do is add noise to the messages that need privatization. Most commonly, noise with exponential distributions, such as Laplacian or Gaussian, is added [131]. Thus, we are motivated initially to consider a privatized decentralized implementation of the following form:

$$\boldsymbol{\phi}_{p,i-1} = \sum_{m\in\mathcal{N}_p} a_{1,mp}\left(\boldsymbol{w}_{m,i-1} + \boldsymbol{g}_{1,mp,i}\right), \tag{4.24}$$

$$\boldsymbol{\psi}_{p,i} = \sum_{m\in\mathcal{N}_p} a_{0,mp}\left(\boldsymbol{\phi}_{m,i-1} + \boldsymbol{g}_{0,mp,i}\right) - \mu\widehat{\nabla_{w^\mathsf{T}} J_p}(\boldsymbol{\phi}_{p,i-1}), \tag{4.25}$$

$$\boldsymbol{w}_{p,i} = \sum_{m\in\mathcal{N}_p} a_{2,mp}\left(\boldsymbol{\psi}_{m,i} + \boldsymbol{g}_{2,mp,i}\right), \tag{4.26}$$

where the $\boldsymbol{g}_{j,mp,i}$ denote zero-mean Laplacian random noises for $j = 0, 1, 2$ for every $m, p = 1, 2, \cdots, P$. For example, in (4.24), agent m shares $\boldsymbol{w}_{m,i-1}$ with agent p over the edge that links them. During this transmission, an amount of Laplacian noise $\boldsymbol{g}_{1,mp,i}$ is added. The subscript mp is used to denote that this noise is for the directed communication from m to p. Similarly, for the other noises.

We next define differential privacy formally [131], and show that the above algorithm is indeed differentially private.

Definition 4.1 ($\epsilon(i)-$**Differential privacy**). *We say that the algorithm given by* (4.24)$-$(4.26) *is $\epsilon(i)-$differentially private for agent p at time i if the following condition on the probabilities for observing the respective events holds on the joint distribution $f(\cdot)$ where the notation $\boldsymbol{y}_{p,j-1}$ represents any of the shared messages $\{\boldsymbol{w}_{p,j-1}, \boldsymbol{\phi}_{p,j}, \boldsymbol{\psi}_{p,j}\}$, $\boldsymbol{g}_{\cdot,pm,j}$ the corresponding added noise $\{\boldsymbol{g}_{1,pm,j}, \boldsymbol{g}_{0,pm,j}, \boldsymbol{g}_{2,pm,j}\}$:*

$$\frac{f\left(\left\{\{\boldsymbol{y}_{p,j-1} + \boldsymbol{g}_{\cdot,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=1}^{i}\right)}{f\left(\left\{\{\boldsymbol{y}'_{p,j-1} + \boldsymbol{g}_{\cdot,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=1}^{i}\right)} \leq e^{\epsilon(i)}. \tag{4.27}$$

The above bounds ensure that for small $\epsilon(i)$, the distributions of the original and modified trajectories are close to each other. This makes it difficult to infer information about the data at the agents since we cannot distinguish the trajectories of the algorithm for different combinations of participating agents. In other words, if agent 1 chooses to replace its original dataset by $\{x'_{1,n}\}$, then the resulting models $\{\boldsymbol{w}'_{p,j-1}, \boldsymbol{\phi}'_{p,j}, \boldsymbol{\psi}'_{p,j}\}$ are close enough in distribution to the original models $\{\boldsymbol{w}_{p,j-1}, \boldsymbol{\phi}_{p,j}, \boldsymbol{\psi}_{p,j}\}$, and an outside observer will not be able to conclude what dataset was used. The two model trajectories resulting from the use of the original and the alternative dataset are indistinguishable.

To show that algorithm (4.24)$-$(4.26) satisfies condition (4.27), we first calculate the sensitivity of the algorithm. The sensitivity at time i is defined in Appendix 4.A as the change in the trajectory of the algorithm if instead of using the original dataset, agent 1 uses the alternative dataset $\{x'_{1,n}\}$. In Appendix 4.A the sensitivity is shown to satisfy:

$$\Delta(i) \overset{\Delta}{=} \|\boldsymbol{\mathcal{W}}_i - \boldsymbol{\mathcal{W}}'_i\| \leq B + B' + \sqrt{P}\|w^o - w'^o\|, \tag{4.28}$$

for some constants B and B' and with high probability. That is, it holds that:

$$\mathbb{P}\left(\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\|\right) \geq \left(1 - \frac{\kappa_2^2 \mathbb{1}^\mathsf{T}\Gamma^i \begin{bmatrix}\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_0\|^2\end{bmatrix} + O(\mu) + O(\mu^{-1})}{B^2}\right)$$
$$\times \left(1 - \frac{\kappa_2'^2 \mathbb{1}^\mathsf{T}\Gamma'^i \begin{bmatrix}\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}'_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}'_0\|^2\end{bmatrix} + O(\mu) + O(\mu^{-1})}{B'^2}\right), \tag{4.29}$$

where $\mathcal{W}_i = \text{col}\Big\{\boldsymbol{w}_{p,i}\Big\}_{p=1}^{P}$, the model error at time zero is denoted by:

$$\widetilde{\mathcal{W}}_0 = \text{col}\Big\{w^o - \boldsymbol{w}_{p,0}\Big\}_{p=1}^{P}, \tag{4.30}$$

and the variables $\{\bar{\mathcal{W}}_0, \check{\mathcal{W}}_0\}$ arise from the partitioning:

$$\mathcal{V}_\theta^\mathsf{T} \widetilde{\mathcal{W}}_0 = \text{col}\{\bar{\mathcal{W}}_0, \check{\mathcal{W}}_0\}, \tag{4.31}$$

with the matrix Γ and the constant κ_2 defined in Appendix 4.B. Result (4.29) means that the sensitivity $\Delta(i)$ is bounded with high probability. The bound constants B and B' are chosen by the user: larger values for B and B' result in higher probability of bounded sensitivity but, as shown in (4.34), they result in a larger privacy bound. In other words, the values of B and B' can be controlled to balance the trade-off between the privacy level and the likelihood of bounded sensitivity. Next, if we denote the variance of the Laplacian noise $\boldsymbol{g}_{j,mp,i}$ by σ_g^2, with $\boldsymbol{y}_{p,j-1} = \boldsymbol{w}_{p,j-1}$ the fraction in (4.27) can be bounded as follows with high probability:

$$\begin{aligned}
\frac{f\left(\left\{\{\boldsymbol{w}_{p,j-1} + \boldsymbol{g}_{0,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=1}^{i}\right)}{f\left(\left\{\{\boldsymbol{w}'_{p,j-1} + \boldsymbol{g}_{0,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{j=1}^{i}\right)} &\overset{(a)}{=} \prod_{j=1}^{i} \frac{f\left(\{\boldsymbol{w}_{p,j-1} + \boldsymbol{g}_{0,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}} | \mathcal{X}_{j-1}\right)}{f\left(\{\boldsymbol{w}'_{p,j-1} + \boldsymbol{g}_{0,pm,j}\}_{m\in\mathcal{N}_p\setminus\{p\}} | \mathcal{X}'_{j-1}\right)} \\
&\overset{(b)}{=} \prod_{j=1, m\in\mathcal{N}_p\setminus\{p\}}^{i} \frac{\exp\left(-\sqrt{2}\|\boldsymbol{w}_{p,j-1} + \boldsymbol{g}_{0,pm,j}\|/\sigma_g\right)}{\exp\left(-\sqrt{2}\|\boldsymbol{w}'_{p,j-1} + \boldsymbol{g}_{0,pm,j}\|/\sigma_g\right)} \\
&\leq \exp\left(\frac{\sqrt{2}}{\sigma_g} \sum_{j=1, m\in\mathcal{N}_p\setminus\{p\}}^{i} \|\boldsymbol{w}_{p,j-1} - \boldsymbol{w}'_{p,j-1}\|\right) \\
&\leq \exp\left(\frac{\sqrt{2}P}{\sigma_g} \sum_{j=1}^{i} \|\mathcal{W}_{j-1} - \mathcal{W}'_{j-1}\|\right),
\end{aligned} \tag{4.32}$$

where the first equality (a) follows from applying Bayes' rule with:

$$\mathcal{X}_{j-1} \overset{\Delta}{=} \{\boldsymbol{w}_{p,j-1}\} \cup \left\{\{\boldsymbol{w}_{p,o-1} + \boldsymbol{g}_{0,pm,o}\}_{m\in\mathcal{N}_p\setminus\{p\}}\right\}_{o=1}^{j-1}, \tag{4.33}$$

and the second equality (b) follows from the independence of $\boldsymbol{w}_{p,j-1} + \boldsymbol{g}_{0,pm,j}$ for $m \in \mathcal{N}_p \setminus \{p\}$ conditioned on $\boldsymbol{w}_{p,j-1}$. A similar bound can be found for $\boldsymbol{y}_{p,j-1} \in \{\boldsymbol{\phi}_{p,j}, \boldsymbol{\psi}_{p,j}\}$.

Thus, the level of privacy is defined by the following choice for $\epsilon(i)$ in terms of the running $\Delta(j)$ values:

$$\epsilon(i) = \frac{\sqrt{2}P}{\sigma_g} \sum_{j=0}^{i-1} \Delta(j) \leq \frac{\sqrt{2}P}{\sigma_g}(B + B' + \sqrt{P}\|w^o - w'^o\|)i. \tag{4.34}$$

These results show that in order to arrive at an $\epsilon(i)-$differentially private algorithm, it is sufficient to select the variance of the Laplacian noise to satisfy (4.34). Expression (4.34) shows that $\epsilon(i)$ is a linear function of the iterations. This means that the process becomes less private at a rate no greater than a linear rate. It is important to note here that most earlier studies on differentially private schemes for multi-agent systems [90, 93, 94] assume bounded gradients for the risk function. However, this condition is rarely satisfied in practice. For instance, even quadratic risks have unbounded gradients. For this reason, in our approach, we have avoided relying on this assumption. Instead, we are able to establish that differential privacy continues to hold with high probability.

We still need to examine the effect of the added noises on performance. To do so, we introduce the extended model $\boldsymbol{\mathcal{W}}_i \overset{\Delta}{=} \text{col}\Big\{\boldsymbol{w}_{p,i}\Big\}_{p=1}^P$ and write the three-step algorithm (4.24)–(4.26) using one single recursion as follows:

$$\begin{aligned}\boldsymbol{\mathcal{W}}_i =& \mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{W}}_{i-1} - \mu\mathcal{A}_2^\mathsf{T}\text{col}\Big\{\widehat{\nabla_{w^\mathsf{T}}J_p}(\boldsymbol{\phi}_{p,i-1})\Big\}_{p=1}^P + \mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\text{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i}) \\ &+ \mathcal{A}_2^\mathsf{T}\text{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i}) + \text{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i}), \end{aligned} \tag{4.35}$$

where for $j = 0, 1, 2$, $\mathcal{A}_j \overset{\Delta}{=} A_j \otimes I_M$, and $\boldsymbol{\mathcal{G}}_{j,i}$ is a matrix whose entries are the added noises $\boldsymbol{g}_{j,mp,i}$. We denote the model error by:

$$\widetilde{\boldsymbol{\mathcal{W}}}_i \overset{\Delta}{=} \text{col}\Big\{w^o - \boldsymbol{w}_{p,i}\Big\}_{p=1}^P, \tag{4.36}$$

and introduce the local gradient noise:

$$\boldsymbol{s}_{p,i} \overset{\Delta}{=} \widehat{\nabla_{w^\mathsf{T}}J_p}(\boldsymbol{\phi}_{p,i-1}) - \nabla_{w^\mathsf{T}}J_p(\boldsymbol{\phi}_{p,i-1}). \tag{4.37}$$

