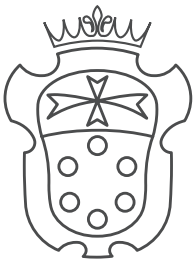


SCUOLA
NORMALE
SUPERIORE

Classe di Scienze

Corso di perfezionamento in
Data Science

XXXVI ciclo

Sense and Sensitivity:
Data Utility and User Privacy in Differentially Private Machine Learning

Settore Scientifico Disciplinare INF/01

Candidato
Dr. Filippo Galli

Relatori
Prof. Tommaso Cucinotta
Scuola Superiore Sant'Anna, Pisa, Italy
Prof. Catuscia Palamidessi
INRIA Saclay; École Polytechnique, Paris, France

Anno accademico 2023/2024

Abstract

This thesis explores existing and novel methods for extracting knowledge from data while preserving users' private information through differentially private machine learning. The central challenge addressed here is handling the sensitivity-utility tradeoff that arises when privatizing queries involving vector averages, which are found everywhere in gradient-based optimization and data science in general. New approaches are thus proposed to provide researchers and practitioners with additional tools to prioritize the use of one strategy over the other, depending on the specific learning context, the privacy expectations, and the accuracy of the resulting model. First, metric privacy concepts are applied to collaborative model training, providing distance-dependent privacy guarantees without pre-defining sensitivity. An online optimization method is then introduced for tuning the clipping threshold concurrently with model training, reducing privacy exposure and computational requirements while improving utility. Efficient strategies for empirically verifying privacy results in the training of large language models are also developed, encouraging practical privacy auditing. Finally, a new perspective is offered on the definition of differential privacy, suggesting that sensitivity with respect to record replacement rather than addition/removal could yield increased utility in federated learning settings. Through theoretical analyses, algorithms, and experimental evaluations, this work presents ideas and actual techniques for optimizing the privacy-utility tradeoff inherent in differentially private machine learning.

List of Figures

1.1 Graphical interpretation of the differential privacy definition (inspired by [Meiser, 2018]). 7

2.1 Learning federated linear models with: (a, b, c) one initial hypothesis and non-sanitized communication, (d, e, f) two initial hypotheses and non-sanitized communication, (g, h, i) two initial hypotheses and sanitized communication. The first two figures of each row show the parameter vectors released by the clients to the server. The last figure of each row illustrates the trend of the validation loss on clients and data not involved in the optimization. 37

2.2 For the experiment on synthetic data, this figure plots the max privacy leakage over clients of the same cluster for a round of training. Intervals with constant privacy leakage indicate that the clients with the largest privacy leakage were not sampled (by chance) to participate in those rounds. 38

2.3 For the experiment on hospital charge data, this histogram plots the empirical distribution of the privacy budget over the clients in a particular configuration: $\nu = 3$, 5 initial hypotheses, seed $= 3$, r is the radius of the neighborhood, and the total number of clients is 2062. 41

2.4 RMSE values for models trained with Algorithm 1 on the Hospital Charge Dataset. Error bars show the empirical standard deviation. Lower RMSE values are better for accuracy. 42

2.5 Effects of the Laplace mechanism in Lemma 2 with different noise multipliers (ref) as a defense strategy against the DLG attack. 44

2.6 The first two plots from the left illustrate the spatial distribution of the samples in g_1 and g_2, respectively, and the third plot shows g_1 and g_2 superimposed together in the same space. 46
2.7 For the experiment with synthetic data, the figure shows the comparison between the personalized and non-personalized models for equal opportunity (a), equalized odds (b), and demographic parity (c), respectively. Experiments were performed for noise multipliers ν of 0.1, 1, 2, and 4. For all the metrics of fairness and the values of the noise multiplier, the personalized model is seen to show improved fairness over the non-personalized model. 47
2.8 For the FEMNIST image classification task, the figure shows the comparison between the personalized and non-personalized models in the same settings described in Figure 2.7. The personalized model again shows improved fairness over the non-personalized model. 49

3.1 The choice of clipping threshold C requires trading off a higher clipping bias at small values, for larger Gaussian noise at large values. Here the clipped, averaged, noised gradient of a CNN for character recognition is compared with the true average gradient at different training iterations $t \in \{100, 250, 500, 750, 950\}$. Note that for some values the sanitized gradient may even have components pointing in the opposite direction w.r.t the true gradient, corresponding to negative cosine similarity. The reported value of cosine similarity is an average over 20 realizations of the Gaussian mechanism. 54
3.2 The Pareto frontiers of the noise multipliers to sanitize $\tilde{g}_t$ and $\tilde{q}_t$, and the chosen values given the heuristic described in Section 3.5, at different privacy requirements. This particular instance comes from the MNIST experiments described in the Section 3.7. 61
3.3 Graphical interpretation of Algorithm 3, where z^{-1} represents the time-shift operator and the numbers in boxes the corresponding lines in the Algorithm. 61
3.4 Accuracy on the MNIST dataset. Higher is better. 68
3.5 Mean Squared Error on the Fashion MNIST dataset. Lower is better. All runs for $\varepsilon = 1$ of `FixedQuantile` result in a diverging optimization and are therefore not included. 69

3.6 Accuracy on the AG News dataset. Higher is better. 70
3.7 $k = 9$. Accuracy on the MNIST dataset. Higher is better. Refer to Table 3.8 for numeric results and optimized hyperparameters. 72
3.8 $k = 9$. Mean Squared Error on the Fashion MNIST dataset. Lower is better. Refer to Table 3.9 for numeric results and optimized hyperparameters. 72
3.9 $k = 9$. Accuracy on the AG News dataset. Higher is better. Refer to Table 3.10 for numeric results and optimized hyperparameters. 73
3.10 $k = 5$. Accuracy on the MNIST dataset. Higher is better. Refer to Table 3.11 for numeric results and optimized hyperparameters. 73
3.11 $k = 5$. Mean Squared Error on the Fashion MNIST dataset. Lower is better. Refer to Table 3.12 for numeric results and optimized hyperparameters. 74
3.12 $k = 5$. Accuracy on the AG News dataset. Higher is better. Refer to Table 3.13 for numeric results and optimized hyperparameters. 74

4.1 An intuitive comparison of computing the score with different calibration strategies: using a shadow model (a) and using a neighboring sample (b). 82
4.2 The AUC of the thresholding classifier for MIA shows a single and prominent peak at the optimal σ value in the *noisy neighbors* strategy. 83
4.3 Efficacy of different strategies for MIA. 86
4.4 Empirical differential privacy measured downstream of training. 87

5.1 Indistinguishability requirements are defined for different pairs of databases: in the upper half according to Definition 10, and at the bottom according to Definition 12 . . . 92
5.2 Normal distributions with the same variance yield different indistinguishability parameters when used in Definition 10 (a) and Definition 12 (b), up to twice as conservative. . . . 94
5.3 Adopting Definition 10 for differential privacy allows to adaptively reduce the sensitivity ${}_mS_2 f$ as in Equation (5.7) while not reducing the average length of gradients and local update vectors. To do so, these vectors can be clipped from above ($C^{(t)}$), below ($c^{(t)}$) and in direction ($a^{(t)}$). . . . 96

5.4 Average cosine similarity among local updates in federated learning (a) and gradients in centralized machine learning (b). 97

List of Tables

2.1 Table of notations for Chapter 2 19
2.2 Qualitative comparison with the most relevant prior research on the topic. More details are provided in Section 2.3. 26
2.3 Regarding the experiment on hospital charge data, for every combination of Noise Multiplier $\times$ Number of Hypotheses, the median and maximum local privacy budgets are reported, over the whole set of clients. These values are averaged over 10 runs with different seeds. $\nu = 0$ means no privacy guarantee and infinite privacy leakage. 40
2.4 Average classification accuracy and standard deviation of a convolutional neural network over three runs seeded with different values. Experiments tested the effect of increasing noise values on the validation accuracy. 43
2.5 NN architecture adopted in the experiments of Section 2.5.1 44

3.1 Dataset and model information shared throughout the experiments. 64
3.2 Best hyperparameters for the MNIST dataset with grid search granularity $k = 7$. Values with * are scaled $\times 10^3$ for better readability. Best `NoDP` result for $\rho = 0.003162$. 66
3.3 Best hyperparameters for the Fashion MNIST dataset with grid search granularity $k = 7$. Best `NoDP` result for $\rho = 0.01467$. All `FixedQuantile` runs diverge for $\varepsilon = 1$. . . 66
3.4 Best hyperparameters for the AG News dataset. Values with * are scaled $\times 10^3$ for better readability. $k = 7$. Best `NoDP` result for $\rho = 0.003162$. 67
3.5 CNN 70
3.6 AutoEncoder 71
3.7 Bag of Words model architecture with a fully connected neural network. 71

3.8 MNIST, $k = 9$, best NoDP for $\rho = 0.003162$, * values are scaled $\times 10^3$. 71
3.9 FashionMNIST, $k = 9$, best NoDP for $\rho = 0.003162$. . . . 75
3.10 AG News, $k = 9$, * values are scaled $\times 10^3$, best NoDP for $\rho = 0.003162$. 75
3.11 MNIST, $k = 5$, * values are scaled $\times 10^3$, best NoDP for $\rho = 0.003162$. 75
3.12 Fashion MNIST, $k = 5$, best NoDP for $\rho = 0.003162$. . . . 75
3.13 AG News, $k = 5$, best NoDP for $\rho = 0.003162$. 76

5.1 Table of notations for Chapter 5 91

Publications and other Contributions

My research efforts towards this thesis have produced the following publications and contributions:

- Filippo Galli, Biswas Sayan, Jung Kangsoo, Tommaso Cucinotta, Palamidessi Catuscia, et al. Group privacy for personalized federated learning. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP)*, volume 1, pages 252–263. SciTePress, 2023c

  Note that an initial, minor version of this work was also presented as:

  Filippo Galli, Sayan Biswas, Kangsoo Jung, Tommaso Cucinotta, and Catuscia Palamidessi. Group privacy for personalized federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022

- Filippo Galli, Kangsoo Jung, Sayan Biswas, Catuscia Palamidessi, and Tommaso Cucinotta. Advancing personalized federated learning: Group privacy, fairness, and beyond. *SN Computer Science*, 4 (6):831, 2023b

- Filippo Galli, Saya Biswas, Kangsoo Jung, Catuscia Palamidessi, and Tommaso Cucinotta. On the adaptive sensitivity of differentially private machine learning. In *The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence (in Conjuction with AAAI 2023)*, 2023a

- Filippo Galli, Catuscia Palamidessi, and Tommaso Cucinotta. Online sensitivity optimization in differentially private learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12109–12117, 2024b

The following work is currently under review:

- Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against LLMs. In *(Under Review) The Fifth Workshop on Privacy in Natural Language Processing (in Conjuction with ACL 2024)*, 2024a

Some of my work fell outside the scope of this thesis. Part of it was initiated before my PhD, or it explores related but separate topics:

- Tommaso Cucinotta, Luigi Pannocchi, Filippo Galli, Silvia Fichera, Sourav Lahiri, and Antonino Artale. Optimum VM placement for NFV infrastructures. In *2022 IEEE International Conference on Cloud Engineering (IC2E)*, pages 205–212. IEEE, 2022
- Giacomo Lanciano, Filippo Galli, Tommaso Cucinotta, Davide Bacciu, and Andrea Passarella. Predictive auto-scaling with openstack monasca. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*, pages 1–10, 2021
- Tommaso Cucinotta, Giacomo Lanciano, Antonio Ritacco, Fabio Brau, Filippo Galli, Vincenzo Iannino, Marco Vannucci, Antonino Artale, Joao Barata, and Enrica Sposato. Forecasting operation metrics for virtualized network functions. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 596–605. IEEE, 2021

As part of the research community on privacy-preserving machine learning, I contributed reviews for papers submitted to the following journals, conferences, and workshops:

- IEEE Transactions on Big Data
- IEEE Transactions on Services Computing
- The Sixth ACM Conference on Fairness, Accountability, and Transparency
- 2023 IEEE International Symposium on Information Theory (ISIT)
- 2023 International Joint Conference on Artificial Intelligence (demonstration track)
- NeurIPS 2023 Workshop on Regulatable ML
- The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence

Contents

1 Introduction **1**
- 1.1 Privacy . . . 2
- 1.2 Differential privacy . . . 5
- 1.3 Differential privacy and machine learning . . . 9
- 1.4 Challenges . . . 10
- 1.5 Contributions . . . 13

2 Distance-based sensitivity in personalized federated learning **15**
- 2.1 Introduction . . . 16
- 2.2 Background . . . 18
  - 2.2.1 Federated learning and personalization . . . 18
  - 2.2.2 Local differential privacy and federated learning . 20
  - 2.2.3 d-privacy . . . 21
  - 2.2.4 Fairness . . . 22
- 2.3 Related works . . . 24
- 2.4 An algorithm for private and personalized federated learning 26
  - 2.4.1 The Laplace mechanism under Euclidean distance in $\mathbb{R}^n$ . . . 29
  - 2.4.2 Sampling from the Laplace mechanism . . . 32
  - 2.4.3 Component-wise variance . . . 33
  - 2.4.4 Limitations of the Laplace mechanism in very high dimensional spaces . . . 34
- 2.5 Experiments . . . 35
  - 2.5.1 Characterizing privacy . . . 35
  - 2.5.2 Fairness analysis . . . 43
- 2.6 Conclusions . . . 50

3 Online optimization of the sensitivity **51**
- 3.1 Introduction . . . 52
- 3.2 Background notions . . . 53
- 3.3 Related works . . . 55

3.4 Method 56
3.5 Privacy Analysis 59
3.6 The OSO-DPSGD Algorithm 60
3.7 Experiments 62
3.8 Conclusions 68
3.9 Appendix: further experiments and details 69

4 Foregoing sensitivity with membership inference attacks 77
4.1 Introduction 78
4.2 Background 79
4.3 Related works 80
4.4 Method 82
4.5 Experiments 84
4.6 Conclusion 85

5 Adaptive sensitivity in the bounded model for differential privacy 88
5.1 Introduction 89
5.2 Sensitivity in the bounded and unbounded models of differential privacy 91
5.3 The unbounded model of DP in machine learning 93
5.4 Conclusions 97

6 Conclusion 98

Chapter 1

Introduction

Understanding intelligence in humans, and animals in general, often serves as a foundation for exploring artificial intelligence. Notably, artificial neural networks were originally inspired by their biological counterpart [Rosenblatt, 1958]. The days of the perceptron are long gone, and for the most part machine learning research strode away from replicating the synaptic responses of our brain cells, but the original structure of a network of computing units still persists. But contamination between areas works in the other direction as well, as advances in machine intelligence are fueling a renewed interest in defining intelligence and sentience. The Turing Test [Turing, 1950] was proposed to probe whether a machine can imitate human responses under specific conditions. The current capabilities of large language models have thus left us wondering: If a machine can mimic human responses well enough to fool a human interrogator, does it mean the machine "thinks" and has "consciousness"? Answering this question is well beyond the scope of this work, but it prompts a discussion of what led us to this point. What is currently acknowledged as a new wave of artificial intelligence benefited from three main favorable conditions that happened to verify roughly at the same time. Firstly, the development of dedicated hardware allowed to vastly speed up data processing and training of machine learning models. One of the first implementations of a convolutional neural network running on a Graphics Processing Unit (GPU) revolutionized the field of computer vision [Krizhevsky et al., 2012], starting a race for ever larger and faster GPUs, the default processor for large-scale machine learning. Secondly, the pervasive advent of broadband internet provided the right infrastructure for the consistent dissemination of public data, especially in the form of images, text, and audio. Driven by a thirst for scientific discovery and capital gain, industry and academia have developed predictive systems that could interpolate and extrapolate through

huge numbers of samples, harnessing what has been called "the unreasonable effectiveness of data" [Halevy et al., 2009]. Thirdly, and for the same reasons, massive research efforts have made long strides in the refinement of algorithms and learning theories that could allow for an efficient processing of the large quantities of data currently available. The back-propagation algorithm [Linnainmaa, 1970; Rumelhart et al., 1986] and the attention mechanism [Vaswani et al., 2017] represent a learning strategy and a unified architecture for machine learning across tasks.

While the quest to derive *sense* from data propels the field of artificial intelligence forward, it is not without its practical and theoretical challenges. Foremost among these is the issue of privacy, as the massive amounts of data processed by AI systems are often inherently personal. This raises significant ethical and regulatory concerns. In fact, the ramifications of not accounting for privacy in machine learning have far-reaching ramifications. One could ask then: isn't the use of publicly-available (but potentially personal) data a solution to the problem? A pragmatic approach would probably argue in favor of a positive answer, but public does not necessarily mean the release was intentional, nor that the scope was unrestricted. Beyond a discussion on ethics and the legality of a practice, we need to start with a rough idea of what we mean by privacy. Simply put, within the context of privacy-preserving machine learning, we will be looking for strategies to learn global patterns and distributions from data, while omitting local and individual information. In a sense, machine learning aims for the same goal, reaching generalization while avoiding overfitting to training data. As we will see, the implications of providing such guarantees will impose stringent requirements on the training strategy. Note that the angle we take to approach the issue focuses on learning. Although the discussion on cryptography and computer security of private data is fundamental and complementary to our study, we will limit our investigation to how to provide some form of privacy guarantees to the individuals whose data will be used in learning algorithms.

1.1 Privacy

Early frameworks for defining and enforcing privacy in statistical data analysis include different takes on the idea of anonymization of the records in the dataset. Although they are still widely in use to this day, and they do contribute to some degree of confidentiality, they present considerable shortcomings that provide the necessary justification for a refined study of privacy-preserving mechanisms. In particular, k-anonymity [Samarati

and Sweeney, 1998], l-diversity [Machanavajjhala et al., 2007], and t-closeness [Li et al., 2006] were prominent methods, each with specific goals and limitations. They describe properties of a "privatized" dataset and numerous efforts have discussed methods to achieve the corresponding guarantees. From a privacy perspective, we can classify the attributes of a database as identifiers, quasi-identifiers, and sensitive attributes. The first includes social security numbers, driver's license numbers, and full names, which are usually removed entirely from the database prior to their release. Quasi-identifiers include age, gender, zip code, profession, and race, which can lead to identification if combined together or with additional information. Sensitive attributes may include medical conditions, income levels and sexual orientation, and other information about an individual that can cause harm and in general have negative consequences should they be disclosed. The following frameworks then describe privacy metrics based on the indistinguishability of individuals starting from information contained in quasi-identifiers and sensitive attributes.

K-anonymity [Samarati and Sweeney, 1998] was designed to ensure that the information for any individual in a dataset could not be distinguished from at least $k-1$ other individuals in the same dataset. Once a set of quasi-identifying attributes is fixed, k-anonimity is provided when every combination of quasi-identifiers in the dataset come in groups of k records, if any. The primary goal was to make re-identification difficult. However, k-anonymity proved vulnerable under certain conditions. For example, if a dataset was released by a hospital where each record is indistinguishable from at least two others based on age and ZIP code, a patient could still be at risk. If all three patients in such a k-anonymous set were recorded as having the same condition, then knowing that a specific individual is in that dataset could reveal their condition through homogeneity. Furthermore, if additional background knowledge is available—such as the fact that only three people of a certain age live in a particular ZIP code—an attacker could deduce private information about an individual.

L-diversity [Machanavajjhala et al., 2007] expanded on k-anonymity by requiring that there be at least l diverse values for sensitive attributes within each group of the dataset. The aim was to prevent attackers from deducing sensitive attributes even when they could identify a person's group. However, l-diversity also had its weaknesses, particularly in dealing with the semantic similarity of attribute values. For instance, if a dataset ensured that each group had three different diagnoses, but all these diagnoses were closely related (such as different types of cancer), the privacy protection was minimal. This similarity means that knowing a person belongs to

a particular group could still reveal that they likely have a type of cancer, despite the diversity of the diagnoses.

T-closeness [Li et al., 2006] sought to further refine these privacy protections by ensuring that the distribution of a sensitive attribute in any group was no more than a threshold t away from the distribution of that attribute across the entire dataset. This method aimed to protect against inferences drawn from the distribution of sensitive attributes. However, t-closeness could vacillate if the overall attribute distribution was unrepresentative of the general population or the attribute was inherently sensitive. For example, if a rare disease was slightly more common within a particular group than in the overall dataset, members of that group could be more likely suspected of having the disease, thus compromising their privacy.

The actual applicability of the above methods presents serious challenges. Generating synthetic data or suppressing data to meet k-anonymity, l-diversity, or t-closeness requirements could significantly reduce the utility of the records. The modifications necessary to comply with these privacy models would often result in the loss of granular information, which could be critical for accurate data analysis. Also, as the number of quasi-identifiers in a dataset increases, it becomes increasingly challenging to apply these models without either compromising the data utility or failing to sufficiently protect privacy. In fact, finding optimal solutions for complying with these privacy models often requires solving NP-hard problems. For instance, partitioning a dataset in groups such that each one contains k indistinguishable individuals from a set of quasi-identifiers, while minimizing the information loss due to the grouping itself, incurs combinatorial complexity.

In addition, there are strong theoretical limitations to the above privacy models. Particularly crucial is the unrestricted potential for adversaries to extract additional individual information from the (anonymized) dataset when they also have access to supplementary sources of background information. In [Narayanan and Shmatikov, 2006], the authors made a field-defining example of a linkage attack, where having access to additional side information led to the re-identification of a record in an anonymized dataset, which in turn allowed for the unintentional disclosure of private information. In particular, within the context of the Netflix Prize - an open competition to find the best algorithm to predict movie ratings for its users - the company released a database of approximately $500'000$ records of user ratings accompanied with the corresponding dates. Several users included a small subset of their ratings also on the Internet Movie Database (IMDb), together with identifiable information on their profiles. With this auxiliary

information, the authors of [Narayanan and Shmatikov, 2006] could then connect, with high confidence, a non-anonymized public profile with limited sensitive information to the anonymized data records in the Netflix dataset, thus exposing the complete list of likes and dislikes of the individuals. Despite this information disclosure may not appear dangerous in and of itself, knowing movie preferences led to detailed and informed guesses on the political, religious, and societal views of the (now completely identified) viewer.

1.2 Differential privacy

Providing absolute privacy guarantees to an individual is provably impossible while maintaining utility. Absolute privacy guarantees are what, in Bayesian terms, is described as the posterior belief being equal to the prior belief, upon observation of a database. Such a scenario would require that access to a database does not enable an adversary to learn anything about an individual that would not be learned without access. To verify this impossibility here follows a different take on the seminal Terry Gross example [Dwork, 2006]. Suppose Alice is a smoker and an adversary has this auxiliary information. The same adversary has access to a medical database, thus concluding smoking causes cancer. Information about Alice has been disclosed (her chances of getting cancer), regardless of Alice's participation in the database.

If the last section underscored the limitations of relying solely on controlling access or anonymizing data to protect privacy, with Alice's example it becomes evident that we should re-evaluate what we mean by privacy entirely. Rather than just considering the absolute amount of information disclosed, we should focus on how much an adversary's knowledge about an individual changes due to their data being included in a dataset. This perspective shift leads us to the concept of *Differential Privacy* (DP). Differential privacy mitigates the risk of disclosure not by removing data or limiting access, but by incorporating randomness into the released information. This approach guarantees that the presence or absence of any individual in the dataset has only a small and quantifiable impact on the output of any analysis. By doing so, differential privacy ensures that the information obtained about any individual remains approximately the same, regardless of their data being included in the dataset or not. In more formal terms, we will include here the definition of differential privacy. Note that, being DP one of the main tools utilized throughout this thesis, we will recall this definition again in the next chapters. This will be useful also in

adapting the notation to the different conventions utilized in different research communities, and to better set the stage for further refinements of the definition, and its later generalizations.

Definition 1 (Differential Privacy)**.** *A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ is (ε, δ) differentially private if $\forall\, S \subseteq$ Range[1]$(\mathcal{M})$ and $\forall\, D, D' \in \mathcal{D}$ differing in one record:*

$$Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} Pr[\mathcal{M}(D') \in S] + \delta \tag{1.1}$$

Note that with respect to [Dwork et al., 2006], Definition 1 already presents a generalization over what is sometimes referred to as "pure" DP, where the term $\delta = 0$. Regardless, we can discuss how to interpret it here. First, with *randomized mechanism* $\mathcal{M}$ we refer to a query on a database, i.e. a function with inputs from the domain of databases $\mathcal{D}$ (i.e. the elements $D \in \mathcal{D}$) with outputs in the set Range$(\mathcal{M})$. Notably, privacy cannot be guaranteed for deterministic functions under this framework. Without delving too much into the details, randomness is understood here as a fundamental building block of indistinguishability. What the inequality roughly means is that the output of the function should not be too different whether the function is applied to either datasets D, D', which differ in only one record, with "too different" being quantified by a multiplicative (e^{ε}) and an additive (δ) factor. Symmetry with respect to D or D' is provided by requiring this property for all pairs of datasets. Further, note that this property needs to be verified for all possible outputs of the query function. Considering that $\mathcal{M}$ could be a real function, these guarantees are expressed in terms of membership to a (measurable) subset of the range of the query.