It is customary to assume that this gradient noise process has zero mean and bounded second-order moment (see, e.g., [9, 17], where this property is actually shown to hold in many important cases of interest and similar arguments can be applied to the current case), namely:

$$\mathbb{E}\{\|\boldsymbol{s}_{p,i}\|^2|\mathcal{F}_{i-1}\} \leq \beta_{s,p}^2\|\widetilde{\boldsymbol{\phi}}_{p,i-1}\|^2 + \sigma_{s,p}^2, \tag{4.38}$$

for some nonnegative constants $\beta_{s,p}^2$ and $\sigma_{s,p}^2$, and where the conditioning is taken over all past models:

$$\mathcal{F}_{i-1} \overset{\Delta}{=} \text{filtration}\{\boldsymbol{w}_{p,j}\}_{p=1,j=0}^{P,i-1}. \tag{4.39}$$

Then, using the extended gradient noise:

$$\boldsymbol{s}_i \overset{\Delta}{=} \text{col}\Big\{\boldsymbol{s}_{p,i}\Big\}_{p=1}^P, \tag{4.40}$$

the error recursion corresponding to (4.35) is given by:

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{W}}}_i =& \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_0^{\mathsf{T}}\mathcal{A}_1^{\mathsf{T}}\widetilde{\boldsymbol{\mathcal{W}}}_{i-1} + \mu\mathcal{A}_2^{\mathsf{T}}\text{col}\Big\{\nabla_{w^{\mathsf{T}}}J_p(\boldsymbol{\phi}_{\boldsymbol{p,i-1}})\Big\}_{p=1}^{P} + \mu\mathcal{A}_2^{\mathsf{T}}\boldsymbol{s}_i - \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_0^{\mathsf{T}}\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i})\\ &- \mathcal{A}_2^{\mathsf{T}}\text{diag}(\mathcal{A}_0^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{0,i}) - \text{diag}(\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{2,i}). \qquad (4.41)\end{aligned}$$

Since $J_p(\cdot)$ are twice differentiable, we appeal to the mean-value theorem to express the gradient in the form [9]:

$$\nabla_{w^{\mathsf{T}}}J_p(\boldsymbol{\phi}_{p,i-1}) = -\boldsymbol{H}_{p,i-1}\widetilde{\boldsymbol{\phi}}_{p,i-1} - \nabla_{w^{\mathsf{T}}}J_p(w^o), \qquad (4.42)$$

where:

$$\boldsymbol{H}_{p,i-1} \triangleq \int_0^1 \nabla^2_{w^{\mathsf{T}}}J_p(w^o - t\boldsymbol{\phi}_{p,i-1})dt. \qquad (4.43)$$

Then, introducing the quantities:

$$\begin{aligned}\boldsymbol{\mathcal{B}}_{i-1} &\triangleq \mathcal{A}_2^{\mathsf{T}}(\mathcal{A}_0^{\mathsf{T}} - \mu\boldsymbol{\mathcal{H}}_{i-1})\mathcal{A}_1^{\mathsf{T}}, &(4.44)\\ \boldsymbol{\mathcal{H}}_{i-1} &\triangleq \text{diag}\{\boldsymbol{H}_{p,i-1}\}_{p=1}^{P}, &(4.45)\\ b &\triangleq \text{col}\Big\{\nabla_{w^{\mathsf{T}}}J_p(w^o)\Big\}_{p=1}^{P}, &(4.46)\end{aligned}$$

we rewrite (4.41) as:

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{W}}}_i =& \boldsymbol{\mathcal{B}}_{\boldsymbol{i-1}}\widetilde{\boldsymbol{\mathcal{W}}}_{i-1} + \mu\mathcal{A}_2^{\mathsf{T}}\boldsymbol{s}_i - \mu\mathcal{A}_2^{\mathsf{T}}b + \mu\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{H}}_{i-1}\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i}) - \text{diag}(\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{2,i})\\ &- \mathcal{A}_2^{\mathsf{T}}\text{diag}(\mathcal{A}_0^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{0,i}) - \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_0^{\mathsf{T}}\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i}). \qquad (4.47)\end{aligned}$$

We show in the next theorem that the weight-error size converges to the neighbourhood of zero, with the size of the neighbourhood determined by the step-size and the added noise variance.

Theorem 4.1 (MSE convergence of privatized decentralized learning). *Under assumptions 4.1 and 4.2, the decentralized recursions* (4.24)−(4.26) *converge exponentially fast for a small enough step-size to a neighbourhood of the optimal model, i.e.:*

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 \leq O(\mu)\sigma_s^2 + O(\mu) + (O(\mu^{-1}) + O(\mu))\sigma_g^2. \qquad (4.48)$$

Proof. See Appendix 4.B. □

By examining the bound in (4.48) on the MSE, we observe that the noise variance σ_g^2 appears multiplied by a term on the order of μ^{-1}, which is detrimental to performance when μ is small. Therefore, the traditional approach of adding Laplacian noise over the edges to ensre privacy is *calamitous* to performance and needs to be improved. We describe next an alternative approach.

4.3.3 Graph-Homomorphic Noise

The noises added to the communication links in the previous section did not take into account the graph topology. As a result, their effect gets magnified by $O(\mu^{-1})$ as shown in (4.48). We now examine another strategy for adding noise, which relies on a graph-homomorphic construction from [86]. Specifically, the noises are now constructed to satisfy the following condition:

$$\sum_{p,m=1}^{P} q_p a_{mp} \boldsymbol{g}_{j,mp,i} = 0, \tag{4.49}$$

for $j = 0, 1, 2$, and where $q = \text{col}\Big\{q_p\Big\}_{p=1}^{P}$ is the Perron eigenvector of $A_2^\mathsf{T} A_0^\mathsf{T} A_1^\mathsf{T}$. This can be satisfied if we continue to choose zero-mean Laplacian noises $\boldsymbol{g}_{j,p,i}$ with variance σ_g^2 and then set:

$$\boldsymbol{g}_{j,pm,i} = \begin{cases} \frac{a_{pm}}{a_{mp}} \boldsymbol{g}_{j,p,i}, & m \neq p \\ -\frac{1-a_{j,pp}}{a_{j,pp}} \boldsymbol{g}_{j,p,i}, & m = p. \end{cases} \tag{4.50}$$

Condition (4.49), along with construction (4.50), ensure that the net effect of the additional noises cancel out over the entire graph during the local aggregation steps. We show in the next theorem that the MSE bound improves in this case. To see this, we first introduce the network centroid $\boldsymbol{w}_{c,i}$ and study its convergence under graph-homomorphic perturbations. Let:

$$\begin{aligned} \boldsymbol{w}_{c,i} &\overset{\Delta}{=} \sum_{p=1}^{P} q_p \boldsymbol{w}_{p,i} \\ &= \boldsymbol{w}_{c,i-1} - \mu \sum_{p=1}^{P} q_p \boldsymbol{s}_{p,i} - \mu \sum_{p=1}^{P} q_p \nabla_{w^\mathsf{T}} J_p \left(\sum_{m \in \mathcal{N}_p} a_{1,mp} (\boldsymbol{w}_{m,i-1} + \boldsymbol{g}_{1,mp,i}) \right) \\ &\quad + \sum_{p,m} q_p \left(a_{1,mp} \boldsymbol{g}_{1,mp,i} + a_{0,mp} \boldsymbol{g}_{0,mp,i} + a_{2,mp} \boldsymbol{g}_{2,mp,i} \right). \end{aligned} \tag{4.51}$$

Since we are using graph-homomorphic perturbations, the sum of the noise terms in the last line cancels out. We can therefore write the following error recursion:

$$\widetilde{\boldsymbol{w}}_{c,i} = \widetilde{\boldsymbol{w}}_{c,i-1} + \mu (q^\mathsf{T} \otimes I) \boldsymbol{s}_i + \mu (q^\mathsf{T} \otimes I) b - \mu \sum_{p=1}^{P} q_p \boldsymbol{H}_{p,i-1} \sum_{m \in \mathcal{N}_p} a_{1,mp} (\widetilde{\boldsymbol{w}}_{m,i-1} - \boldsymbol{g}_{1,mp,i}). \tag{4.52}$$

Before stating the theorem on the MSE convergence, we bound the network disagreement defined as the average second-order moment of the difference between the local models and the centroid model.

Lemma 4.1 (**Network disagreement**). *The average deviation from the centroid is uniformly bounded during each iteration i, and, moreover, it holds asymptotically that:*

$$\limsup_{i\to\infty} \frac{1}{P}\sum_{p=1}^{P} \mathbb{E}\|\boldsymbol{w}_{p,i} - \boldsymbol{w}_{c,i}\|^2 \leq O(1)\sigma_g^2 + O(\mu) \tag{4.53}$$

Proof. See Appendix 4.C. □

Expression (4.53) shows that the local models will be at most $O(1)\sigma_g^2$ away from the centroid model. Thus, if the centroid model manages to converge to the optimal model w^o with only a slight variation, then the local models will always be a constant, proportional to the noise variance σ_g^2, away from the true model. In the next theorem, we show that the added noise only alters the centroid model by an $O(1)$ factor.

Theorem 4.2 (**MSE convergence of the network centroid**). *Under assumptions 4.1 and 4.2, the network centroid defined in* (4.51) *converges exponentially fast for a small enough step-size to a neighbourhood of the optimal model:*

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq O(\mu)\sigma_s^2 + O(1)\sigma_g^2 + O(\mu^2). \tag{4.54}$$

Proof. See Appendix 4.D. □

Thus, the network centroid is at most $O(1)\sigma_g^2$ away from the true minimizer w^o, even with added noise, as opposed to $O(\mu^{-1})\sigma_g^2$. In Lemma 4.1, we showed that the individual models $\boldsymbol{w}_{p,i}$ are $O(1)\sigma_g^2$ away from the centroid model. Thus, by using the graph-homomorphic perturbations (4.49)–(4.50), the MSE is not inversely proportional to μ anymore, which is an improvement relative to (4.48).

4.3.4 Local Graph-Homomorphic Noise

We explain how to improve on the $O(1)\sigma_g^2$ deviation and replace it by $O(\mu)\sigma_g^2$, by relying on the use of *local* graph-homomorphic noise. To do so, we construct the noises to satisfy the following alternative condition as opposed to (4.49):

$$\sum_{m\in\mathcal{N}_p} a_{j,mp}\boldsymbol{g}_{j,mp,i} = 0. \tag{4.55}$$

Observe that we are requiring the sum of the noises to cancel out *locally*, rather than globally as required in the previous section. The neighbours of every agent p must collaborate together to generate dependent random noises that will cancel out locally at p. The collaboration will occur through agent p, since a direct link might not exist amongst these neighbours. A similar problem exists in blockchain applications where the generation of a random number is required to occur in a decentralized manner [137].