Although this privacy model may appear obscure, and to some degree hard to justify and communicate to a less technical audience, we claim that part of its strength lies in its broad applicability and overall concise definition, which come from its abstract nature. We further underline the shift in perspective, where privacy is defined in terms of indistinguishability between two worlds: one where the function is applied to a dataset containing a certain private record, and one where it is not. The privacy community frequently discusses different interpretations of the above definition. Having multiple perspectives on the same definition can provide meaningful insights and a stronger grip on its meaning and implications. In that sense, we provide in Figure 1.1 a graphical interpretation of DP.

[1] Recall the range of a function is its image, which is a subset or equal to the function's codomain.

There, the probability density function of $\mathcal{M}$ is plotted when applied to D, D'. Additionally, the dotted line represents $e^{\varepsilon}\mathcal{M}(D')$, and δ (in grey) is the area where pure differential privacy would not be enough. If the area under the curve $e^{\varepsilon}\mathcal{M}(D')$ (that is a scaled probability) plus δ dominates the area under $\mathcal{M}(D)$ for all pairs of D, D' differing in one record, then (ε, δ) differential privacy is satisfied.

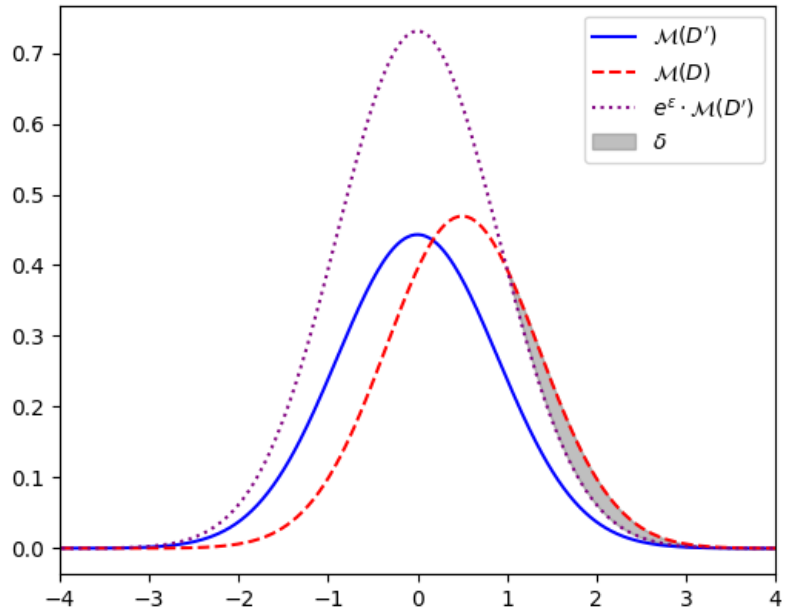


Figure 1.1: Graphical interpretation of the differential privacy definition (inspired by [Meiser, 2018]).

The above characterization provides a set of desirable properties to differential privacy (refer to [Dwork et al., 2014a] for proofs and further details), that will be exploited in the rest of this work:

- Robustness against an adversary's side knowledge: Offering relative guarantees that limit information disclosure effectively circumvents the issue of an adversary using additional background information. For a person who desires privacy protection under this framework, the decision boils down to either contributing to or abstaining from participation in a database. The assurance provided is that whatever extra information an adversary might obtain through database access, would be revealed irrespective of the individual's choice to participate.

- Quantification of privacy loss: the value ε is referred to as privacy

leakage and (along with δ) it is a measure of privacy that allows for a quantitative evaluation of different techniques and algorithms, with lower ε, δ meaning a more private result.

- Composition: multiple ε-differentially private queries on a database incur a privacy leakage that is determined by the privacy losses of the single queries. With (ε, δ)-DP, composition theorems provide the necessary tools to evaluate the overall privacy budget incurred in successive (ε, δ) queries.

- Closure under post-processing: a data analyst seeing the output of a differentially private query cannot increase the privacy leakage by any further analysis of that output. That is, the guarantees provided to the output of a DP query cannot be degraded by downstream operations on that output.

In order to provide a better understanding of differential privacy, here follows a list of limiting factors and open problems. Far from being exhaustive, it is included in an attempt to give a broader overview of the topic:

- Differential privacy does not provide any guarantee that access to the database will not disclose information, be it to a trusted party or an adversary. In fact, to preserve utility, such guarantees would be impossible. It is important to note a key difference in the settings in which privacy and security operate: security aims at obfuscating information from a third party other than the individual who owns the information and the individuals who are supposed to have access to it; conversely, privacy aims at obfuscating information to anyone other than the owner. In the latter, an adversary and a data analyst are seen as equal. Therefore, preventing any access to information to the adversary equally denies any utility to a data analyst.

- As mentioned, an ε-differentially private random function querying a database incurs a privacy loss value of ε. For practical implementations of information disclosure in statistical databases, it is then mandatory to identify a *privacy budget* that the data curator and the data owners are willing to provide the analysts. Once the privacy budget is exhausted (due to the compositionality of differentially private queries), the database cannot be queried anymore without failing to provide the pre-determined level of privacy.

- Pure ε-differential privacy is impractical to implement with real-valued functions in floating-point arithmetic, as some values may be output with null probability with database D and non-negligible probability with database D', because of rounding errors.

- It is not always easy to determine the *granularity* of differential privacy, depending on the structure of the data that need to be privatized. The formulation introduced earlier takes into account tabular data, but it is not trivial to extend the results to other data structure, graph data being a notable example: how are adjacent graphs defined? Are the privacy guarantees desired at a node or edge level?

1.3 Differential privacy and machine learning

Although searching for a unique and all-encompassing definition of Machine Learning (ML) is beyond the scope of this dissertation, we will refer broadly to ML — and to a certain extent, to artificial intelligence — as the study of parametric models designed to estimate and generalize from distributions of the training data, focusing on building models that adapt and optimize performance based on data feedback, unlike traditional statistics that often emphasize hypothesis testing and parameter estimation. Zooming out from the specific task and type of data, this definition allows us to include a multitude of training strategies, different types of models, and the overall objectives. Within that context, a machine learning algorithm queries a dataset of records and returns a set of parameters that together with the associated parametric function aim to model the desired patterns in the dataset. If the dataset includes personal records we wish to keep differentially private, we could identify the randomized mechanism $\mathcal{M}$ in Definition 1 as the algorithm that returns the parameters. Note that in general, these algorithms are deterministic (or the underlying randomness is unknown) and therefore the privacy guarantees need to be implemented by introducing DP mechanisms. Considering the definition of differential privacy, where we require the outcome of a random function on a dataset to be roughly equivalent (in probabilistic terms) whether a record is included or not, we are essentially bounding the sensitivity[2] of the function itself. More formally:

[2] *Sensitivity* may have different definitions according to the specific implementation of differential privacy and ultimately depends on the DP guarantees we are aiming to provide.

Definition 2 (ℓ_2 sensitivity)**.** *The ℓ_2 sensitivity of a deterministic real function $f : \mathcal{D} \to \mathbb{R}^k$ is*

$$S_2(f) = \max_{\substack{D,D' \in \mathcal{D} \\ D \text{ adj } D'}} \|f(D) - f(D')\|_2 \tag{1.2}$$

where $D \operatorname{adj} D'$ is any pair of adjacent datasets, i.e. differing in at most one record.

Finally, in order to turn the deterministic function f into its probabilistic DP approximation, we may adopt the Gaussian mechanism, where we replace f with a DP approximation, according to Theorem 1.

Theorem 1 (Gaussian mechanism [Dwork et al., 2014a])**.** *Let $\varepsilon \in (0, 1)$ be arbitrary and $f : \mathcal{D} \to \mathbb{R}^k$. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian mechanism*

$$f \to f + \rho,\ \rho \sim \mathcal{N}(0, I_k\sigma^2) \tag{1.3}$$

with parameter $\sigma > cS_2(f)/\varepsilon$ is (ε, δ)-differentially private.

1.4 Challenges

To introduce the use of such mechanisms and their issues, we will provide an example of DP sanitization of an ML model, that is we will show a differentially private approximation of the Principal Component Analysis (PCA). This strategy has been previously explored in foundational works such as [Dwork et al., 2014b] and [Abadi et al., 2016]. However, this thesis will adopt a more practical approach to describing the procedure with the aim of providing tangible insights into the issues that will be addressed throughout this work. This hands-on perspective is intended to bridge the gap between theoretical constructs and their practical implications, offering a clearer understanding of the challenges inherent in implementing differential privacy effectively.

Principal Component Analysis (PCA) is a mathematical technique for dimensionality reduction of a dataset. It aims at transforming a number of possibly correlated variables into a set of linearly uncorrelated variables called principal components. Assuming a dataset $X \in \mathbb{R}^{n \times d}$ of n of records with zero mean and unit variance, PCA computes the covariance matrix $C \in \mathbb{R}^{d \times d}$:

$$C = \frac{1}{n-1} X^T X = \frac{1}{n-1} \sum_{i=1}^{d} x_i^T x_i \tag{1.4}$$

Then follows the eigenvalue decomposition of S:

$$CV = V\Lambda \tag{1.5}$$

where V is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. By selecting the first k sorted eigenvectors one constitutes the transformation matrix $W \in \mathbb{R}^{d\times k}$, which can be used to transform to and from the reduced space: $X_{\text{PCA}} = XW$ and $X_{\text{PCA}}W^T = X'$, where X is approximately equal to X'.

Differentially Private PCA (DP-PCA) is an adaptation of the standard Principal Component Analysis (PCA) that incorporates differential privacy. In order to compute the parameters of our PCA model, i.e. the transformation matrix W, we need to query the dataset for the covariance matrix. We notice however that the sensitivity $S_2(C)$ is not bounded as the terms in the summation in Equation 1.4 are also unbounded. In particular, if we compute the sensitivity of the summation Σ:

$$S_2(\Sigma) = \max_{\substack{D,D'\in\mathcal{D}\\ D \text{ adj } D'}} \left\|(X^TX) - (X^TX + x^Tx)\right\|_2 \tag{1.6}$$

$$= \max_{\substack{D,D'\in\mathcal{D}\\ D \text{ adj } D'}} \left\|x^Tx\right\|_2 \tag{1.7}$$

where x^Tx signals the addition of a record (from $\mathbb{R}^{1\times d}$), and it constitutes a rank-1 matrix where each entry is x_jx_k for the j-th and k-th attribute of x. Therefore, to apply the Gaussian mechanism, we need to enforce a limit p to the norm of x, by *clipping* the records:

$$\text{clip}(x,p) = \begin{cases} x \cdot \frac{p}{\|x\|} & \text{if } \|x\| > p \\ x & \text{otherwise} \end{cases} \tag{1.8}$$

In fact, we see that the Frobenius norm (which is also an upper bound for the ℓ_2 norm of the matrix for sensitivity purposes) of x^Tx is the square root of the sum of squares of all entries:

$$\|x^Tx\|_F = \sqrt{\sum_{j=1}^{d}\sum_{k=1}^{d}(x_jx_k)^2} = \|x\|_2 \tag{1.9}$$

which, for clipped records, maxes out at p. Finally, we can compute the sanitized covariance matrix by means of the Gaussian mechanism:

$$C_{DP} = \frac{1}{n-1}\left(\sum_{i=1}^{d}\text{clip}(x_i,p)^T\text{clip}(x_i,p) + \rho\right) \tag{1.10}$$

$$\rho \sim \mathcal{N}(0, I_d \cdot (cp/\varepsilon)^2) \tag{1.11}$$

with c being computed from the required privacy parameters according to Theorem 1. By virtue of DP being immune to post-processing, one can follow the classical steps to find the (now differentially private) transformation matrix. The implications of the choice of p are now clear: the summation can be vastly affected by reducing the threshold, giving a biased estimator of the covariance matrix, as in Equation 1.10. But increasing p introduces a Gaussian noise with a larger standard deviation, as in Equation 1.11.

Having illustrated the application of differential privacy in Principal Component Analysis, where the trade-off between data utility and privacy protection becomes evident, we now turn to a broader issue central to this thesis. The challenge of implementing differential privacy in queries involving the averaging of vectors in $\mathbb{R}^d$ epitomizes the tension between two critical aspects. To maintain privacy guarantees, it is necessary to clip the norm of these vectors and introduce noise proportional to the clipping threshold p. This setup gives rise to an intrinsic trade-off:

- Choosing a smaller p may lead to the introduction of less noise, however, it does so at the cost of inducing a significant bias in the computed average.