We now devise a decentralized scheme that leads to noises that satisfy condition (4.55). For the sake of demonstration, we describe the protocol through an example. Thus, assume agent 1 is connected to 5 agents labelled $2, 3, 4, 5, 6$ (Figure 4.2, *left*). Since the neighbours of agent 1 need not be connected to each other through direct links, all communications will take place through agent 1. In this subnetwork, we allow agent 1 to be the orchestrator of the scheme. The first step is for agent 1 to split its neighbours into two disjoint sets $\mathcal{N}_1 = \mathcal{N}_+ \bigcup \mathcal{N}_-$. For example, we may collect the even numbered agents into $\mathcal{N}_+$, and the odd numbered agents into $\mathcal{N}_-$. Then, we allow every pair of agents from the two disjoint sets to agree on a noise value they will add to their message such that they will cancel out at agent 1. We force agents from $\mathcal{N}_-$ to multiply the noise they will add to their messages by a negative sign. Therefore, agent 2 will add to its message two noise terms, one generated with agent 3 and another with agent 5. We denote the noise term generated by agents 2 and 3 that will be sent to agent 1 by $\boldsymbol{g}_{\{23\}1,i}$. Since the messages are scaled by the weights attributed by agent 1 to its neighbours, the added noise must then be divided by the weights, i.e., the message sent by agent 2 to agent 1 is the original message $\boldsymbol{w}_{2,i}$ and the two generated noises by agent 2 with agents 3 and 5 scaled by the corresponding weight a_{12}:

$$\boldsymbol{w}_{2,i} + \frac{\boldsymbol{g}_{\{23\}1,i}}{a_{12}} + \frac{\boldsymbol{g}_{\{25\}1,i}}{a_{12}}. \tag{4.56}$$

However, this requires that agent 2 know the weight attributed to its messages by agent 1. Thus, agent 1 will have to make the weights public in case of a non-doubly stochastic combination matrix. The same process occurs between agents 4 and 6 with both 3 and 5. Then, the aggregate messages sent to agent 1 will end up being the sum of the unmasked weights:

$$\sum_{k\in\mathcal{N}_+} a_{1k}\left(\boldsymbol{w}_{k,i} + \sum_{\ell\in\mathcal{N}_-}\frac{\boldsymbol{g}_{\{k\ell\}1,i}}{a_{1k}}\right) + \sum_{k\in\mathcal{N}_-} a_{1k}\left(\boldsymbol{w}_{k,i} - \sum_{\ell\in\mathcal{N}_+}\frac{\boldsymbol{g}_{\{\ell k\}1,i}}{a_{1k}}\right) = \sum_{k\in\mathcal{N}_1} a_{1k}\boldsymbol{w}_{k,i}. \tag{4.57}$$

We move to the method used to generate the pairwise noise terms $\boldsymbol{g}_{\{k\ell\}1,i}$. We rely on the

Diffie-Helman key exchange protocol where each pair of agents shares a secret key that is used to generate the added noise. Given two agents, say 2 and 3, we assume they have individual secret keys v_2 and v_3, respectively. A known modulus π and base b is agreed upon amongst the agents. Then, agent 2 broadcasts its public key $V_2 = (b^{v_2} \mod \pi)$ and agent 3 does the same $V_3 = (b^{v_3} \mod \pi)$. Agent 2 then calculates, $v_{23} = (V_3^{v_2} \mod \pi) = (b^{v_2 v_3} \mod \pi)$ which is the same as what agent 3 calculates $v_{23} = (V_2^{v_3} \mod p) = (b^{v_3 v_2} \mod \pi)$. Thus, the two agents now share a secret key v_{23} only known to them. This secret key can then be used as the added noise to mask the messages, i.e., $\boldsymbol{g}_{\{23\}1,i} = v_{23}$. However, to make the process differentially private we need the resulting added noise to be Laplacian, $\text{Lap}(0, \sigma_g/\sqrt{2})$. In what follows, we describe a scheme, which we call the local graph-homomorphic processing scheme that ensures the added noise is Laplacian. An illustration of this process is found in the right subfigure of Figure 4.2.

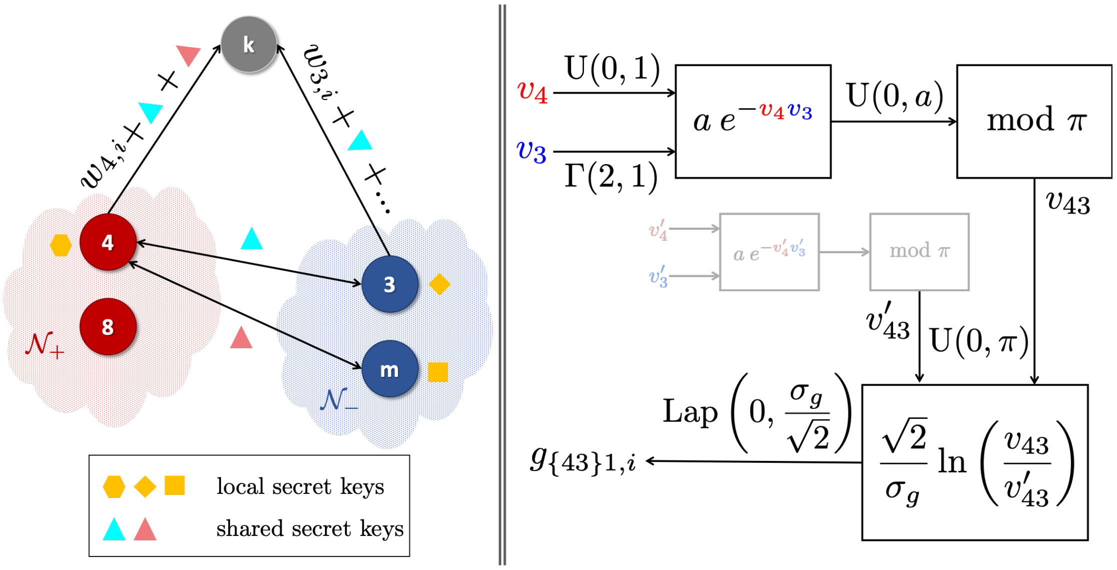


Figure 4.2 – Illustration of the local graph-homomorphic process. The figure on the left describes the Diffie-Helman key exchange procedure. The figure on the right shows the transformation the random variable goes through.

Definition 4.2 (**Local graph-homomorphic process**). *We are given a subnetwork of agent k, and neighbours $\ell \in \mathcal{N}_+$ and $m \in \mathcal{N}_-$. Let agent ℓ sample two secrect keys $\boldsymbol{v}_\ell$ and $\boldsymbol{v}'_\ell$ from a uniform distribution on $[0,1]$, and let agent m sample its keys $\boldsymbol{v}_m$ and $\boldsymbol{v}'_m$ from a gamma distribution $\Gamma(2,1)$. Let π be some large prime number and let a be a multiple of π. Then, for:*

$$\boldsymbol{v}_{\ell m} = a\, e^{-\boldsymbol{v}_\ell \boldsymbol{v}_m} \mod \pi, \tag{4.58}$$

$$\boldsymbol{v}'_{\ell m} = a\, e^{-\boldsymbol{v}'_\ell \boldsymbol{v}'_m} \mod \pi, \tag{4.59}$$

the desired Laplacian noise can be constructed as:

$$\boldsymbol{g}_{\{\ell m\}k,i} = \frac{\sqrt{2}}{\sigma_g} \ln \left(\frac{\boldsymbol{v}_{\ell m}}{\boldsymbol{v}'_{\ell m}} \right). \tag{4.60}$$

The local graph-homomorphic process proposed here is related to methods that fall under secure aggregation (see [99]). However, the main difference between our method and earlier investigations is that we devise a scheme for the more general *decentralized* setting, while other works focus largely on the particular case of federated learning with its specialized structure with a central processor. Furthermore, while we generate random numbers making our scheme more secure, the work [99] adds pseudo-random numbers to the shared messages. Since pseudo-random numbers are generated by deterministic algorithms, it makes the noise predictable and susceptible to attacks, contrary to random numbers. Furthermore, we quantify the privacy of our scheme as opposed to [99]. In the next theorem, we show that using construction (4.60) results in a differentially private algorithm.

Theorem 4.3 (**Privacy with local graph-homomorphic perturbations**). *Under the local graph-homomorphic process defined by* (4.60) *the resulting privatized algorithm is* $\epsilon(i)-$*differentially private with high probability and with* $\epsilon(i)$ *defined in* (4.34).

Proof. See Appendix 4.E. □

Note that since the noises cancel out locally during each iteration, the algorithm follows the same trajectory as the non-privatized algorithm. This implies that the MSD performance of the privatized decentralized learning algorithm with the local graph-homomorphic perturbation (4.60) is the same as the non-privatized decentralized learning algorithm. In particular, the results on the convergence of the non-privatized algorithm from Theorem 9.1 in [9] will continue to hold. This implies that the MSE will now be on the order of $O(\mu)\sigma_g^2$.

The above result highlights one difficulty with differential privacy. Note that, in principle, as the variance σ_g^2 of the added noise is increased, the level of privacy is also increased. However, this process introduces an additional communication cost. For example, agent 1 needs to communicate to its neighbourhood the splitting into the positive agents and negative agents. It will also need to communicate with almost half the neighbourhood of its neighbours to agree on an added noise $\boldsymbol{g}_{\{1m\}k,i}$ for $k \in \mathcal{N}_1$ and $m \in \mathcal{N}_k$. Thus, in total, the communication cost for agent 1 will increase by at most $|\mathcal{N}_1| + \sum_{k=2}^{6} |\mathcal{N}_k|/2$. The additional communication cost is not captured by the privacy measure even though it clearly affects the level of privacy. If we were to decrease the number of times a random noise is generated by the local graph-homomorphic process and instead re-use the noise, then we would be decreasing the communication cost but increasing the chance of an attacker learning the noise and unmasking the messages. This reduces its functionality and motivates the search for other privacy metrics, such as those based on information-theoretic measures [55]. For example, one metric could be the mutual information between the original message and the perturbed shared message [138]. Thus, if we assume the individual messages $\{\boldsymbol{w}_{k,i}\}$ are Guassian random variables with variance $\sigma_{\boldsymbol{w}}^2$, and if we perturb them with a total Guassian noise $\boldsymbol{g}_{k,i}$ of variance σ_g^2, then the mutual information is given by:

$$\begin{aligned} I(\boldsymbol{w}_{k,i}; \boldsymbol{w}_{k,i} + \boldsymbol{g}_i) &= H(\boldsymbol{w}_{k,i} + \boldsymbol{g}_{k,i}) - H(\boldsymbol{w}_{k,i} + \boldsymbol{g}_{k,i} | \boldsymbol{w}_{k,i}) \\ &= H(\boldsymbol{w}_{k,i} + \boldsymbol{g}_{k,i}) - H(\boldsymbol{g}_{k,i}) \\ &= \frac{1}{2} \log \left(1 + \frac{\sigma_{\boldsymbol{w}}^2}{\sigma_g^2} \right). \end{aligned} \tag{4.61}$$

Obeserve again that as we increase the noise variance σ_g^2, mutual information decreases while privacy increases. Mutual information again fails to capture the communication cost incurred by the process. It appears that no metric capturing the communication-privacy trade-off exists as of yet in the literature. This calls for the search of a more appropriate privacy metric for secure aggregation methods.

4.4 Experimental Results

We run two experiments. In the first experiment we focus on a linear regression problem with simulated data. We then study a classification problem on real data.

4.4.1 Generalized Decentralized Privacy Learning

For each of consensus, CTA, and ATC diffusion, we compare four algorithms: the standard decentralized algorithm, the privatized algorithm with random perturbations, the privatized algorithm with graph-homomorphic perturbations, and the privatized algorithm with local graph-homomorphic perturbations. We consider a network of 30 agents (Figure 4.3) and a regularized quadratic loss function:

$$\min_{w \in \mathbb{R}^2} \frac{1}{30} \sum_{p=1}^{30} \frac{1}{100} \sum_{n=1}^{100} (\boldsymbol{d}_p(n) - \boldsymbol{u}_{p,n}^{\mathsf{T}} w)^2 + 0.01\|w\|^2. \tag{4.62}$$

We generate a random dataset $\{\boldsymbol{u}_{p,n}, \boldsymbol{d}_p(n)\}_{n=1}^{100}$ as follows: we let the two-dimensional

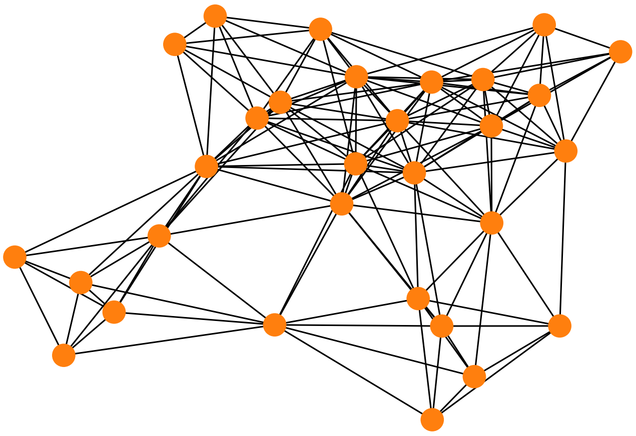


Figure 4.3 – The generated network of agents.

feature vector $\boldsymbol{u}_{p,n} \sim \mathcal{N}(0; R_u)$, and add noise $\boldsymbol{o}_p(n) \sim \mathcal{N}(0; \sigma^2_{o,p})$ such that $\boldsymbol{d}_p(n) = \boldsymbol{u}^{\mathsf{T}}_{p,n} w^\star + \boldsymbol{o}_p(n)$, for some generative model $w^\star \in \mathbb{R}^2$ and randomly set variance R_u and added noise variance $\sigma^2_{o,p}$. To make the data distributions non-iid, we use different noise variances $\sigma^2_{o,p}$ across the agents. The optimal global model has a closed form solution with $\widehat{R}_u$ and $\widehat{r}_{uo}$ as defined previously:

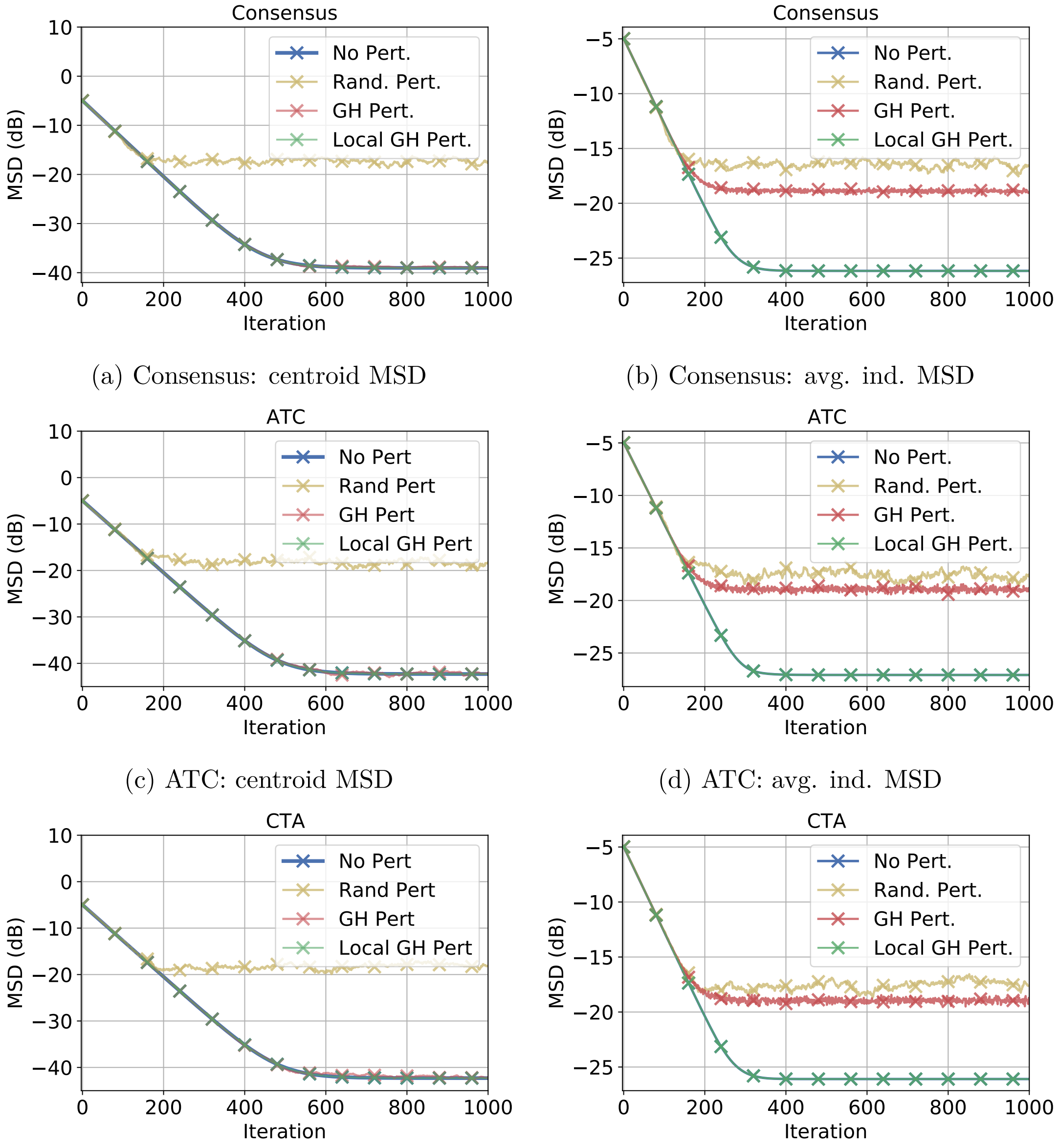
$$w^o = (\widehat{R}_u + 0.01I)^{-1}(\widehat{R}_u w^\star + \widehat{r}_{uo}). \tag{4.63}$$

We set the step-size $\mu = 0.4$, the noise variance $\sigma^2_g = 0.01$, and run the algorithms for 1000 iterations. We repeat the experiment 20 times and plot the MSD in the log domain of the centroid model and the average of the individual MSDs:

$$\text{MSD}_i \triangleq \|\boldsymbol{w}_{c,i} - w^o\|^2, \tag{4.64}$$

$$\text{MSD}_{\text{avg},i} \triangleq \frac{1}{P}\sum_{p=1}^{P} \|\boldsymbol{w}_{p,i} - w^o\|^2. \tag{4.65}$$

As we observe in Figure 4.4, graph-homomorphic perturbations do not hinder the performance of the algorithm in approximating the true model as do random perturbations. The random perturbations MSD curve (yellow) is significantly higher than the graph homomorphic perturbations MSD curve (red) which is close to the non-perturbed MSD curve (blue). If we examine the average MSD of the individual models, we observe that the decay in performance is not as much as that for random perturbations. Moreover, since local graph-homomorphic perturbations do not affect the performance of the algorithms, we observe that the MSD curve follows that of the non-privatized algorithm.



(a) Consensus: centroid MSD

(b) Consensus: avg. ind. MSD

(c) ATC: centroid MSD

(d) ATC: avg. ind. MSD

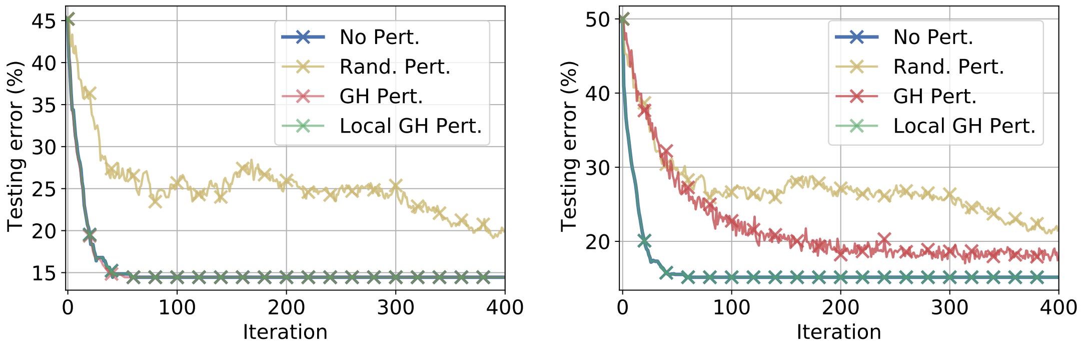
(e) CTA: centroid MSD

(f) CTA: avg. ind. MSD

Figure 4.4 – MSD plots for the three decentralized learning algorithms.

4.4.2 Classification in Decentralized Learning

We next run an experiment on a classification dataset. We use the Avazu click through dataset [79], which contains a set of online add clicks. We distribute the data among $P = 50$ agents. To get non-iid data, we add non-iid Gaussian noise to each agent's dataset. We let $\mu = 0.5$, $\rho = 0.001$, and $\sigma_g^2 = 0.8$. We plot the testing error in Figure 4.5 of the standard algorithm, the privatized algorithm with random perturbations, the privatized algorithm with graph-homomorphic perturbations, and the privatized algorithm with local graph-homomorphic perturbations. It comes as no surprise that using graph-homomorphic perturbations does not hinder the testing error as random perturbations do, and local graph-homomorphic perturbations do not change the testing error.



(a) Centroid testing error

(b) Average individual testing error

Figure 4.5 – Testing error of standard ATC, privatized ATC with random perturbations, and privatized ATC with graph-homomorphic perturbations.

4.5 Conclusion

The goal of this chapter has been to study the effect of privacy in decentralized learning. We established the superiority of graph-homomorphic perturbations in the model performance, as opposed to random perturbations. We then designed local graph-homomorphic perturbations that ensure the added noise does not affect the model performance. Thus, the main takeaway from this work is that graph-homomorphic perturbations are better than random perturbations in decentralized learning.

4.A Sensitivity of the Decentralized Algorithm

We study the sensitivity of the decentralized learning algorithm (4.21)–(4.23), which is defined at each time instant by the expression:

$$\Delta(i) = \|\mathcal{W}_i - \mathcal{W}_i'\|. \tag{4.66}$$

This definition captures the change when the data samples of a single agent are changed. The prime symbol represents the new trajectory. We can bound the sensitivity using the triangle inequality by the individual errors and the difference in the optimal models:

$$\Delta(i) \leq \|\widetilde{\mathcal{W}}_i'\| + \|\widetilde{\mathcal{W}}_i\| + \sqrt{P}\|w^o - w'^o\|. \tag{4.67}$$

Then, for any constants B and B' chosen by the desiger, we can use Markov's inequality to get the bounds:

$$\mathbb{P}(\|\widetilde{\boldsymbol{w}}_i\| \geq B) \leq \frac{\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2}{B^2}, \tag{4.68}$$

$$\mathbb{P}(\|\widetilde{\boldsymbol{w}}'_i\| \geq B') \leq \frac{\mathbb{E}\|\widetilde{\boldsymbol{w}}'_i\|^2}{B'^2}. \tag{4.69}$$

Now we recal from Theorem 4.1 that:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2 \leq \kappa_2^2 \mathbb{1}^\mathsf{T}\Gamma^i \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{w}}_0\|^2 \\ \|\mathbb{E}\check{\boldsymbol{w}}_0\|^2 \end{bmatrix} + O(\mu) + O(\mu^{-1}), \tag{4.70}$$

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}'_i\|^2 \leq \kappa_2'^2 \mathbb{1}^\mathsf{T}\Gamma'^i \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{w}}'_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{w}}'_0\|^2 \end{bmatrix} + O(\mu) + O(\mu^{-1}). \tag{4.71}$$

It follows that the sensitivity is bounded by:

$$\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\| \tag{4.72}$$

with high probability given by:

$$\begin{aligned} \mathbb{P}\left(\Delta(i) \leq B + B' + \sqrt{P}\|w^o - w'^o\|\right) \geq & \left(1 - \frac{C + O(\mu) + O(\mu^{-1})}{B^2}\right) \\ & \times \left(1 - \frac{C' + O(\mu) + O(\mu^{-1})}{B'^2}\right). \end{aligned} \tag{4.73}$$

where C and C' are given by:

$$C = \kappa_2^2 \mathbb{1}^\mathsf{T}\Gamma^i \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{w}}_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{w}}_0\|^2 \end{bmatrix}, \tag{4.74}$$

$$C' = \kappa_2'^2 \mathbb{1}^\mathsf{T}\Gamma'^i \begin{bmatrix} \mathbb{E}\|\bar{\boldsymbol{w}}'_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{w}}'_0\|^2 \end{bmatrix}. \tag{4.75}$$