- Conversely, opting for a larger p tends to minimize the bias but results in increased noise.

The implications of this choice extend deeply into the functioning of contemporary machine learning models, where averaging plays a pivotal role. Notably, in algorithms like Stochastic Gradient Descent (SGD) and its variants, which are foundational in determining the optimal set of parameters to minimize a predefined cost function. This optimization is achieved through iterative queries to the dataset and subsequent model refinements. A smaller p results in a reduced average, consequently diminishing the magnitude of the iterative updates. This can potentially trap the model in suboptimal local minima and require an increased number of queries to the dataset [Abadi et al., 2016]. In turn, this gives further privacy concerns as cumulative DP queries reduce the overall privacy budget. A reduced sensitivity introduces questionable results in terms of fairness and robustness, focusing on learning the bulk of the distribution while reducing the impact of outliers and less represented segments of the population [Suriyakumar et al., 2021].

1.5 Contributions

This thesis explores different methods to extract meaning from data while managing the challenge of sensitivity in Differentially Private Machine Learning (DPML). The goal is to propose novel approaches that strike a balance between uncovering valuable insights and protecting privacy. In particular, this work contributes to the field through several key developments:

- Approach the study of DPML from a fresh angle by incorporating metric privacy techniques, eliminating the necessity of defining a clipping threshold entirely. In particular, Chapter 2 will discuss the use of d-privacy – a generalization of differential privacy – to provide distance-dependent guarantees to the individuals taking part in collaborative training of machine learning models. The idea is derived from the field of location privacy, shifting the focus from a location in $\mathbb{R}^2$ to a parameter vector of an ML model in a generic space $\mathbb{R}^d$. Abandoning a predefined sensitivity under this framework allows for increased fairness and utility in some contexts while being under-performing in others, which we identify in detail. Additionally, it results in the introduction of novel theoretical and empirical results that provide a novel point of view on the issue.

- Providing a theoretical framework and results to address the online optimization of the clipping threshold in the classical setting of differentially private learning. In Chapter 3, we draw from the results in online hyperparameter optimization to derive an update rule to optimize the sensitivity concurrently to the optimization of the machine learning model parameters. This chapter will focus on the reduction of privacy exposure and the increase in utility of training a set of machine learning models whose hyperparameters are optimized via grid search. By optimizing the clipping threshold directly during model training, we show how to better exploit the overall grid search privacy budget on each single run, leading to state-of-the-art results.

- Introducing efficient strategies to verify empirical Differential Privacy (eDP) guarantees, focusing on large language models. In Chapter 4 we abstain from providing *a priori* DP guarantees, and instead, we verify the privacy metrics *a posteriori* by means of Membership Inference Attacks (MIA), whose results give a lower bound to the eDP parameters. This allows us to sidestep the problem of setting

an appropriate sensitivity level and training could potentially be carried out without explicitly considering privacy countermeasures. We carry out this study to encourage the use of privacy auditing techniques in accordance with current regulatory frameworks, and we do so by improving the computational and memory efficiency of the MIA attacks.

- Proposing a novel position on the definition of DP, which we posit as a potential step forward in Differentially Private Machine Learning (DPML) with adaptive sensitivity. In particular, Chapter 5 challenges the prevalent position in DPML, where the sensitivity is defined starting from the notion of *addition/removal* of a record, to reflect the choice given to an individual when asked to participate to a database. We instead suggest that, under certain conditions, sensitivity with respect to the *replacement* of a record with another may yield increased utility and improved performance, detailing potential avenues for future experiments within the context of collaborative machine learning.

Lastly, Chapter 6 concludes the thesis, summarizing the results and outlining the open problems.

Chapter 2

Distance-based sensitivity in personalized federated learning

In this chapter, we introduce a novel approach to address sensitivity issues in differentially private machine learning in a distributed setting, where we will incur into the problem of bounding the average of parameter vectors. Specifically, we explore this in Federated Learning (FL), a form of distributed learning initially designed to enhance privacy for users by allowing data to remain local to each client. Federated learning involves peers or clients processing their data locally and sharing only model updates, which helps protect privacy by preventing the exposure of raw data. Despite its advantages, federated learning can introduce challenges, such as private information leakage, inadequate model personalization, and disparities in model fairness across different client groups. To tackle these issues, we propose a method that leverages d-privacy, or metric privacy, which provides localized differential privacy guarantees by using a distance-oriented obfuscation to preserve the data's topological distribution. Our approach not only facilitates personalized training within a federated setting but also enhances group fairness, as measured by standard metrics, compared to models trained under conventional FL methods. We will discuss the theoretical foundations of our method and demonstrate its effectiveness through experimental validation on real-world datasets. The results included in this chapter were first published as part of the following papers:

> Filippo Galli, Biswas Sayan, Jung Kangsoo, Tommaso Cucinotta, Palamidessi Catuscia, et al. Group privacy for personalized federated learning. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP)*, volume 1, pages 252–263. SciTePress, 2023c

Filippo Galli, Kangsoo Jung, Sayan Biswas, Catuscia Palamidessi, and Tommaso Cucinotta. Advancing personalized federated learning: Group privacy, fairness, and beyond. *SN Computer Science*, 4 (6):831, 2023b

2.1 Introduction

There has been a significant surge in the value and need of data to perform various kinds of statistical analyses, typically with a requirement of collecting and centralizing massive datasets from users, often containing their sensitive personal information. There are multiple advantages to having access to all the necessary data in a single location, mostly related to efficiency: faster computation, reduced communication costs between the computing and storage nodes, and, in general, a more direct control over the population of data points. However, alongside this massive rise in the need to collect and store data, the risks of violation of the users' privacy are becoming more and more significant and concerning [NIST; Le Métayer and De, 2016].

Federated learning [McMahan et al., 2017a] (FL) is a machine learning approach that uses user devices for both data collection and model training without sending raw data to a central server. The central server distributes a model to selected users for local optimization and aggregates the updates to refine the global model. This process repeats until the model converges. Nonetheless, avoiding the release of user's raw data only provides a lax protection from potential attacks violating the users' privacy [Hitaj et al., 2017; Nasr et al., 2019; Zhu et al., 2019], as it falls in the pitfall of "only releasing summary statistics" [Dwork et al., 2014a], which is the set of updated model parameters transmitted to the central server.

Researchers have integrated differential privacy into the Federated Learning process to ensure that outputs do not significantly change whether or not a specific personal record is included [Dwork et al., 2006]. However, the central model for DP still risks security breaches from a single failure point, potential adversarial actions and it does not solve the problem of data centralization. To avoid these risks, Local Differential Privacy (LDP) has gained attention, allowing users to perturb their data locally before transmission [Duchi et al., 2013; Kairouz et al., 2016]. In FL, LDP helps to mask individual data contributions during model updates, balancing privacy with potential reductions in model accuracy due to randomization.

As previously discussed in Chapter 1, differential privacy necessitates a bounded sensitivity of the query function. Since the vectors of model pa-

rameters are not inherently bounded, a significant error source arises from *clipping* the domain of information released by users [Andrew et al., 2021]. This enforced truncation results in a clipped distribution of parameter vectors as perceived by the server. Consequently, the aggregation phase of the optimization process often becomes biased [Suriyakumar et al., 2021]. Specifically, the model's fairness is compromised when parameter vectors from *minority* groups in the dataset are excluded post-truncation, leading to biases against users with under-represented data [Li et al., 2020; Suriyakumar et al., 2021].

This issue is critical in the context of personalized FL when the empirical distribution is employed for tailoring the models to distinct groups in the population. A personalized approach improves model effectiveness in applications like natural language processing tailored to regional dialects, recommender systems based on political affiliations, and facial expression recognition for diverse ethnic groups. Given that the clustering of parameter vectors must be performed on sanitized user-reported values, receiving an unbiased distribution is crucial for fostering fairness in the global model.

To address the above-mentioned issues, we investigate the possibility of providing local privacy guarantees and allowing for personalized models in FL. Thus, we propose the adoption of d-privacy mechanisms, devised within the field of location privacy [Chatzikokolakis et al., 2013; Bordenabe et al., 2014; Fernandes et al., 2021], to obfuscate the information released locally by each user in the federated training. Using d-privacy sidesteps the issue of clipping, thus maintaining the original topology of the distribution. Concurrently, we define an algorithm for personalized federated learning that takes advantage of distance metric-based privacy guarantees for clustering participating users.

More precisely, this chapter presents the following contributions to the field:

1. It provides an algorithm for the collaborative training of machine learning models, which build on top of state-of-the-art strategies for model personalization.

2. It formalizes the privacy guarantees in terms of d-privacy to provide local privacy guarantees in the context of personalized federated learning.

3. It presents a study of the Laplace mechanism on high dimensions, under Euclidean distance, based on a generalization of the Laplace distribution in $\mathbb{R}$, and we give a closed-form expression.

4. It provides an efficient procedure for sampling from such distribution.

5. It shows that personalized federated learning under formal privacy guarantees improve group fairness significantly compared to the non-personalized federated learning framework, and establish that this method enhances the trade-off between privacy and fairness.

The rest of the Chapter is organized as follows. Section 2.2 introduces fundamental notions for federated learning and differential privacy. Section 2.3 discusses related work. Section 2.4 explains the proposed algorithm for personalized federated learning with group privacy. Section 2.5 validates the proposed procedure through experimental results. Section 2.6 concludes the Chapter.

2.2 Background

2.2.1 Federated learning and personalization

Collaborative learning with privacy and communication constraints has received much attention since the introduction of federated learning [McMahan et al., 2017a; Konečny et al., 2016a,b; Hard et al., 2018], which aims to train a global machine learning model on a distributed collection of non-i.i.d. datasets stored on devices whose raw data cannot be disclosed. Focusing on the personalized federated learning setting, we adopt the notation of [Ghosh et al., 2020] to cast the problem in the framework of stochastic optimization and find the set of minimizers θ_j^* with $j \in \{1, \dots, k\}$ of the cost functions

$$F(\theta_j) = \mathbb{E}_{z \sim \mathcal{P}_j} \left[f(\theta_j; z) \right], \tag{2.1}$$

where $\mathcal{P}_j$ is the data distribution which can only be accessed through a collection of datasets $Z_c = \{z_i | z_i \sim \mathcal{P}_j, z_i \in \mathbb{D}\}$ with $c \in C = \{1, \dots, N\}$, the set of clients. C is partitioned in k disjoint sets

$$S_j^* = \{c \mid \forall z \in Z_c,\, z \sim \mathcal{P}_j\} \, \forall \, j \in [k] \tag{2.2}$$

The mapping $c \to j$ is unknown and we rely on estimates S_j of the membership of Z_c to compute the empirical cost functions

$$\tilde{F}(\theta_j) = \frac{1}{|S_j|} \sum_{c \in S_j} \tilde{F}_c(\theta_j; Z_c) \tag{2.3}$$

Table 2.1: Table of notations for Chapter 2

Notation	Description
$\mathcal{X}$	Domain of original values
$d(.)$	Distance metric on $\mathcal{X}$
$\mathcal{Y}$	Domain of secrets
$\Pr_{\mathcal{K}}[y\|x]$	Probability that mechanism $\mathcal{K}$ reports $x \in \mathcal{X}$ as $y \in \mathcal{Y}$
$\mathbb{D}$	Domain of the data points held by the users
k	Number of clusters, hypotheses, and distributions
n	Number of model parameters
$f(.)$	$f\colon \mathbb{R}^n \times \mathbb{D} \mapsto \mathbb{R}_{\geq 0}$; Cost function
$\mathcal{P}_j$	Probability density function of the j^{th} distribution
Z_c	Collection of data points held by client c
S_j^*	Subset of clients whose data is sampled from $\mathcal{P}_j$
S_j	Estimate of S_j^*
θ_j^*	Minimizer of $F(\theta_j)$
θ_j	Parameter vector
$\tilde{\theta}_j^*$	Estimate of θ_j^*
$F(\theta_j)$	Expectation of $f(.)$ over $z \sim \mathcal{P}_j$
$\tilde{F}(\theta_j)$	Empirical estimate of $F(\theta_j)$
$\tilde{F}_c(\theta_j; Z_c)$	$\tilde{F}(\theta_j)$ evaluated on client's c data points
$\hat{\theta}_{j,c}^{(t)}$	Sanitized and updated j^{th} parameter vector released by c
$\hat{\xi}_c$	The sanitized update to the model parameters by c
$\mathcal{L}_\varepsilon$	$\mathcal{L}_\varepsilon\colon \mathbb{R}^n \mapsto \mathbb{R}^n$; Laplace mechanism providing ε-d-privacy
$\gamma_{\varepsilon,n}(r)$	Gamma distribution with shape n and rate ε
$\mathbb{S}_n(r)$	Surface of the sphere in $\mathbb{R}^n$ of radius r
$\Gamma(.)$	Gamma function
ν	Noise multiplier
${}_1x_n$	A unit vector in $\mathbb{R}^n$
Δ	A generic random vector

with

$$\tilde{F}_c(\theta_j; Z_c) = \frac{1}{|Z_c|} \sum_{z_i \in Z_c} f(\theta; z_i) \tag{2.4}$$

The cost function $f\colon \mathbb{R}^n \times \mathbb{D} \mapsto \mathbb{R}_{\geq 0}$ is applied on $z \in \mathbb{D}$, parametrized by the vector $\theta_j \in \mathbb{R}^n$. Thus, the optimization aims to find, $\forall\, j \in [k]$,

$$\tilde{\theta}_j^* = \arg\min_{\theta_j} \tilde{F}(\theta_j) \tag{2.5}$$

Considering the notation-heavy nature of this Chapter, a summary of the main notational elements specific to this discussion can be found in Table 2.1.