4.B Proof of Theorem 4.1

The following proof follows similar steps to those used in [9] for non-private algorithms. Using the Jordan decomposition of $A_2^\mathsf{T} A_0^\mathsf{T} A_1^\mathsf{T}$:

$$\begin{aligned} A_2^\mathsf{T} A_0^\mathsf{T} A_1^\mathsf{T} &= V_\theta J V_\theta^{-1}, && (4.76) \\ V_\theta &\triangleq \begin{bmatrix} q & V_R \end{bmatrix}, && (4.77) \\ V_\theta^{-1} &\triangleq \begin{bmatrix} \mathbb{1}^\mathsf{T} \\ V_L^\mathsf{T} \end{bmatrix}, && (4.78) \\ J &\triangleq \begin{bmatrix} 1 & 0 \\ 0 & J_\theta \end{bmatrix}, && (4.79) \end{aligned}$$

where q is the Perron eigenvector of $A_2^\mathsf{T} A_0^\mathsf{T} A_1^\mathsf{T}$ and J_θ contains Jordan blocks of the corresponding eigenvalues λ of the form (example of a 3×3 matrix):

$$\begin{bmatrix} \lambda & 0 & 0 \\ \theta & \lambda & 0 \\ 0 & \theta & \lambda \end{bmatrix}, \tag{4.80}$$

with a constant θ in the subdiagonal. We first write:

$$\boldsymbol{\mathcal{B}}_{i-1} = (\mathcal{V}_\theta^{-1})^\mathsf{T} (\mathcal{J} - \boldsymbol{\mathcal{D}}_{i-1}^\mathsf{T}) \mathcal{V}_\theta^\mathsf{T}, \tag{4.81}$$

where:

$$\begin{aligned} \mathcal{V}_\theta^{-1} &\triangleq V_\theta^{-1} \otimes I_M, && (4.82) \\ \mathcal{V}_\theta &\triangleq V_\theta \otimes I_M, && (4.83) \\ \mathcal{J} &\triangleq J \otimes I_M = \begin{bmatrix} I_M & 0 \\ 0 & \mathcal{J}_\theta \end{bmatrix}, && (4.84) \\ \boldsymbol{\mathcal{D}}_{i-1}^\mathsf{T} &\triangleq \mu \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} \boldsymbol{\mathcal{H}}_{i-1} \mathcal{A}_1^\mathsf{T} (\mathcal{V}_\theta^{-1})^\mathsf{T} = \begin{bmatrix} \boldsymbol{D}_{11,i-1}^\mathsf{T} & \boldsymbol{D}_{21,i-1}^\mathsf{T} \\ \boldsymbol{D}_{12,i-1}^\mathsf{T} & \boldsymbol{D}_{22,i-1}^\mathsf{T} \end{bmatrix}. && (4.85) \end{aligned}$$

It is shown in the proof of Theorem 9.1 in [9] that:

$$\begin{aligned} \|I_M - \boldsymbol{D}_{11,i-1}\| &\le 1 - \sigma_{11}\mu, && (4.86) \\ \|\boldsymbol{D}_{ij}\| &\le \sigma_{ij}\mu, && (4.87) \end{aligned}$$

for some positive constants σ_{ij} for $i, j = 1, 2$. Multiplying both sides of the error recursion (4.47) from the left by $\mathcal{V}_\theta^\mathsf{T}$:

$$\begin{aligned} \mathcal{V}_\theta^\mathsf{T} \widetilde{\boldsymbol{\mathcal{W}}}_i = {} & \mathcal{V}_\theta^\mathsf{T} \boldsymbol{\mathcal{B}}_{i-1} (\mathcal{V}_\theta^{-1})^\mathsf{T} \mathcal{V}_\theta^\mathsf{T} \widetilde{\boldsymbol{\mathcal{W}}}_{i-1} + \mu \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} \boldsymbol{s}_i - \mu \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} b + \mu \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} \boldsymbol{\mathcal{H}}_{i-1} \text{diag}(\mathcal{A}_1^\mathsf{T} \boldsymbol{\mathcal{G}}_{1,i}) \\ & - \mathcal{V}_\theta^\mathsf{T} \text{diag}(\mathcal{A}_2^\mathsf{T} \boldsymbol{\mathcal{G}}_{2,i}) - \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} \text{diag}(\mathcal{A}_0^\mathsf{T} \boldsymbol{\mathcal{G}}_{0,i}) - \mathcal{V}_\theta^\mathsf{T} \mathcal{A}_2^\mathsf{T} \mathcal{A}_0^\mathsf{T} \text{diag}(\mathcal{A}_1^\mathsf{T} \boldsymbol{\mathcal{G}}_{1,i}), \end{aligned} \tag{4.88}$$

and introducing the new notation:

$$\mathcal{V}_\theta^\mathsf{T}\widetilde{\boldsymbol{\mathcal{w}}}_i = \begin{bmatrix}(q^\mathsf{T}\otimes I_M)\widetilde{\boldsymbol{\mathcal{w}}}_i \\ (V_R^\mathsf{T}\otimes I)\widetilde{\boldsymbol{\mathcal{w}}}_i\end{bmatrix} \triangleq \begin{bmatrix}\bar{\boldsymbol{\mathcal{w}}}_i \\ \check{\boldsymbol{\mathcal{w}}}_i\end{bmatrix}, \tag{4.89}$$

$$\mu\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\boldsymbol{s}_i = \mu\begin{bmatrix}(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\boldsymbol{s}_i \\ (V_R^\mathsf{T}\otimes I)\mathcal{A}_2^\mathsf{T}\boldsymbol{s}_i\end{bmatrix} \triangleq \begin{bmatrix}\bar{\boldsymbol{s}}_i \\ \check{\boldsymbol{s}}_i\end{bmatrix}, \tag{4.90}$$

$$\mu\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}b = \mu\begin{bmatrix}(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}b \\ (V_R^\mathsf{T}\otimes I)\mathcal{A}_2^\mathsf{T}b\end{bmatrix} \triangleq \begin{bmatrix}0 \\ \check{b}\end{bmatrix}, \tag{4.91}$$

we get:

$$\begin{aligned}\begin{bmatrix}\bar{\boldsymbol{\mathcal{w}}}_i \\ \check{\boldsymbol{\mathcal{w}}}_i\end{bmatrix} = &\begin{bmatrix}I_M - \boldsymbol{D}_{11,i-1}^\mathsf{T} & -\boldsymbol{D}_{21,i-1}^\mathsf{T} \\ -\boldsymbol{D}_{12,i-1}^\mathsf{T} & \mathcal{J}_\epsilon\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{\mathcal{w}}}_{i-1} \\ \check{\boldsymbol{\mathcal{w}}}_{i-1}\end{bmatrix} + \begin{bmatrix}\bar{\boldsymbol{s}}_i \\ \check{\boldsymbol{s}}_i\end{bmatrix} + \begin{bmatrix}0 \\ \check{b}\end{bmatrix} + \mu\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i}) \\ &- \mathcal{V}_\theta^\mathsf{T}\mathrm{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i}) - \mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i}) - \mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i}).\end{aligned} \tag{4.92}$$

Then, taking the expectation of the ℓ_2-norm, and using Jensen's inequality, we have:

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{\mathcal{w}}}_i\|^2 \leq\; &(1-\sigma_{11}\mu)\mathbb{E}\|\bar{\boldsymbol{\mathcal{w}}}_{i-1}\|^2 + \frac{\sigma_{21}^2\mu}{\sigma_{11}}\mathbb{E}\|\check{\boldsymbol{\mathcal{w}}}_{i-1}\|^2 + \mathbb{E}\|\bar{\boldsymbol{s}}_i\|^2 \\ &+ 2\mu^2\mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + 2\mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 \\ &+ \mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathrm{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i})\|^2 + \mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i})\|^2,\end{aligned} \tag{4.93}$$

and:

$$\begin{aligned}\mathbb{E}\|\check{\boldsymbol{\mathcal{w}}}_i\|^2 \leq\; &\left(\rho(J_\theta) + \theta + \frac{2\sigma_{22}^2\mu^2}{1-\rho(J_\theta)-\theta}\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{w}}}_{i-1}\|^2 + \frac{3\sigma_{21}^2\mu^2}{1-\rho(J_\theta)-\theta}\mathbb{E}\|\bar{\boldsymbol{\mathcal{w}}}_{i-1}\|^2 \\ &+ \frac{3\|\check{b}\|^2}{1-\rho(J_\theta)-\theta} + \mathbb{E}\|\check{\boldsymbol{s}}_i\|^2 + 2\mu^2\mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 \\ &+ \mathbb{E}\|J_\epsilon^\mathsf{T}(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + \mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathrm{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i})\|^2 \\ &+ \mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i})\|^2,\end{aligned} \tag{4.94}$$

with the cross terms equal to zero due to the independence of the zero-mean random variables. Then, we bound the sum of the gradient noise:

$$\begin{aligned}\mathbb{E}\|\bar{\boldsymbol{s}}_i\|^2 + \mathbb{E}\|\check{\boldsymbol{s}}_i\|^2 &\leq \|\mathcal{V}_\theta\|^2\mu^2\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 \\ &\leq \kappa_1^2\mu^2\sum_{p=1}^{P}\beta_{s,p}^2\mathbb{E}\|\widetilde{\boldsymbol{\phi}}_{p,i-1}\|^2 + \sigma_{s,p}^2 \\ &\leq \kappa_1^2\mu^2\sum_{p=1}^{P}\beta_{s,p}^2\sum_{m=1}^{P}\left(\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 + \mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2\right) + \sigma_{s,p}^2\end{aligned}$$

$$
\begin{aligned}
&\leq \kappa_1^2\mu^2\beta_s^2\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \kappa_1^2\mu^2\sigma_s^2 + \kappa_1^2\mu^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2 \\
&\leq \kappa_1^2\kappa_2^2\mu^2\beta_s^2\left(\mathbb{E}\|\bar{\boldsymbol{s}}_i\|^2 + \mathbb{E}\|\check{\boldsymbol{s}}_i\|^2\right) + \kappa_1^2\mu^2\sigma_s^2 + \kappa_1^2\mu^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2, \qquad (4.95)
\end{aligned}
$$

where we introduced the constants β_s^2 and σ_s^2, which are the sums of $\beta_{s,p}^2$ and $\sigma_{s,p}^2$, respectively. Then, going back:

$$
\begin{aligned}
\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_i\|^2 \leq{}& (1-\sigma_{11}\mu + \kappa_1^2\kappa_2^2\beta_s^2\mu^2)\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \left(\frac{\sigma_{21}^2\mu}{\sigma_{11}} + \kappa_1^2\kappa_2^2\mu^2\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \kappa_1^2\mu^2\sigma_s^2 \\
&+ \kappa_1^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mu^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2 + 2\mu^2\mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 \\
&+ 2\mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + \mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathrm{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i})\|^2 \\
&+ \mathbb{E}\|(q^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i})\|^2, \qquad (4.96)
\end{aligned}
$$

and:

$$
\begin{aligned}
&\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_i\|^2 \\
&\leq \left(\rho(J_\theta) + \theta + \frac{3\sigma_{22}^2\mu^2}{1-\rho(J_\theta)-\theta} + \kappa_1^2\kappa_2^2\beta_s^2\mu^2\right)\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 \\
&\quad + \left(\frac{3\sigma_{12}^2\mu^2}{1-\rho(J_\theta)-\theta} + \kappa_1^2\kappa_2^2\beta_s^2\mu^2\right)\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 \\
&\quad + \frac{3\|\check{b}\|^2}{1-\rho(J_\theta)-\theta} + \kappa_1^2\mu^2\sigma_s^2 + \kappa_1^2\mu^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2 \\
&\quad + 2\mu^2\mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + \mathbb{E}\|J_\theta^\mathsf{T}(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 \\
&\quad + \mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathrm{diag}(\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i})\|^2 + \mathbb{E}\|(V_R^\mathsf{T}\otimes I_M)\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i})\|^2. \qquad (4.97)
\end{aligned}
$$

Adding the two bounds:

$$
\begin{aligned}
&\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 \\
&\leq \kappa_2^2\Bigg(\bar{\gamma}\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \check{\gamma}\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \frac{3\|\check{b}\|^2}{1-\rho(J_\theta)-\theta} + 2\kappa_1^2\mu^2\sigma_s^2 + 2\kappa_1^2\mu^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2 \\
&\quad + 2\mu^2\mathbb{E}\|\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{H}}_{i-1}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + \mathbb{E}\|\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\mathcal{A}_0^\mathsf{T}\mathrm{diag}(\mathcal{A}_1^\mathsf{T}\boldsymbol{\mathcal{G}}_{1,i})\|^2 \\
&\quad + \mathbb{E}\|\mathcal{V}_\theta^\mathsf{T}\mathrm{diag}\ (\mathcal{A}_2^\mathsf{T}\boldsymbol{\mathcal{G}}_{2,i})\|^2 + \mathbb{E}\|\mathcal{V}_\theta^\mathsf{T}\mathcal{A}_2^\mathsf{T}\mathrm{diag}(\mathcal{A}_0^\mathsf{T}\boldsymbol{\mathcal{G}}_{0,i})\|^2\Bigg)
\end{aligned}
$$

$$\leq \kappa_2^2\bigg(\bar{\gamma}\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \check{\gamma}\mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 + \frac{3\|\check{b}\|^2}{1-\rho(J_\theta)-\theta} + 2\kappa_1^2\mu^2\sigma_s^2 + 2\kappa_1^2\mu^2\sum_{p,m=1}^{P}\beta_{s,p}^2\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2$$
$$+ \kappa_1^2(1+2\delta^2\mu^2)\mathbb{E}\|\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i})\|^2 + \kappa_1^2\mathbb{E}\|\text{diag}(\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{2,i})\|^2 + \kappa_1^2\mathbb{E}\|\text{diag}(\mathcal{A}_0^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{0,i})\|^2\bigg). \tag{4.98}$$