2.2.2 Local differential privacy and federated learning

Recall differential privacy [Dwork et al., 2006] was introduced as a property of queries of statistical databases to measure information leakage, and it is used to formalize privacy guarantees by mathematically ensuring that the output of a given query does not change (in probabilistic terms) irrespective of whether a specific record is contained in it or not. Refer to Section 1.2 for a more detailed discussion and the corresponding definition.

To mitigate one of the major drawbacks of the central model of DP, which is that it requires a dependency on a trusted central server, a local variant of the central model has been studied recently by the community and called local differential privacy (LDP) [Duchi et al., 2013], where the users locally obfuscate their data and send the noisy data to the server such that a particular entry of a user's data probabilistically does not have an impact on the outcome of the query.

Definition 3 (Local differential privacy)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ denote the spaces of the original and the perturbed noisy data, respectively* [1]*. A mechanism $\mathcal{M}$ provides* (ε, δ)-local differential privacy *if, for all $x,\, x' \in \mathcal{X}$, and all measurable $S \subseteq \mathcal{Y}$, we have:*

$$\Pr\left[\mathcal{M}(x) \in S\right] \leq e^{\varepsilon} \Pr\left[\mathcal{M}(x') \in S\right] + \delta \tag{2.6}$$

The local model for differential privacy [Kasiviswanathan et al., 2011; Duchi et al., 2013] can be derived from Definition 1 when $x,\, x'$ are taken

[1] Usually in LDP $\mathcal{X}$ and $\mathcal{Y}$ are discrete domains but for the sake of uniformity with the other definitions we extend LDP to continuous domains.

to be datasets of only one record. Therefore LDP is a stronger condition as it requires the mechanism to satisfy DP for any two values of the domain of data $\mathcal{X}$.

There are different approaches studied in the literature that apply DP in machine learning [Shokri and Shmatikov, 2015a; Abadi et al., 2016; McMahan et al., 2017b], but, possibly, one of the most successful lines of work is based on evaluating how much each user participating in the training dataset has contributed to the trained model. Essentially, gradient-based optimization of a machine learning model, parametrized by θ, works by computing the gradient of a loss function $\nabla f(\theta, z)$ with respect to θ, for a number of iterations, evaluated over a batch of z, and updating the parameters according to the (stochastic) gradient descent algorithm [Bottou, 2012]. If $\|\nabla f(\theta, z)\|_2$ is clipped to a value $g_{\max}$, then the function querying the dataset has bounded sensitivity and, thus, the Gaussian mechanism with the properties described in [Abadi et al., 2016] can be applied to sanitize the queries to a user's data point z.

In the context of FL, the procedure described in [McMahan et al., 2017b; Andrew et al., 2021] requires the clients to perform a few iterations of gradient descent over their local datasets Z_c and only report the difference in the parameter vector before and after the update, clipped in norm to a value $g_{\max}$, to the central server. The server then applies the Gaussian mechanism to compute sanitized average updates to the model parameters, thus preserving DP with a preferred privacy level.

2.2.3 d-privacy

d-privacy [Chatzikokolakis et al., 2013] is a generalization of DP for any domain $\mathcal{X}$, representing the space of original data, endowed with a distance measure d: $\mathcal{X}^2 \mapsto \mathbb{R}_{\geq 0}$, and any space of secrets $\mathcal{Y}$. In particular:

Definition 4 (d-privacy)**.** *A random mechanism $\mathcal{R} : \mathcal{X} \mapsto \mathcal{Y}$ is called ε d-private if for all $x_1,\ x_2 \in \mathcal{X}$ and measurable $S \subseteq \mathcal{Y}$:*

$$\Pr\left[\mathcal{R}(x_1) \in S\right] \leq e^{\varepsilon d(x_1, x_2)} \Pr\left[\mathcal{R}(x_2) \in S\right] \tag{2.7}$$

Note that when x_1, x_2 are elements of the domain of databases, and d is the distance on the Hamming graph of their adjacency relation, then Definitions 1 and 4 are equivalent, reducing the applicability of d-privacy to that of DP. It is also worthy of note that, in general, $\mathcal{X}$ and $\mathcal{Y}$ may be different. However, in the context of this study, we have the space of original data and the space of secrets to be the same, i.e., $\mathcal{X} = \mathcal{Y}$.

This notion of metric-based privacy has been found particularly effective in the context of location privacy [Chatzikokolakis et al., 2013; Andrés et al., 2013], where $\mathcal{X} = \mathbb{R}^2$ and d is the Euclidean distance. The authors show how the formal privacy guarantees degrade gracefully with the distance between two points, which is especially beneficial when the service provider or the server is interested in an approximate value of the true location of the users, thus striking a balance between the privacy level required by the users and the statistical accuracy of their reported values. This approach differs from that of DP, preferable only when an aggregated information is required. To sanitize the values in $\mathcal{X}$, [Chatzikokolakis et al., 2013] introduces a generalized Laplace mechanism, although an analytical form of the probability distribution for, or the sampling procedure from, a domain in $\mathbb{R}^n$, for $n > 2$, has not been presented. It is worth noting that the clients may decide the standard deviation of the noise they choose to inject to their real data based on a radius within which they want to be indistinguishable. For instance, providing a sanitized location with a noise of standard deviation in the order of 1 km may be sufficient for a user to report her rough location to query for suggestions on nearby restaurants to a service provider, and at the same time concealing her exact coordinates.

2.2.4 Fairness

With the recent surge of interest in building ethical ways to train machine learning models, the topic of fairness in machine learning has been in the spotlight, and, correspondingly, various metrics and algorithms to quantify and establish fairness in model training have been studied from a variety of perspectives and in different contexts [Verma and Rubin, 2018; Hanna and Linden, 2009; Makhlouf et al., 2021]. Most fairness metrics consider the simple case of having a *privileged* group and an *unprivileged* group in the population. Under this assumption, typically one attribute of the dataset is selected as a sensitive attribute (e.g., gender, race, etc.) that defines the privileged and the unprivileged groups. The goal of fairness in machine learning is to ensure fair and non-discriminated results regardless of an individual's association with any sensitive attributes. The two main notions of fairness considered by the community are individual fairness and group fairness: *Individual fairness* [Dwork et al., 2012] claims that similar individuals should be treated similarly, and *group fairness* requires that different demographic subgroups should receive equal treatment with respect to their sensitive attributes. While both notions of fairness are important, we will focus on group fairness because our goal is to analyze and miti-

gate the potential bias against certain groups (e.g. demographic groups) through personalization techniques. The following metrics are considered for evaluating group fairness, and they refer to the outcome of classification tasks in machine learning, where $\hat{Y} = 1$, $\hat{Y} = 0$ is used to represent the positive and negative prediction respectively, while $S = 1$, $S = 0$ to represent the privileged and unprivileged groups.

Definition 5 (Demographic parity [Dwork et al., 2012])**.** *Demographic parity is achieved by a system when the prediction $\hat{Y}$ of the target label Y is statistically independent of the sensitive attributes S, i.e.,*

$$\Pr\left[\hat{Y} = 1|S = 1\right] = \Pr\left[\hat{Y} = 1|S = 0\right] \tag{2.8}$$

Imposing demographic parity has often a strong negative impact on accuracy, and, consequently, more refined notions were proposed afterward. In particular:

Definition 6 (Equalized odds)**.** *A system satisfies* equalized odds *if its prediction $\hat{Y}$ is conditionally independent of the sensitive attribute S given the target label Y and $y \in \{0, 1\}$:*

$$\Pr\left[\hat{Y} = 1|Y = y, S = 1\right] = \Pr\left[\hat{Y} = 1|Y = y, S = 0\right] \tag{2.9}$$

In other words, the notion of equalized odds requires the privileged and unprivileged groups to have equal true positive rates and equal false positive rates. Equal opportunity is a relaxation of equalized odds, in the sense that it only requires equal true positive rates across the groups.

Definition 7 (Equal opportunity [Hardt et al., 2016])**.** *Equal opportunity is satisfied by a system if its prediction $\hat{Y}$ is conditionally independent of the sensitive attribute S given the target label Y*

$$\Pr\left[\hat{Y} = 1|Y = 1, S = 1\right] = \Pr\left[\hat{Y} = 1|Y = 1, S = 0\right] \tag{2.10}$$

In practice, however, it is difficult to obtain perfect equality for any of the aforementioned notions. Hence, typically the aim is to minimize the absolute value of the difference between the privileged and unprivileged groups, rather than requiring this difference to be exactly zero. For instance, the *demographic parity difference* is defined as

$$\left|\Pr\left[\hat{Y} = 1|S = 1\right] - \Pr\left[\hat{Y} = 1|S = 0\right]\right| \tag{2.11}$$

and similarly for the *equalized odd difference* and *equal opportunity difference*.

2.3 Related works

With the generalized Federated Averaging algorithm [Konečny et al., 2016a; Reddi et al., 2020] to solve the empirical risk minimization problem in Equation (2.5), an aggregated global model is optimized iteratively by a series of communications between a central server and a subset of clients where the local datasets reside. In each round, the server communicates the current state of the global model and the participating clients run a number of local optimization steps before communicating back to the server the updated model or the differential update. This approach has been shown to be under-performing when the local datasets are samples of non-congruent distributions, failing to minimize both the local and global objectives at the same time.

The need for personalized federated learning, therefore, emerged as a means to address this issue, with many different techniques being proposed. In [Mansour et al., 2020], the authors suggest three methods for personalization based on clustering, model interpolation, and data interpolation. The idea of hypothesis-based clustering is also studied in [Ghosh et al., 2020], which further provides convergence guarantees of the population loss function. The clustering of participating clients to give rise to a personalized model is also the approach taken in [Sattler et al., 2020], which goes on to introduce a meta-algorithm to determine whether the clients belong to non-congruent distributions, whether the federated optimization has reached minimums of both the clients' and server's objectives, and a method for clustering based on the cosine similarity of the updates.

In the works introduced above, the claims of privacy protection derive from the local raw data of the clients not being disclosed throughout the communication rounds between the server and the clients. As discussed in [Dwork et al., 2014a], disclosing any answer to a deterministic query can release private information and relying on the "release of summary statistics" argument (i.e. releasing only model updates instead of releasing clients' raw data) can have dramatic effects on the privacy of individuals.

To confront this issue, a number of works have focused on the privatization of the (federated) optimization algorithm under the framework of DP [Abadi et al., 2016; Geyer et al., 2017; McMahan et al., 2017b; Andrew et al., 2021], thus providing formal guarantees that the learned model will not depend too much on the presence or absence of a particular user's record in the dataset used in the federated optimization. The model of the attacker is thus reduced to an honest but curious adversary who only

has access to the trained model [Abadi et al., 2016; Andrew et al., 2021; McMahan et al., 2017b]. However, in this setting, no protection is ensured against the server and any possible man-in-the-middle attacker between the clients and the server who might access the clients' updates. This has been shown to be problematic as a malicious adversary with only access to the model updates sent by the clients has enough information to reconstruct samples from the local datasets [Zhu et al., 2019]. In [Bonawitz et al., 2016], the authors addressed this concern of communicating non-privatized updates to a central server by introducing a cryptographically secure aggregation protocol for the central server to compute the updated global model state from the encrypted client's updates, but at the cost of increased communication and computation requirements for both the clients and the server.