Then, recursively bounding the MSE and taking the limit as i tends to infinity, we get:

$$\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 \leq \kappa_2^2\mathbb{1}^{\mathsf{T}}(I-\Gamma)^{-1}\begin{bmatrix}\kappa_1^2\mu^2\sigma_s^2 + \left(4+(2\delta^2+P\beta_s^2)\mu^2\right)\sigma_g^2\\ \frac{3\|\check{b}\|^2}{1-\rho(J_\theta)-\theta} + \kappa_1^2\mu^2\sigma_s^2 + \kappa_1^2\left(3+(2\delta^2+P\beta_s^2)\mu^2\right)\sigma_g^2\end{bmatrix}$$
$$= O(\mu)\sigma_s^2 + O(\mu) + (O(\mu^{-1})+O(\mu))\sigma_g^2, \tag{4.99}$$

where:

$$\Gamma \triangleq \begin{bmatrix}\bar{\gamma} & \frac{\sigma_{21}^2\mu}{\sigma_{11}} + \kappa_1^2\kappa_2^2\mu^2\\ \frac{3\sigma_{12}^2\mu^2}{1-\rho(J_\theta)-\theta} + \kappa_1^2\kappa_2^2\beta_s^2\mu^2 & \check{\gamma}\end{bmatrix}. \tag{4.100}$$

4.C Proof of Lemma 4.1

We define $\boldsymbol{\mathcal{W}}_{c,i} \triangleq (q\mathbb{1}^{\mathsf{T}}\otimes I)\boldsymbol{\mathcal{W}}_i$ and write:

$$\begin{aligned}\boldsymbol{\mathcal{W}}_i - \boldsymbol{\mathcal{W}}_{c,i} &= \left(I - q\mathbb{1}^{\mathsf{T}}\otimes I\right)\boldsymbol{\mathcal{W}}_i\\ &= (V_L^{\mathsf{T}}\otimes I)(V_R\otimes I)\boldsymbol{\mathcal{W}}_i\\ &= (V_L^{\mathsf{T}}\otimes I)J_\theta(V_R\otimes I)\boldsymbol{\mathcal{W}}_{i-1} - \mu(V_L^{\mathsf{T}}\otimes I)(V_R\otimes I)\text{col}\Big\{\nabla_{w^{\mathsf{T}}}J_p(\boldsymbol{\phi}_{p,i-1})\Big\}\\ &\quad - \mu(V_L^{\mathsf{T}}\otimes I)(V_R\otimes I)\boldsymbol{s}_i + (V_L^{\mathsf{T}}\otimes I)(V_R\otimes I)\Big(\text{diag}(\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{2,i}) + \mathcal{A}_2^{\mathsf{T}}\text{diag}(\mathcal{A}_0^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{0,i})\\ &\quad + \mathcal{A}_2^{\mathsf{T}}\mathcal{A}_0^{\mathsf{T}}\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i})\Big).\end{aligned} \tag{4.101}$$

We bound $\mathbb{E}\|(V_R\otimes I)\boldsymbol{\mathcal{W}}_i\|^2$ by using Jensen's inequality with a constant $\rho(J_\theta) < 1$ and define $\kappa_1^2 = \|\mathcal{V}_\theta\|^2$ and $\kappa_2^2 = \|\mathcal{V}_\theta^{-1}\|^2$:

$$\begin{aligned}&\mathbb{E}\|(V_R\otimes I)\boldsymbol{\mathcal{W}}_i\|^2\\ &\leq \rho(J_\theta)\mathbb{E}\|(V_R\otimes I)\boldsymbol{\mathcal{W}}_{i-1}\|^2 + \frac{\kappa_1^2\kappa_2^2\mu^2}{1-\rho(J_\theta)}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{H}_{p,i-1}\widetilde{\boldsymbol{\phi}}_{p,i-1} + \nabla_{w^{\mathsf{T}}}J_p(w^o)\|^2\\ &\quad + \kappa_1^2\kappa_2^2\mu^2\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 + v_1^2v_2^2\Big(\mathbb{E}\|\text{diag}(\mathcal{A}_2^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{2,i})\|^2 + \mathbb{E}\|\mathcal{A}_2^{\mathsf{T}}\text{diag}(\mathcal{A}_0^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{0,i})\|^2\\ &\quad + \mathbb{E}\|\mathcal{A}_2^{\mathsf{T}}\mathcal{A}_0^{\mathsf{T}}\text{diag}(\mathcal{A}_1^{\mathsf{T}}\boldsymbol{\mathcal{G}}_{1,i})\|^2\Big)\end{aligned}$$

$$
\begin{aligned}
&\leq \rho(J_\theta)\mathbb{E}\|(V_R \otimes I)\boldsymbol{\mathcal{W}}_{i-1}\|^2 + \frac{2\kappa_1^2\kappa_2^2\mu^2\|b\|^2}{1-\rho(J_\theta)} + \kappa_1^2\kappa_2^2\mu^2\sum_{p=1}^{P}\beta_{s,p}^2\sum_{m\in\mathcal{N}_p} a_{1,mp}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 \\
&\quad + \kappa_1^2\kappa_2^2\mu^2\sigma_s^2 + \kappa_1^2\kappa_2^2\mu^2\sum_{p,m=1}^{P} a_{1,mp}^2\sigma_g^2 + \kappa_1^2\kappa_2^2\sum_{p,m=1}^{P}(a_{2,mp}^2 + a_{0,mp}^2 + a_{1,mp}^2)\sigma_g^2. \qquad (4.102)
\end{aligned}
$$

Then, the individual errors $\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2$ can be bounded as shown in Theorem 4.1:

$$
\begin{aligned}
\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 &\leq \mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_{i-1}\|^2 \\
&\leq \kappa_2^2\mathbb{1}^\mathsf{T}\left(\Gamma^{i-1}\begin{bmatrix}\mathbb{E}\|\bar{\boldsymbol{\mathcal{W}}}_0\|^2 \\ \mathbb{E}\|\check{\boldsymbol{\mathcal{W}}}_0\|^2\end{bmatrix} + (I-\Gamma)^{-1}(I-\Gamma^{i-1})\begin{bmatrix}O(\mu^2)+O(1) \\ O(\mu^2)+O(1)\end{bmatrix}\right),
\end{aligned}
\qquad (4.103)
$$

where the $O(\mu^2)$ and $O(1)$ terms are constants depending on the gradient noise variance, the bias term b, and the noise variance. Also, the matrix Γ captures the rate of the recursion and was previously defined in Appendix 4.B.

Then, we plug back this bound into the main inequality (4.102) and recursively bound over i. The network disagreement is then bounded as:

$$
\frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i} - \boldsymbol{w}_{c,i}\|^2 \leq \frac{\kappa_2^2}{P}\mathbb{E}\|(V_R \otimes I)\boldsymbol{\mathcal{W}}_i\|^2, \qquad (4.104)
$$

and in the limit:

$$
\begin{aligned}
\limsup_{i\to\infty}\frac{1}{P}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i} - \boldsymbol{w}_{c,i}\|^2 \leq & \frac{2\kappa_1^2\kappa_2^4\mu^2\|b\|^2}{P(1-\rho(J_\theta))^2} + \frac{\kappa_1^2\kappa_2^4}{P(1-\rho(J_\theta))}\sigma_g^2 + O(\mu) \\
& + \frac{\kappa_1^2\kappa_2^4}{P(1-\rho(J_\theta))}\left(O(\mu^2)\sigma_s^2 + O(\mu^2)\sigma_g^2\right). \qquad (4.105)
\end{aligned}
$$

4.D Proof of Theorem 4.2

Starting from (4.52) and taking the conditional mean of the squared Euclidean norm over the past models, we can split the norm into three independent terms: the model error, the gradient noise, and the added noise. Taking again expectations and using

Jensen's with $\alpha = \sqrt{1 - 2\nu\mu + \delta^2\mu^2}$, we have:

$$\begin{aligned}
\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 &\leq \alpha\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\mathbb{E}\left\|(q^\mathsf{T} \otimes I)\boldsymbol{s}_i\right\|^2 \\
&\quad + \frac{\mu^2}{1-\alpha}\mathbb{E}\left\|\sum_{p=1}^{P} q_p \boldsymbol{H}_{p,i-1} \sum_{m\in\mathcal{N}_p} a_{1,mp}(\boldsymbol{w}_{m,i-1} - \boldsymbol{w}_{c,i-1})\right\|^2 \\
&\quad + \mu^2\mathbb{E}\left\|\sum_{p=1}^{P} q_p \boldsymbol{H}_{p,i-1} \sum_{m\in\mathcal{N}_p} a_{1,mp}\boldsymbol{g}_{1,mp,i}\right\|^2 \\
&\leq \alpha\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\mathbb{E}\left\|(q^\mathsf{T} \otimes I)\boldsymbol{s}_i\right\|^2 + \frac{\delta^2\mu^2}{1-\alpha}\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2 \\
&\quad + \mu^2\mathbb{E}\left\|\sum_{p=1}^{P} q_p \boldsymbol{H}_{p,i-1} \sum_{m\in\mathcal{N}_p} a_{1,mp}\boldsymbol{g}_{1,mp,i}\right\|^2 . \qquad (4.106)
\end{aligned}$$

We bound the gradient noise by starting from (4.38) and using Jensen's inequality to introduce $\widetilde{\boldsymbol{w}}_{c,i-1}$:

$$\begin{aligned}
\mathbb{E}\left\|(q^\mathsf{T} \otimes I)\boldsymbol{s}_i^2\right\|^2 &= \sum_{p=1}^{P} q_p^2\mathbb{E}\|\boldsymbol{s}_{p,i}\|^2 \\
&\leq \sum_{p=1}^{P} q_p^2\beta_{s,p}^2\mathbb{E}\|\widetilde{\boldsymbol{\phi}}_{p,i-1}\|^2 + q_p^2\sigma_{s,p}^2
\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{p=1}^{P} q_p^2\beta_{s,p}^2 \sum_{m\in\mathcal{N}_p} a_{1,mp}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{m,i-1}\|^2 + a_{1,mp}\mathbb{E}\|\boldsymbol{g}_{1,mp,i}\|^2 + \sigma_s^2 \\
&\leq 2\beta_s^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \sigma_s^2 + \beta_s^2\sigma_g^2 \\
&\quad + 2\sum_{p=1}^{P} q_p^2\beta_{s,p}^2 \sum_{m\in\mathcal{N}_p} a_{1,mp}\mathbb{E}\|\boldsymbol{w}_{m,i-1} - \boldsymbol{w}_{c,i-1}\|^2 \\
&\leq 2\beta_s^2\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \sigma_s^2 + \beta_s^2\sigma_g^2 + 2\beta_s^2\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2. \qquad (4.107)
\end{aligned}$$

The noise term can be bounded as follows by using twice Jensen's inequality:

$$\mathbb{E}\left\|\sum_{p=1}^{P} q_p \boldsymbol{H}_{p,i-1} \sum_{m\in\mathcal{N}_p} a_{1,mp}\boldsymbol{g}_{1,mp,i}\right\|^2 \leq \delta^2\sigma_g^2. \qquad (4.108)$$

We plug the bounds on the gradient noise and the added privacy noise in (4.106):

$$\begin{aligned}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq \left(\alpha + 2\beta_s^2\mu^2\right)\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i-1}\|^2 + \mu^2\sigma_s^2 + (\beta_s^2+\delta^2)\mu^2\sigma_g^2 \\ + \left(2\beta_s^2 + \frac{\delta^2}{1-\alpha}\right)\mu^2\sum_{p=1}^{P}\mathbb{E}\|\boldsymbol{w}_{p,i-1} - \boldsymbol{w}_{c,i-1}\|^2. \qquad (4.109)\end{aligned}$$