Since various kinds of communication constraints form some of the most defining characteristics of the FL setting, other works examined, instead, the use of local differential privacy mechanisms for protection against any strong adversary that may have access to the clients' updates [Truex et al., 2020; Zhao et al., 2020]. One such example is [Truex et al., 2020] which obfuscates each parameter within a certain adaptively-defined range of values and adopts a parameter shuffling mechanism to amplify the privacy guarantees being motivated by the shuffle model of DP [Bittau et al., 2017], which has been extensively studied of late in the literature [Sommer et al., 2019; Cheu et al., 2019; Cheu and Zhilyaev, 2021; Erlingsson et al., 2019, 2020; Balle et al., 2019, 2020; Meehan et al., 2021; Koskela et al., 2021b; Kairouz et al., 2016; Koskela et al., 2021a; Feldman et al., 2020]. It must be noted that the mechanism in [Truex et al., 2020] requires each parameter of the local model to be uploaded to the server one at a time, which can drastically increase the wall clock convergence time of the algorithm when used to train modern machine learning models which easily require millions of parameters.

In [Girgis et al., 2021] and [Erlingsson et al., 2020] the authors adopt the framework of local differential privacy and exploit shuffling, subsampling and other techniques to amplify the guarantees in terms of central differential privacy. Notably, these techniques still rely on a trusted aggregator. Work [Agarwal et al., 2018] examines quantization techniques used for improving communication efficiency to establish local differential privacy guarantees against an untrusted or negligent aggregator. Relatively to the works just mentioned, we highlight how the use of local differential privacy with non-trivial guarantees would be problematic with personalization, as, by definition, client updates belonging to the bounded domain

	Central Privacy	Local Privacy	Personalization	Mild Assumptions on Training	Fairness Analysis
[McMahan et al., 2017b]	✓	×	×	✓	×
[Hu et al., 2020]	✓	✓	✓	×	×
[Truex et al., 2020]	✓	✓	×	✓	×
This Chapter	✓	✓	✓	✓	✓

Table 2.2: Qualitative comparison with the most relevant prior research on the topic. More details are provided in Section 2.3.

of diameter $2 \cdot g_{\max}$ should be indistinguishable up to a small multiplicative factor. In [Hu et al., 2020] the authors address the problem of personalized and locally differentially private federated learning, but for the simple case of convex, 1-Lipschitz cost functions of the inputs. Note that this assumption is unrealistic in most machine learning models, and excludes many statistical modeling techniques, notably neural networks. Conversely, we do not make these assumptions. In Table 2.2 is provided a qualitative comparison of this effort compared with the most relevant prior work on the subject, in order to provide context of the problem and hand and its proposed solution.

2.4 An algorithm for private and personalized federated learning

The following section introduces our proposed algorithm for federated learning with local guarantees to provide group privacy (Algorithm 1). Locality refers to the sanitization of the information released by the client to the server, whereas group privacy refers to indistinguishability with respect to a neighborhood of clients defined with respect to a certain distance metric. Algorithm 1 is motivated from the Iterative Federated Clustering Algorithm (IFCA) [Ghosh et al., 2020] and builds on top of it to provide formal privacy guarantees. The main differences lie in the introduction of the `SanitizeUpdate` function described in Algorithm 2 and k-means for server-side clustering of the updated models.

The optimization strategy adopted here for the personalization of the federated models is discussed in the works of [Ghosh et al., 2020] and [Mansour et al., 2020] which converge to proposing similar algorithms independently. In summary, the intuition is to initialize a set of hypotheses for the parameter vectors, one for each potential cluster. In the t^{th} iteration, a subset of users receive the hypotheses, following which, each

participating user determines which one of them to optimize by evaluating which parameter vector yields the lowest cost over the local dataset. The assumption is that users with similar data distributions will adopt the same hypothesis. The updated models are then privatized before being returned to the server for averaging. The server is now tasked with deciding which models belong to the same cluster, in order to aggregate the corresponding parameter vectors. To do so, it performs k-means clustering starting from a specific choice of centroids, providing fast convergence. Estimating the clusters is effective under the assumption that the sanitized update to the model parameters $\hat{\xi}_c^{(t)}$ is relatively smaller than the difference between hypotheses at time t. With the notation described in Equations (2.1) through (2.5) and adopted in Algorithm 1, it means that $\forall\, j, i \in [k]\,,\, j = \bar{j},\, j \neq i,\, \forall\, c \in C^{(t)}$:

$$\hat{\xi}_c^{(t)} := \left\| \hat{\theta}_{\bar{j},c}^{(t)} - \theta_j^{(t)} \right\|_2 \ll \left\| \theta_i^{(t)} - \theta_j^{(t)} \right\|_2 \tag{2.12}$$

It is possible to see experimentally that these assumptions are mild and typically verified with machine learning models with a small number of parameters and a careful tuning of the Laplacian noise, although the optimal hypotheses depend of course on the (unknown) data distributions.

To introduce privacy guarantees in Algorithm 1, we deviate from the standard implementation of IFCA [Ghosh et al., 2020; Mansour et al., 2020] in the following ways:

1. We expect all the information leaving the users to be obfuscated locally before reaching the server.

2. Information about the number of samples a user trained the model on is not disclosed at all.

3. Users do not communicate the cluster membership to the server. This would be yet another information to sanitize, and we opt instead for letting the server evaluate membership based on the already privatized parameter vectors.

4. It follows that users cannot communicate $\hat{\xi}_c^{(t)}$ but the full sanitized and updated parameter vector $\hat{\theta}_{\bar{j},c}^{(t)}$. In other words, Algorithm 1 cannot rely on gradient averaging [Ghosh et al., 2020] and resorts to model averaging.

Algorithm 1 An algorithm for personalized federated learning with formal privacy guarantees in local neighborhoods.

Require: number of clusters k; initial hypotheses $\theta_j^{(0)}, j \in [k]$; number of rounds T; number of users per round U; number of local epochs E; local step size s; user batch size B_s; noise multiplier ν; local dataset Z_c held by user c.

for $t = \{0, 1, \ldots, T-1\}$ **do** ▷ Server-side loop
  $C^{(t)} \leftarrow$ SampleUserSubset(U)
  BroadcastParameterVectors($C^{(t)}; \theta_j^{(t)}, j \in [k]$)
  for $c \in C^{(t)}$ **do** in parallel ▷ Client-side loop
    $\bar{j} = \arg\min_{j \in [k]} F_c(\theta_j^{(t)}; Z_c)$
    $\theta_{\bar{j},c}^{(t)} \leftarrow$ LocalUpdate($\theta_{\bar{j}}^{(t)}; s; E; Z_c$)
    $\hat{\theta}_{\bar{j},c}^{(t)} \leftarrow$ SanitizeUpdate($\theta_{\bar{j},c}^{(t)}; \nu$)
  end for
  $\{S_1, \ldots, S_k\} =$ k-means($\hat{\theta}_{\bar{j},c}^{(t)}, c \in C^{(t)}; \theta_j^{(t)}, j \in [k]$)
  $\theta_j^{(t+1)} \leftarrow \frac{1}{|S_j|} \sum_{c \in S_j} \hat{\theta}_{\bar{j},c}^{(t)}, \quad \forall j \in [k]$
end for

Algorithm 2 SanitizeUpdate obfuscates a vector $\theta \in \mathbb{R}^n$, with a Laplacian noise tuned on the radius of a certain neighborhood and centered in 0.

function SANITIZEUPDATE($\theta_{\bar{j}}^{(t)}; \theta_{\bar{j},c}^{(t)}; \nu$)
  $\xi_c^{(t)} = \theta_{\bar{j},c}^{(t)} - \theta_{\bar{j}}^{(t)}$
  $\varepsilon = \frac{n}{\nu \|\xi_c^{(t)}\|}$
  Sample $\rho \sim \mathcal{L}_{0,\varepsilon}(x)$
  $\hat{\theta}_{\bar{j},c}^{(t)} = \theta_{\bar{j},c}^{(t)} + \rho$
  return $\hat{\theta}_{\bar{j},c}^{(t)}$
end function

2.4.1 The Laplace mechanism under Euclidean distance in $\mathbb{R}^n$

In Algorithm 2, `SanitizeUpdate` requires a careful consideration as it is the main privacy-preserving mechanism. All the server sees when a user communicates back is a parameter vector $\theta \in \mathbb{R}^n$. Without implementing the privacy mechanism, the true value from the user would be disclosed. Therefore, the following part of the section presents the motivation for and derivation of a particular flavor of the Laplace mechanism, and the heuristic used in `SanitizeUpdate` to define the neighborhood of a client.

Motivation

From the literature on geo-indistinguishability [Andrés et al., 2013], we extend the Laplace mechanism with Euclidean distance for any metric space $\mathbb{R}^n$ as described in Lemma 2. Note that there is no univocal definition of the multivariate Laplace distribution, and many different results can be considered generalizations of the univariate case. We resort to the Laplace mechanism under Euclidean distance because of the two following reasons:

i) Clustering is performed on θ with the k-means algorithm under Euclidean distance. Since we define clusters or groups of users based on how close their model parameters are under L_2 norm, we are looking for a d-privacy mechanism that obfuscates the reported values within a certain group and allows the server to differentiate among users belonging to different clusters.

ii) Consider an input-output relation of the kind $y = f(x, \theta)$ with f differentiable with respect to θ. Its parameter vector θ is to be estimated with Algorithm 1, such that it minimizes the Root Mean Square Error (RMSE) cost function

$$F_c = \sqrt{\frac{\sum_{i=1}^{|Z_c|}(y_i - f(x_i, \theta))^2}{|Z_c|}} = \frac{\|Y - f(X, \theta)\|_2}{\sqrt{|Z_c|}} \tag{2.13}$$

and $X = \left[x_1, \ldots, x_{|Z_c|}\right]^T$, $Y = \left[y_1, \ldots, y_{|Z_c|}\right]^T$, with $|Z_c|$ being the number of data points held by client c. If a client releases to the server its parameters θ_c sanitized by the addition of random vector Δ, we can evaluate how the cost function would change with respect

to the non-sanitized communication. Dropping the multiplicative constant we find:

$$\begin{aligned} \|Y - f(X,\theta_c)\|_2 - \|Y - f(X,\theta_c+\Delta)\|_2 \leq \\ \|Y - f(X,\theta_c) - Y + f(X,\theta_c+\Delta)\|_2 = \\ \|f(X,\theta_c+\Delta) - f(X,\theta_c)\|_2 \approx \\ \left\|f(X,\theta_c) + \nabla f(X,\theta_c)^T\Delta - f(x,\theta_c)\right\|_2 = \\ \left\|\nabla f(X,\theta_c)^T\Delta\right\|_2 \leq \\ \|\nabla f(X,\theta_c)\|_2 \|\Delta\|_2 \end{aligned} \tag{2.14}$$

Hence, we notice how we can bound such value proportionally to the Euclidean norm of the random noise. Notably, it does not depend on the direction of Δ. Thus, we require that points with the same bound on the increase of the cost function (which are all points distant $\|\Delta\|_2$ from θ_c) will be sampled with the same probability.

Derivation

Lemma 2. *Let $\mathcal{L}_\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ be the Laplace mechanism of the form $\mathcal{L}_{x_0,\varepsilon}(x) = \Pr[\mathcal{L}_\varepsilon(x_0) = x] = Ke^{-\varepsilon d(x,x_0)}$ with $d(x,x_0) = \|x - x_0\|_2$. The mechanism is ε d-private and*

$$K = \frac{\varepsilon^n \Gamma(\frac{n}{2})}{2\pi^{\frac{n}{2}}\Gamma(n)} \tag{2.15}$$

Proof. If $\mathcal{L}_{x_0,\varepsilon}(x) = Ke^{-\varepsilon d(x,x_0)}$ is a probability density function of a point in $\mathbb{R}^n$ then there is a K such that $\int_{\mathbb{R}^n} \mathcal{L}_{x_0}(x)dx = 1$. We note that it depends only on the distance from x_0 and we can write $Ke^{-\varepsilon d(x,x_0)} = Ke^{-\varepsilon r}$ where r is the radius of the ball in $\mathbb{R}^n$ centered in x_0. Without loss of generality, let us now take $x_0 = 0$. The probability density of the event $x \in \mathbb{S}_n(r) = \{x : \|x\|_2 = r\}$ is then $p(x \in \mathbb{S}_n(r)) = Ke^{-\varepsilon r}\mathbb{S}_n(1)r^{n-1}$ where $S_n(1)$ is the surface of the unitary ball in $\mathbb{R}^n$ and $S_n(r) = S_n(1)r^{n-1}$ is the surface of a generic ball of radius r. Given that

$$S_n(1) = \frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})} \tag{2.16}$$

solving

$$\begin{aligned} \int_0^{+\infty} \Pr[x \in \mathbb{S}_n(r)]\, dr = \int_0^{+\infty} Ke^{-\varepsilon r} S_n(1) r^{n-1} dr = \\ = K\frac{2\pi^{n/2}\Gamma(n)}{\varepsilon^n \Gamma(\frac{n}{2})} = 1 \end{aligned} \tag{2.17}$$

results in

$$K = \frac{\varepsilon^n \Gamma(\frac{n}{2})}{2\pi^{\frac{n}{2}}\Gamma(n)} \tag{2.18}$$

where $\Gamma(\cdot)$ denotes the gamma function. By plugging $\mathcal{L}_{x_0,\varepsilon}(x) = Ke^{-\varepsilon d(x,x_0)}$ in Equation 2.7:

$$Ke^{-\varepsilon d(x,x_1)} \leq e^{\varepsilon d(x_1,x_2)} Ke^{-\varepsilon d(x,x_2)} \tag{2.19}$$

$$e^{\varepsilon(\|x-x_2\|_2 - \|x-x_1\|_2)} \leq e^{\varepsilon\|x_1-x_2\|} = e^{\varepsilon d(x_1,x_2)} \tag{2.20}$$

□

One of the biggest advantages of d-privacy is that the level of privacy can be derived for a repeated number of independent queries since it satisfies the compositionality theorem [Dwork et al., 2014a], which is one of the key properties for the applicability of DP and its variants for formalizing the privacy guarantee for a composition of independent queries.

Theorem 3. *[Compositionality Theorem for d-privacy] Let $\mathcal{K}_i$ be (ε_i)-d-private mechanism for $i \in \{1,2\}$. Then their independent composition is $(\varepsilon_1+\varepsilon_2)$-$d$-private, i.e., for every $S_1,\ S_2 \subseteq \mathcal{Y}$ and all $x_1,\ x_1',\ x_2,\ x_2' \in \mathcal{X}$, we have:*

$$\begin{aligned} &\Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1,x_2)] \\ \leq e^{\varepsilon_1 d(x_1,x_1') + \varepsilon_2 d(x_2,x_2')} &\Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1',x_2')] \end{aligned} \tag{2.21}$$

Proof. Let us simplify the notation and denote:

$$P_i = \Pr_{\mathcal{K}_i}[y_i \in S_i | x_i]$$

$$P_i' = \Pr_{\mathcal{K}_i}[y_i \in S_i | x_i']$$

for $i \in \{1,2\}$. As mechanisms $\mathcal{K}_1$ and $\mathcal{K}_2$ are applied independently, we have:

$$\Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1,x_2)] = P_1.P_2$$

$$\Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1',x_2')] = P_1'.P_2'$$

Therefore, we obtain:

$$\begin{aligned} \Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1,x_2)] &= P_1.P_2 \\ &\leq \left(e^{\varepsilon_1 d(x_1,x_1')} P_1'\right)\left(e^{\varepsilon_2 d(x_2,x_2')} P_2'\right) \\ \leq e^{\varepsilon_1 d(x_1,x_1') + \varepsilon_2 d(x_2,x_2')} &\Pr_{\mathcal{K}_1,\mathcal{K}_2}[(y_1,y_2) \in S_1 \times S_2 | (x_1',x_2')] \end{aligned}$$

□

A heuristic for defining the neighborhood of a client

In the t^{th} iteration, when a user c calls the `SanitizeUpdate` routine in Algorithm 2, it has already received a set of hypotheses, optimized $\theta_{\bar{j}}^{(t)}$ (the one that fits best its data distribution), and got $\theta_{\bar{j},c}^{(t)}$. It is reasonable to assume that clients whose datasets are sampled from the same underlying data distribution $\mathcal{P}_{\bar{j}}$ (as described in Section 2.2.1) will perform an update similar to $\xi_c^{(t)}$.

Definition 8. *For any model parametrized by $\theta \in \mathbb{R}^n$, we define its r-*neighborhood *as the set of points in the parameter space which are at a L_2 distance of at most r from θ, i.e., $\{\phi \in \mathbb{R}^n \colon \|\theta, \phi\|_2 \leq r\}$*

Definition 9. *Clients whose models are parametrized by $\theta \in \mathbb{R}^n$ in the same r-neighborhood are said to be in the same* group*, or* cluster.

Therefore, we require that points which are within the $\xi_c^{(t)}$-neighborhood of $\hat{\theta}_{\bar{j},c}^{(t)}$ to be indistinguishable. To provide this guarantee, we tune the Laplace mechanism such that the points within the neighborhood are $\varepsilon\|\xi_c^{(t)}\|_2$ differentially private. With the choice of $\varepsilon = n/(\nu\xi_c^{(t)})$, one finds that $\varepsilon\|\xi_c^{(t)}\|_2 = n/\nu$, and we call ν the *noise multiplier*. It is straightforward to observe that the larger the value of ν gets, the stronger the privacy guarantee. Note that in order to derive this result, we exploited the fact that the norm of the noise vector sampled from Laplace distribution is distributed according to Equation (2.22) and its expected value is $\mathbb{E}\left[\gamma_{\varepsilon,n}(r)\right] = n/\varepsilon$.

2.4.2 Sampling from the Laplace mechanism

Exploiting the radial symmetry of the Laplace distribution, we note that, in order to sample a point $x_s \sim \mathcal{L}_0(x)$ in $\mathbb{R}^n$, it is possible to first sample the set of points distant $d(x, 0) = r$ from $x_0 = 0$ and then sample uniformly from the resulting hypersphere. Accordingly, the p.d.f. of the event $x \in \mathbb{S}_n(r) = \{x : \|x\|_2 = r\}$ is then $\Pr\left[x \in \mathbb{S}_n(r)\right] = Ke^{-\varepsilon r}\mathbb{S}_n(1)r^{n-1}$, where K is as in Lemma 2 and $\mathbb{S}_n(r)$ is the surface of the sphere with radius r in $\mathbb{R}^n$. Hence, we can write

$$\gamma_{\varepsilon,n}(r) = \frac{\varepsilon^n e^{-\varepsilon r} r^{n-1}}{\Gamma(n)} \tag{2.22}$$

which is the gamma distribution with shape n and scale $1/\varepsilon$. Drawing from $\gamma_{\varepsilon,n}(r)$ is implemented in multiple routines in common programming languages. Equation (2.22) represents the p.d.f. of sampling the hypersphere

of radius $\|x_s\| = r \sim \gamma_{\varepsilon,n}(r)$. To sample a point uniformly from the corresponding hypersphere one can sample ${}_1x_n \in \mathbb{S}_n(1)$, a point from the hypersphere of radius 1, and have that $x_s = {}_1x_n \|x_s\|$, where ${}_1x_n = \frac{x_n}{\|x_n\|}$. This can be done operationally by sampling x_n from the n-dimensional vector whose components are sampled from a Gaussian distribution centered at 0 and with a variance σ^2, i.e., $x_n \sim \mathcal{N}_n(0, \sigma^2)$ and letting ${}_1x_n = \frac{x_n}{\|x_n\|}$.

2.4.3 Component-wise variance

To better characterize the distribution in Lemma 2, we now proceed to show how to derive the variance of each single component x_i of $x = [x_1, \dots, x_n]^T$.

Lemma 4. *Let $x \sim \mathcal{L}_{0,\varepsilon}$, $x \in \mathbb{R}^n$ as in Lemma 2 and $r \sim \gamma_{\varepsilon,n}$ as in Equation* (2.22)*, then we have that the variance of the i-th component of x is $\sigma^2_{x_i} = \frac{n+1}{\varepsilon^2}$.*

Proof. With $r \sim \gamma_{\varepsilon,n}$ we have that, by construction,

$$\mathbb{E}\left[r^2\right] = \mathbb{E}\left[\sum_{i=1}^{n} x_i^2\right] = n\mathbb{E}\left[x_i^2\right] = n\sigma^2_{x_i} \tag{2.23}$$

With the last equality holding since $\mathcal{L}_{0,\varepsilon}$ is isotropic and centered in zero. Recalling that

$$\mathbb{E}\left[r^2\right] = \frac{d^2}{dt^2} M_r(t)\Big|_{t=0} \tag{2.24}$$

with $M_r(t)$ the moment generating function of the gamma distribution $\gamma_{\varepsilon,n}$,

$$\begin{aligned}
&\frac{d^2}{dt^2}\left(\left(1 - \frac{t}{\varepsilon}\right)^{-n}\right)\Bigg|_{t=0} = \\
&= \frac{n(n+1)}{\varepsilon^2}\left(1 - \frac{t}{\varepsilon}\right)^{-(n+2)}\Bigg|_{t=0} = \\
&= \frac{n(n+1)}{\varepsilon^2}
\end{aligned}$$

which leads to

$$\sigma^2_{x_i} = \frac{n+1}{\varepsilon^2} \tag{2.25}$$

□

2.4.4 Limitations of the Laplace mechanism in very high dimensional spaces

As already described in Section 2.2.3 and 2.4.1, d-privacy provides differential privacy guarantees to a point $x_0 \in \mathcal{X}$, with privacy parameter at most εr, with respect to any point x, such that $d(x, x_0) \leq r$. These local differential/d-privacy guarantees for federated learning models are a desirable feature which would make any information disclosure from the client to the server indistinguishable up to a certain multiplicative factor. Local DP mechanisms ensure also central DP, and thus would provide its guarantees as well. However, LDP is notoriously hard to achieve while maintaining the utility of the queries. In [Bassily et al., 2017] are evaluated the lower bounds of the error on the estimate of a counting query under both local and central DP with the Laplace mechanism. They are found to be $O(1/\varepsilon)$ and $\Omega(\sqrt{N}/\varepsilon)$ respectively, which for the latter depend on the number of participating individuals N. In the context of federated learning though, where individual information is aggregated e.g. by average, the Central Limit Theorem would yield a reduction of the standard deviation of the aggregate error by $\sqrt{N}$ in the local model. Instead, we want to highlight what we consider to be the hardest obstacle in providing LDP guarantees in federated learning.

Assume that we want to sanitize information locally with the Laplace mechanism defined in Lemma 2. With the results found in Section 2.4.2 we see that each point $x \in \mathbb{R}^n$ would be sanitized by the addition of a vector ρ whose norm is distributed as $\|\rho\|_2 \sim \gamma_{\varepsilon,n}(r)$. Its mean is found to be $\mathbb{E}\left[\gamma_{\varepsilon,n}(r)\right] = n/\varepsilon$, and we highlight the linear dependency on n. In large machine learning models where the number of parameters easily reach a few million, this would completely destroy utility, as maintaining LDP with small ε values would require noise levels that dwarf the true values of the parameters. Indeed, in Section 2.5.1 we conduct experiments on model architecture leading to $\theta \in \mathbb{R}^{1206590}$, and we can see that maintaining low levels of the LDP parameters would destroy the model's accuracy. Conversely, maintaining high utility would yield huge values of LDP parameters, rendering formal LDP guarantees practically meaningless.

However, in the case of federated machine learning, the typical white-box attack is the Deep Leakage from Gradients (DLG) [Zhu et al., 2019]. In our experiments, we have empirically verified that we can achieve a strong defense against this kind of attacker while maintaining a good level of accuracy.

2.5 Experiments

All the following experiments are run on a local server running Ubuntu 20.04.3 LTS with an AMD EPYC 7282 16-Core processor, 1.5TB of RAM, and $8\times$ NVIDIA A100 GPUs. Python and PyTorch are the main software tools adopted for simulating the federation of clients and their corresponding collaborative training. These experiments are structured to first evaluate the utility-privacy trade-off, and then to connect the research with the existing fairness metrics already discussed in the past sections.