We use the bound from Lemma 4.1. Recursively bounding the second-order moment of the error and taking the limit:

$$\begin{aligned}\limsup_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{c,i}\|^2 \leq & \frac{\mu^2\left(\sigma_s^2 + (\beta_s^2+\delta^2)\sigma_g^2\right)}{1-\gamma_c} + O(\mu^2) + \frac{\mu}{1-\gamma_c}\left(2\beta_s^2 + \frac{\delta^2}{1-\alpha}\right)\frac{\kappa_1^2\kappa_2^4}{1-\rho(J_\theta)}\sigma_g^2 \\ = & O(\mu)\sigma_s^2 + O(1)\sigma_g^2 + O(\mu^2). \qquad (4.110)\end{aligned}$$

4.E Proof of Theorem 4.3

It suffices to show the noise generated from the local graph-homomorphic process is Laplacian since we already know that adding Laplacian noise makes the algorithm differentially private (see [86, 131]) with high probability. Thus, it is well known that the product of a uniform random variable $U(0,1)$ with a gamma random variable $\Gamma(2,1)$ results in an exponential random variable Exp(1) [139]. Then $e^{-\boldsymbol{v}_\ell \boldsymbol{v}_m}$ is uniformly distributed on $[0,1]$:

$$\mathbb{P}(e^{-\boldsymbol{v}_\ell\boldsymbol{v}_m} \leq c) = \mathbb{P}(\boldsymbol{v}_\ell\boldsymbol{v}_m \geq -\ln c) = e^{\ln c} = c. \qquad (4.111)$$

But multiplying it by a makes the resulting variable uniformly distributed on $[0,a]$. The modulo p of a uniform random variable is uniform on $[0,\pi]$ so long as a is a multiple of π. Let $a = t\pi$ for some integer t and $\boldsymbol{x} \sim \mathrm{U}(0,a)$. We divide the intervel $[0,a]$ into t disjoint sub-intervals of length π, $[0,a] = [0,1)\cup[1,2)\cdots\cup[(t-1)\pi,a]$. On each of these sub-intervals $[i\pi,(i+1)\pi)$, $\boldsymbol{x}$ is uniformly distributed with:

$$\mathbb{P}\Big(\boldsymbol{x} \leq x | \boldsymbol{x} \in [i\pi,(i+1)\pi)\Big) = x, \qquad (4.112)$$

and so will $\boldsymbol{x} \mod \pi = \boldsymbol{x} - \lfloor \boldsymbol{x}/\pi \rfloor = \boldsymbol{x} - i\pi$ on $[0,\pi]$. Thus, since $a = t\pi$ we get:

$$\begin{aligned}\mathbb{P}(\boldsymbol{x} \leq x) &= \sum_{i=0}^{t-1}\mathbb{P}\Big(\boldsymbol{x} \leq x|\boldsymbol{x} \in [i\pi,(i+1)\pi)\Big) + \mathbb{P}\Big(\boldsymbol{x} \in [i\pi,(i+1)\pi)\Big) \\ &= \sum_{i=0}^{t-1} x\frac{\pi}{a} = x. \qquad (4.113)\end{aligned}$$

This now means that $\boldsymbol{v}_{\ell m} \sim \mathrm{U}(0,\pi)$. Then, taking the difference of two exponential random variables results in a Laplacian. Thus, we require to transform two uniform random variables to two exponential random variables with parameter $\frac{\sigma_g}{\sqrt{2}}$. Taking

$-\frac{\sqrt{2}}{\sigma_g} \ln \boldsymbol{v}_{\ell m}$ results in an exponential random variable:

$$\mathbb{P}\left(-\frac{\sqrt{2}}{\sigma_g} \ln(\boldsymbol{v}_{\ell m}) \leq c\right) = \mathbb{P}\left(\boldsymbol{v}_{\ell m} \geq e^{-\frac{c\sigma_g}{\sqrt{2}}}\right) = 1 - e^{-\frac{\sigma_g c}{\sqrt{2}}}. \tag{4.114}$$

5 Conclusion

In this dissertation, we studied the effect of privatization on some multi-agent systems. We dropped the common claim on the boundedness of the gradients of the risk or loss functions, and introduced a local graph-homormorphic noise construction. Under these conditions, we were able to show that the multi-agent systems continue to be differentially private with high probability.

5.1 Summary of Main Results

In Chapter 2, we studied the convergence of federated learning and improved the algorithm's performance in the MSE sense by introducing importance sampling on the level of client selection and data sampling. We then privatized the algorithm by perturbing the gradients before they were sent to the server. We showed that the effect of the added noise does not alter the bound on the MSE from $O(\mu)$ to $O(\mu^{-1})$ as it does when noisy iterates are shared.

In Chapter 3, we extended the architecture of federated learning to a network of servers connected to multiple clients. We then compared two privatization schemes of the decentralized learning algorithms: random noise perturbation, and graph-homomorphic perturbation. We showed that generating noise in accordance with the graph topology only increases the bound on the MSE to $O(1)$ as opposed to the $O(\mu^{-1})$ introduced by constructing ad-hoc noise.

Finally, in Chapter 4, we focused on a fully decentralized network of agents. We studied the privatization of the general decentralized learning algorithm. We introduced a noise generation scheme called local graph-homomorphic process. We showed that it does not affect the performance of the algorithm since it maintains the MSE bound at $O(\mu)$.

5.2 Future Directions

The study of secure aggregation-type methods suggests that a privacy measure capturing the trade-off between the communication cost and the level of privacy is required. From our study of the local graph-homomorphic process in Chapter 4, we observe that such mechanisms introduce extra communication between the agents to ensure the added noise cancels out and its effect on the learned model is eliminated. Thus, such methods canceled the utility-privacy trade-off and replaced it with a communication-privacy trade-off. Differential privacy does a good job at capturing the former trade-off in the ϵ parameter and its appearance in the MSE bound.

Another interesting study is personalization in federated learning. In applications where the data is non-iid across agents, such as text prediction or recommender systems, personalization is a desired feature. Customizing the global model to target local requirements might help improve performance locally at each agent and provide a more personalized experience. In recommender systems, users would rather get recommendations based on their own interests. We have done some preliminary work in this area. We introduced a new personalization algorithm and are currently studying its performance. With the introduction of personalization, a natural next question is its effect on privacy. By introducing personalization, we could be leaking personal information. Thus, a privatized personalization algorithm is needed.

Finding closed form expressions of the MSD for asynchronous networks and FL will help us answer questions on the effect of the different parameters on performance. One important parameter is the number of local steps. The problem with asynchronous networks is that agents move toward their local models during the local steps. There could be an optimal number of local steps that allow agents to utilize the information in their local dataset and decrease the generalization error without biasing the learned model. From the preliminary work done under this topic, we have shown that the number of local steps does not affect the MSD expression. Instead, the effect appears in higher order terms of the step-size μ.

Finally, similar to the sampling of the agents in FL, we could extend the study of the sampling of agents to a decentralized network. In large networks with limited resources, it might not always be efficient to allow all agents to participate during each iteration. Thus, we could model such a scenario as a time-varying network where each agent samples its neighbourhood. Then, to improve performance, we could introduce importance sampling to ensure that the best neigbours are chosen during each iteration.

Bibliography

[1] C. G. Langton, *Artificial life: An overview.* MIT Press, 1997.

[2] W. Meng, Z. He, R. Teo, R. Su, and L. Xie, "Integrated multi-agent system framework: decentralised search, tasking and tracking," *IET Control Theory & Applications*, vol. 9, no. 3, pp. 493–502, 2015.

[3] J. Hu, L. Xie, J. Xu, and Z. Xu, "Multi-agent cooperative target search," *Sensors*, vol. 14, no. 6, pp. 9408–9428, 2014.

[4] Q. Wang, M. Su, M. Zhang, and R. Li, "Integrating digital technologies and public health to fight covid-19 pandemic: key technologies, applications, challenges and outlook of digital healthcare," *International Journal of Environmental Research and Public Health*, vol. 18, no. 11, pp. 1–50, 2021.

[5] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ Digital Medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[6] J. Xie and C.-C. Liu, "Multi-agent systems and their applications," *Journal of International Council on Electrical Engineering*, vol. 7, no. 1, pp. 188–197, 2017.

[7] M. Oprea, "Applications of multi-agent systems," in *Information Technology.* Springer, 2004, pp. 239–270.

[8] M. Wooldridge, *An Introduction to Multiagent Systems.* John Wiley & sons, 2009.

[9] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[10] A. H. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers Inc., 2014.

[11] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *Proc. International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 20–22 April 2017.

[12] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

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

[14] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[15] V. Bordignon, V. Matta, and A. H. Sayed, "Adaptive social learning," *IEEE Transactions on Information Theory*, vol. 67, no. 9, pp. 6053–6081, 2021.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, pp. 1–9, 2008.

[17] A. H. Sayed, *Inference and Learning from Data.* Cambridge University Press, 2022.

[18] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[19] M. J. Kearns and U. Vazirani, *An introduction to Computational Learning Theory.* MIT Press, 1994.

[20] A. Blum, "On-line algorithms in machine learning," *Online algorithms*, pp. 306–325, 1998.

[21] O. Bousquet, S. Boucheron, and G. Lugosi, *Introduction to Statistical Learning Theory.* Springer, 2004.

[22] C. Lemaréchal, "Cauchy and the gradient method," *Doc Math Extra*, vol. 251, no. 254, p. 10, 2012.

[23] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association.*, vol. 69, no. 345, pp. 118–121, 1974.

[24] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[25] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[26] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, “Crypten: Secure multi-party computation meets machine learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 4961–4973, 2021.

[27] X. Sun, P. Zhang, J. K. Liu, J. Yu, and W. Xie, “Private machine learning classification based on fully homomorphic encryption,” *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 352–364, 2020.

[28] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, New York, Oct 2017, p. 1175–1191.

[29] C. Weng, K. Yang, X. Xie, J. Katz, and X. Wang, “Mystique: Efficient conversions for {Zero-Knowledge} proofs with applications to machine learning,” in *USENIX Security Symposium*, Aug 2021, pp. 501–518.

[30] E. Rizk, S. Vlaski, and A. H. Sayed, “Federated learning under importance sampling,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 5381–5396, 2022.

[31] A. Agarwal and J. C. Duchi, “Distributed delayed stochastic optimization,” in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[32] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, “Parallelized stochastic gradient descent,” in *Proc. Advances in Neural Information Processing Systems*, 2010, pp. 2595–2603.

[33] D. P. Bertsekas, “A new class of incremental gradient methods for least squares problems,” *SIAM J. Optim*, vol. 7, pp. 913–926, 1996.

[34] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks — part i: Transient analysis,” *IEEE Trans. Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.

[35] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[36] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in *ICML AMTL Workshop*, Long Beach, CA, June 2019, pp. 1–28.

[38] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, CA, December 2017, pp. 4424–4434.

[39] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv:1812.07210*, 2018.

[40] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, November 2019.

[41] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv:1610.05492*, 2016.

[42] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. International Conference on Machine Learning*, Long Beach, CA, Jun 2019, pp. 4615–4625.

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

[44] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, December 2019, pp. 5915–5926.

[45] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv:1802.07876*, 2019.

[46] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv:1712.07557*, 2017.

[47] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. International Conference on Learning Representations*, Vancouver, April 2018, pp. 1–14.

[48] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, December 2018, pp. 2525–2536.

[49] A. Khaled, K. Mishchenko, and P. Richtárik, "First analysis of local gd on heterogeneous data," *arXiv:1909.04715*, 2019.

[50] S. U. Stich, "Local SGD converges fast and communicates little," in *International Conference on Learning Representations*, New Orleans, May 2019, pp. 1–19.

[51] J. Wang and G. Joshi, “Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms,” *Journal of Machine Learning Research*, vol. 22, no. 213, pp. 1–50, 2021.

[52] F. Zhou and G. Cong, “On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization,” *Proc. International Joint Conference on Artificial Intelligence*, pp. 3219–3227, July 2018.

[53] H. Yu, S. Yang, and S. Zhu, “Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning,” in *Proc. AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, July 2019, pp. 5693–5700.

[54] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, “Adaptive federated learning in resource constrained edge computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[55] H. Yu, R. Jin, and S. Yang, “On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization,” in *Proc. International Conference on Machine Learning*, Long Beach, CA, Jun 2019, pp. 7184–7193.

[56] C. Xie, S. Koyejo, and I. Gupta, “Asynchronous federated optimization,” *arXiv:1903.03934*, 2019.

[57] 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*, Addis Ababa, Ethiopia, 2020, pp. 1–26.

[58] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” pp. 1–6, 2020.

[59] E. Rizk, S. Vlaski, and A. H. Sayed, “Dynamic federated learning,” in *Proc. IEEE SPAWC*, Atlanta, Georgia, 26–29 May 2020, pp. 1–5.

[60] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” in *Proc. IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1–7.

[61] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, “Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data,” in *Proc. IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1–7.

[62] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, “Fast-convergent federated learning,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2021.

[63] W. Xia, T. Q. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7108–7123, 2020.

[64] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *IEEE International Conference on Communications Workshops*, Jun 2020, pp. 1–6.

[65] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021.

[66] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. International Con-ference on Machine Learning (ICML)*, vol. 97, Long Beach, CA, 09–15 Jun 2019, pp. 4615–4625.

[67] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in *Proc. International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020, pp. 1–27.

[68] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2019.

[69] H. H. Yang, A. Arafa, T. Q. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 8743–8747.

[70] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, New York, Feb 2020, p. 393–399.

[71] K. Yuan, B. Ying, S. Vlaski, and A. H. Sayed, "Stochastic gradient descent with finite samples sizes," in *Proc. International Workshop on Machine Learning for Signal Processing (MLSP)*, Salerno, Italy, 2016, pp. 1–6.

[72] G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio, "Variance reduction in SGD by distributed importance sampling," *arXiv:1511.06481*, 2015.

[73] D. Needell, R. Ward, and N. Srebro, "Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 1017–1025.

[74] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *Proc. International Con-ference on Machine Learning (ICML)*, Lille, France, 2015, pp. 1355–1363.

[75] S. U. Stich, A. Raj, and M. Jaggi, “Safe adaptive importance sampling,” in *Proc. Advances in Neural Information Processing Systems*, Long Beach, CA, 2017, pp. 4381–4391.

[76] D. G. Horvitz and D. J. Thompson, “A generalization of sampling without replacement from a finite universe,” *Journal of the American statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[77] H. O. Hartley and J. N. K. Rao, “Sampling with unequal probabilities and without replacement,” *Ann. Math. Statist.*, vol. 33, no. 2, pp. 350–374, 06 1962.

[78] D. Prokhorov, “Ijcnn 2001 neural network competition,” 2001. [Online]. Available: http://www.csie.ntu.edu.tw/-cj1in/libsvmtools/

[79] Avazu and Kaggle, “Avazu’s click-through rate prediction,” 2014. [Online]. Available: http://www.csie.ntu.edu.tw/-cj1in/libsvmtools/

[80] K. R. W. Brewer and M. Hanif, *Sampling with Unequal Probabilities.* Springer, 1983.

[81] E. Rizk, S. Vlaski, and A. H. Sayed, “Privatized graph federated learning,” *arXiv.2203.07105*, 2022.

[82] B. Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the gan: Information leakage from collaborative deep learning,” in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, New York, Oct 2017, pp. 603–618.

[83] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” in *IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, May 2019, pp. 691–706.

[84] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” in *IEEE symposium on Security and Privacy (SP)*, San Jose, CA, May 2019, pp. 739–753.

[85] L. Zhu and S. Han, “Deep leakage from gradients,” in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec 2019, pp. 17–31.

[86] S. Vlaski and A. H. Sayed, “Graph-homomorphic perturbations for private decentralized learning,” in *Proc. ICASSP*, Toronto, Canada, June 2021, pp. 1–5.

[87] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, “Client-edge-cloud hierarchical federated learning,” in *IEEE International Conference on Communications (ICC)*, Jun 2020, pp. 1–6.

[88] E. Rizk and A. H. Sayed, “A graph federated architecture with privacy preserving learning,” in *IEEE International Workshop on Signal Processing Advances in Wireless Communications*, Lucca, Italy, Sep 2021, pp. 1–5.

[89] B. Wang, J. Fang, H. Li, X. Yuan, and Q. Ling, “Confederated learning: Federated learning with decentralized edge servers,” *arXiv:2205.14905*, May 2022.

[90] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, “Personalized federated learning with differential privacy,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[91] A. Triastcyn and B. Faltings, “Federated learning with bayesian differential privacy,” in *IEEE International Conference on Big Data*, Los Angeles, CA, Dec 2019, pp. 2587–2596.

[92] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, “Ldp-fed: Federated learning with local differential privacy,” in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[93] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[94] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, “Distributed learning without distress: Privacy-preserving empirical risk minimization,” in *Advances in Neural Information Processing Systems*, vol. 31, Montreal, Canad, Dec 2018.

[95] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, “Differentially private distributed online learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1440–1453, 2018.

[96] J. Zhu, C. Xu, J. Guan, and D. O. Wu, “Differentially private distributed online algorithms over time-varying directed networks,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 4–17, 2018.

[97] M. A. Pathak, S. Rane, and B. Raj, “Multiparty differential privacy via aggregation of locally trained classifiers.” in *Advances in Neural Information Processing Systems*, Vancouver, Canada, Dec 2010, pp. 1876–1884.

[98] S. Gade and N. H. Vaidya, “Private learning on networks,” *arxiv.1612.05236*, 2016.

[99] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in *Proc. of ACM SIGSAC Conference on Computer and Communications Security*, New York, 2017, pp. 1175–1191.

[100] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur, and D. Evans, "Privacy-preserving distributed linear regression on high-dimensional data," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 345–364, 2017.

[101] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, May 2017, pp. 19–38.

[102] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *IEEE Symposium on Security and Privacy*, Berkeley, CA, 19 – 22 May 2013, pp. 334–348.

[103] W. Zheng, R. A. Popa, J. E. Gonzalez, and I. Stoica, "Helen: Maliciously secure coopetitive learning for linear models," in *IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, May 2019, pp. 724–738.

[104] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Cryptography with constant computational overhead," in *Proceedings Annual ACM Symposium on Theory of Computing*, Victoria, British Columbia, May 2008, pp. 433–442.

[105] I. Damgård, Y. Ishai, and M. Krøigaard, "Perfectly secure multiparty computation and the computational overhead of cryptography," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, France, May 2010, pp. 445–465.

[106] E. Rizk, S. Vlaski, and A. H. Sayed, "Enforcing privacy in distributed learning with performance guarantees," *arXiv.2301.06412*, 2022.

[107] D. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, pp. 913–926, Nov. 1997.

[108] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, pp. 29–51, 2007.

[109] F. S. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental lms processing for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1465–1480, 2011.

[110] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[111] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optim.*, vol. 20, pp. 1157–1170, Jan 2009.

[112] E. S. Helou and A. R. De Pierro, “Incremental subgradients for constrained convex optimization: A unified framework and new methods,” *SIAM J. Optim.*, vol. 20, no. 3, p. 1547–1572, Dec 2009.

[113] M. G. Rabbat and R. D. Nowak, “Quantized incremental algorithms for distributed optimization,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, 2005.

[114] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[115] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, Sep 2004.

[116] S. Barbarossa and G. Scutari, “Bio-inspired sensor network design,” *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, May 2007.

[117] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in *Proc. IEEE Conf. Dec. Control (CDC)*, Cancun, Mexico, December 2008, pp. 4185–4190.

[118] W. Ren and R. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[119] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, Jun 2006.

[120] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise,” *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan 2009.

[121] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, Aug 2011.

[122] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, “Likelihood consensus and its application to distributed particle filtering,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, Aug 2012.

[123] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug 2012.

[124] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec 2012.

[125] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—part i: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, Dec 2015.

[126] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, 2012.

[127] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments—part I: Agreement at a linear rate," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1242–1256, 2021.

[128] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[129] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2009.

[130] D. Froelicher, J. R. Troncoso-Pastoriza, A. Pyrgelis, S. Sav, J. S. Sousa, J.-P. Bossuat, and J.-P. Hubaux, "Scalable privacy-preserving distributed learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 2, pp. 323–347, 2021.

[131] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[132] B. Ying and A. H. Sayed, "Performance limits of stochastic sub-gradient learning, part ii: Multi-agent case," *Signal Processing*, vol. 144, pp. 253–264, 2018.

[133] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments—part i: Agreement at a linear rate," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1242–1256, 2021.

[134] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.

[135] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[136] 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*, Addis Ababa, Ethiopia, 2020, pp. 1–26.

[137] Ginar, "A review of random number generator (rng) on blockchain," Dec 2019. [Online]. Available: https://medium.com/ginar-io/a-review-of-random-number-generator-rng-on-blockchain-fe342d76261b

[138] Y. Wang and H. V. Poor, "Decentralized stochastic optimization with inherent privacy protection," *IEEE Transactions on Automatic Control*, pp. 1–16, 2022.

[139] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. Wiley, 1995.

Curriculum Vitae

Elsa Rizk
elsa.rizk@epfl.ch

Education

Ecole Polytechnique Fédéral de Lausanne (EPFL) September 2018 – April 2023
PhD in Computer Science

American University of Beirut (AUB) September 2016 – July 2018
ME in Electrical Engineering

American University of Beirut (AUB) September 2012 – July 2016
BE in Computer Engineering

Research Experience

PhD Thesis September, 2018 – April, 2023
EPFL

- "Multi-agent Learning with Privacy Guarantees." Supervisor: Prof. Ali H. Sayed

ME Thesis September, 2016 – July, 2018
AUB

- "On the Entropy of Some Classes of Distributions and their Mixtures." Supervisor: Prof. Ibrahim Abou Faycal

Research Intern May, 2015 – September, 2015
University of Maryland College Park

- "Studying Brain Connectivity at a Structural Level Using fMRI Images." Supervisor: Prof. Joseph Jaja

Research Publications

E. Rizk, S. Vlaski, and A. H. Sayed, "Federated learning under importance sampling," *IEEE Trans. Signal Processing*, vol. 70, pp. 5381-5396, 2022.

E. Rizk, S. Vlaski, and A. H. Sayed, "Enforcing privacy in distributed learning with performance guarantees," arXiv.2301.06412, pp. 1-13, Dec. 2022. (*submitted for publication)*

E. Rizk, S. Vlaski, and A. H. Sayed, "Privatized graph federated leaning," arxiv.2203.07105, pp. 1-13, Dec. 2022. (*submitted for publication)*

Rizk, E., Vlaski, S., and Sayed, A. H., "Local graph-homomorphic processing for privatized distributed systems", Proc. IEEE ICASSP, pp. 1-5, Rhodes Island, Greece, June 2023

M. Issa, R. Nassif, E. Rizk, and A. H. Sayed, "Decentralized semi-supervised learning over multitask graphs," *Proc. Asilomar Conf. Signals*, Systems, and Computers, pp. 1-7, Oct.-Nov. 2022.

E. Rizk and A. H. Sayed, "A graph federated architecture with privacy preserving learning," *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1-5, Lucca, Italy, 2021.

E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," *Proc. IEEE ICASSP*, pp. 1-5, Toronto, Canada, June 2021.

S. Vlaski, E. Rizk, and A. H. Sayed, "Second-order guarantees in federated learning," *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 1-8, Pacific Grove, CA, Nov. 2020.

S. Vlaski, E. Rizk, and A. H. Sayed, "Tracking performance of online stochastic learners," *IEEE Signal Processing Letters*, vol. 27, pp. 1385-1389, 2020.

E. Rizk, S. Vlaski, and A. H. Sayed, "Dynamic federated learning," *Proc. International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1-5, May 2020.

Awards & Honors

IC distinguished services award December, 2022
EPFL

IC distinguished services award December, 2021
EPFL

IC distinguished services award December, 2020
EPFL

Skills

Languages Coded: Python, Matlab, Java, C++, Labview, Latex
Languages Spoken: English, French, Arabic