2.5.1 Characterizing privacy

Synthetic data

The first experiment tests Algorithm 1 on synthetic data generated from a linear mapping with a set of predetermined optimal parameters. In particular, we generate data according to $k = 2$ different distributions

$$y = x^T\theta_1^* + u; \quad u \sim \text{Uniform}\,[0, 1) \tag{2.26}$$

$$y = x^T\theta_2^* + u; \quad u \sim \text{Uniform}\,[0, 1) \tag{2.27}$$

with $\theta_1^* = [+5, +6]^T, \theta_2^* = [+4, -4.5]^T$. A total of 100 users holds 10 samples each, drawn from either one of the distributions. They participate in a training of two initial hypotheses which are sampled from a Gaussian distribution centered in 0 and unit variance at iteration $t = 0$. A total of $U = 7$ users are asked to participate in the optimization at each round and train locally the hypothesis that fits better their dataset for $E = 1$ epochs each time. The noise multiplier is set to $\nu = 5$. Local step size $s = 0.1$ and a batch size $B_s = 10$ complete the required inputs to the algorithm. To verify the training process, another set of users with the same characteristics are held out from training to perform validation and stop the federated optimization once the is no improvement in the loss function in Equation (2.13) for 6 consecutive rounds. Results of the training process are shown in Figures 2.1g, 2.1h, 2.1i. Note that the real clients' parameters would not be visible to the server but are drawn on the plots for clarity. Although at first the updates seem to be distributed all over the domain, in just a few rounds of training the process converges to values very close to the two optimal parameters. With the heuristic presented in Section 2.4.1 it is easy to find that whenever a user participates in an optimization round it incurs a privacy leakage of at most $n/\nu = 2/5 = 0.4$, in a differential private

sense, with respect to points in its neighborhood. Using the result in Theorem 3 clients can compute the overall privacy leakage of the optimization process, should they be required to participate multiple times. With the uniform sampling of the clients (without replacement) that was used in this experiment, the maximum composed value of the privacy leakage was 2.4. For any user, whether to participate or not in a training round can be decided right before releasing the updated parameters, in case that would increase the privacy leakage above a threshold value decided beforehand.

In a concise ablation study, we assess how training progresses when two characteristic features of Algorithm 1 are removed:

- the privatization of the client parameters
- model personalization

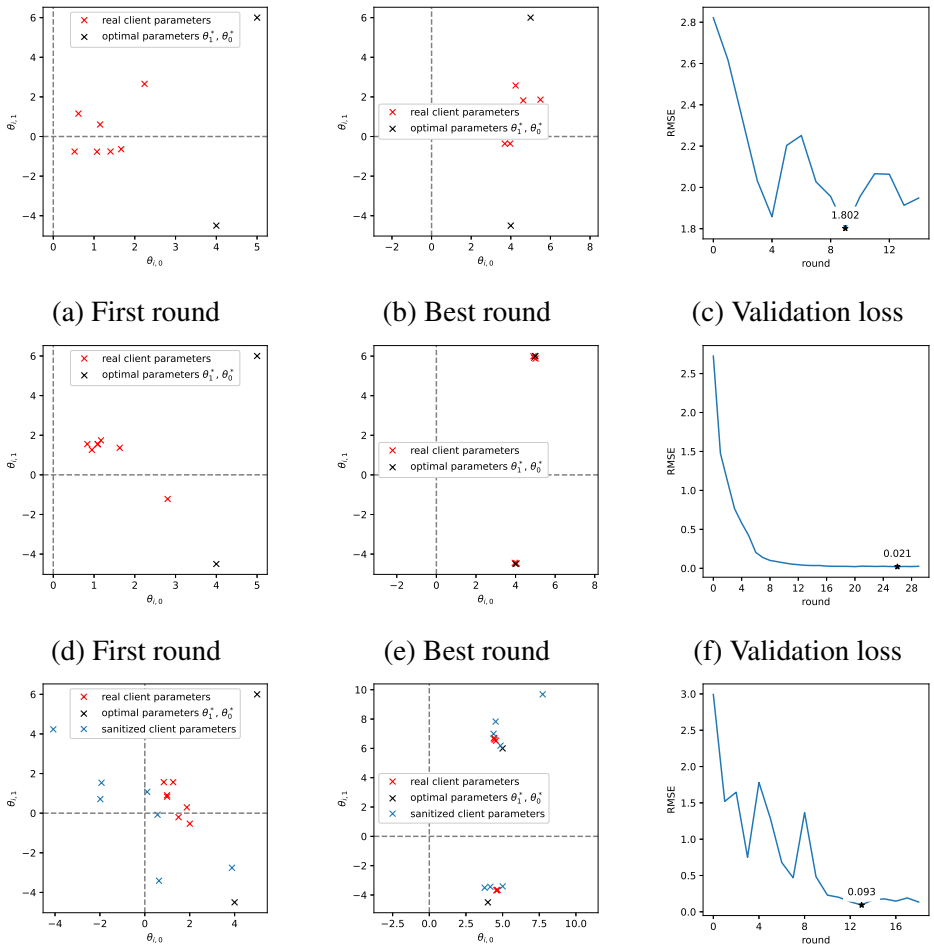
In Figure 2.1d, 2.1e, 2.1f no sanitization is performed on the updated parameters sent by the users and the optimization terminates with the clients very close to the optimal parameters. This is reflected in the validation loss reaching the lowest value among the three cases. We highlight, though, how it is still in the same order of magnitude as the sanitized case.

In Figure 2.1a, 2.1b, 2.1f the clients are left to optimize the initial hypotheses without personalization, and we find that the validation loss is considerably larger than both the non-sanitized and sanitized case. This is evident also as the real client parameters transmitted to the server converge somewhere in between the optimal parameters. Further, in Figure 2.2 is provided the increase in the maximum value of privacy leakage clients incur, per cluster.

Hospital charge data

This experiment is performed on real-world data, specifically, the Hospital Charge Dataset published by the Centers for Medicare and Medicaid Services of the US Government. It contains data about charges for the 100 most common inpatient services and the 30 most common outpatient services. It shows a great variety of charges applied by healthcare providers with details mostly related to the type of service and the location of the provider. Preprocessing of the dataset includes a number of procedures, the most important of which are described here:

i) Selection of the 4 most widely treated conditions, which amount to simple pneumonia; kidney and urinary tract infections; heart failure and shock; esophagitis and digestive system disorders.



(a) First round (b) Best round (c) Validation loss

(d) First round (e) Best round (f) Validation loss

(g) First round (h) Best round (i) Validation loss

Figure 2.1: Learning federated linear models with: (a, b, c) one initial hypothesis and non-sanitized communication, (d, e, f) two initial hypotheses and non-sanitized communication, (g, h, i) two initial hypotheses and sanitized communication. The first two figures of each row show the parameter vectors released by the clients to the server. The last figure of each row illustrates the trend of the validation loss on clients and data not involved in the optimization.

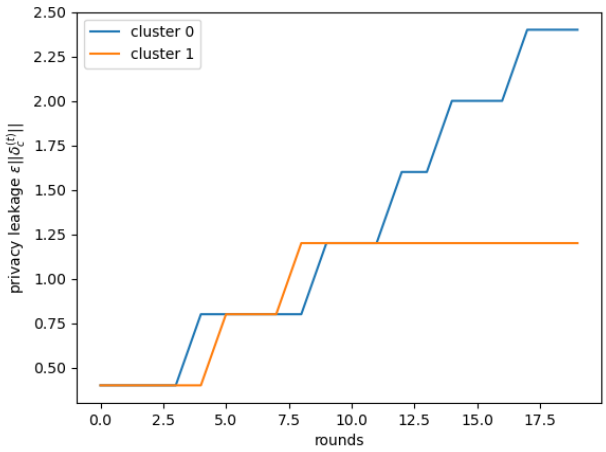


Figure 2.2: For the experiment on synthetic data, this figure plots the max privacy leakage over clients of the same cluster for a round of training. Intervals with constant privacy leakage indicate that the clients with the largest privacy leakage were not sampled (by chance) to participate in those rounds.

ii) Transformation of ZIP codes into numerical coordinates in terms of longitude and latitude.

iii) Setting as target the Average Total Payments, i.e. the cost of the service averaged among the times it was given by a certain provider.

iv) As it is a standard procedure in the context of gradient-based optimization, dependent and independent variables are brought to be in the range of the *units* before being fed to the machine learning model. Note that this point takes the spot of the common feature normalization and standardization procedures, which we decided not to perform here to keep the setting as realistic as possible. In fact, both would require the knowledge of the empirical distribution of all the data. Although it is available in simulation, it would not be available in a real scenario, as each user would only have access to their dataset.

To simulate a federated learning process, healthcare providers are here considered the set of clients willing to collaborate to train a machine learning model. Given the preprocessing described above, the dataset results in 2947 clients randomly split into train and validation subsets with 70 and 30

percent of the total clients each. The goal is to be able to predict the cost that a service would require given where it is performed in the country, and what kind of procedure it is. The model that was adopted in this context is a fully connected neural network (NN) of two layers, with a total of 11 parameters and a Rectified Linear Unit (ReLU) activation function. Inputs to the model are an increasing index that uniquely defines the healthcare service and the longitude and latitude of the provider. The output of the model is the expected cost. Tests have been performed to minimize the RMSE loss on the clients selected for training (100 per round) and at each round, the performance of the model is checked against a held-out set of validation clients, from where 200 are sampled every time. If 30 validation rounds are passed without improvement in the cost function, the optimization process is terminated. To assess the trade-off between privacy, personalization, and accuracy, a different number of initial hypotheses has been checked, as it is not known *a priori* how many distributions generated the data. For the same reason, accuracy has been checked at different values of the noise multiplier ν. Further, in order to decrease the variability of the results, a total of 10 runs have been performed with different seeds for every combination of the number of hypotheses and noise multiplier. Results are shown in Figure 2.4.

When the federated training is performed with only 1 initial hypothesis, the accuracy of the model is poor, which is indicative of the model not being able to capture the variety of data distributions that is being fed with. In fact, increasing to 3 the number of initial hypotheses for the parameter vector leads to the biggest improvement on the RMSE loss. Additionally, we can see that the model's performance degrades with increasing values of the noise multiplier (and therefore increasing ε's), as expected. The large variability in performance when the communication is sanitized with $\nu \in \{2, 3, 5\}$ may be due to the assumption in Equation (2.12) failing to be satisfied in certain runs, leading to all clients being grouped under a single cluster, and reaching RMSE comparable to that obtained with only 1 initial hypothesis. The best results in terms of both accuracy and low variability are when the number of initial hypotheses is set to 5 and 7. Although a prescriptive characterization of the decrease in the model's performance with varying noise multiplier levels is yet to be derived, we highlight how experimentally there are regions of the hyper-parameter space (i.e. the choice of ν and the number of initial hypotheses) where a reasonable compromise can be found between privacy and model personalization.

Finding the privacy leakage is straight-forward, as each time a user is required to participate in a training round it will enjoy $\varepsilon\|\xi_c^{(t)}\|_2 = n/\nu =$

	Hypotheses			
Noise Multiplier	7	5	3	1
0	-, -	-, -	-, -	-, -
0.100	517.0, 1551.0	418.0, 1342.0	473.0, 1386.0	528.0, 1540.0
1	36.3, 126.5	40.7, 127.6	44.0, 138.6	49.5, 147.4
2	15.4, 57.8	14.3, 54.5	22.0, 69.3	21.5, 66.6
3	7.7, 32.3	8.4, 36.7	12.5, 40.0	12.1, 40.0
5	5.7, 21.3	5.9, 22.0	5.5, 21.6	5.3, 20.9

Table 2.3: Regarding the experiment on hospital charge data, for every combination of Noise Multiplier $\times$ Number of Hypotheses, the median and maximum local privacy budgets are reported, over the whole set of clients. These values are averaged over 10 runs with different seeds. $\nu = 0$ means no privacy guarantee and infinite privacy leakage.

$11/\nu$ differential privacy with any point in its $\xi_c^{(t)}$-neighborhood. Accordingly, Figure 2.3 provides the empirical privacy leakage distribution of the clients involved in a particular training configuration, whereas Table 2.3 shows privacy leakage statics over multiple rounds and for all configurations.

FEMNIST image classification

In this Section we evaluate how Algorithm 1 behaves when tested beyond the scope of its applicability, as described in Section 2.4.4. The task consists of performing image classification on the FEMNIST [Caldas et al., 2018] dataset, which is a standard benchmark dataset for federated learning, based on EMNIST [Cohen et al., 2017] and with the data points grouped by user. It consists of a large number of images of handwritten digits, lower and upper case letters of the Latin alphabet. As a pre-processing step, images of client c are rotated 90 degrees counterclockwise depending on the realization of the random variable $\text{rot}_c \sim \text{Bernoulli}(0.5)$. This is a common practice in machine learning to simulate local datasets held by different clients being generated by very different distributions [Ghosh et al., 2020; Goodfellow et al., 2013; Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017].

The chosen architecture is described in Table 2.5 and yields a parameter vector $\theta \in \mathbb{R}^{n_0}$, $n_0 = 1206590$. Runs are performed with a maximum of 500 rounds of federated optimization unless 5 consecutive validation rounds are conducted without improvements on the validation loss. The

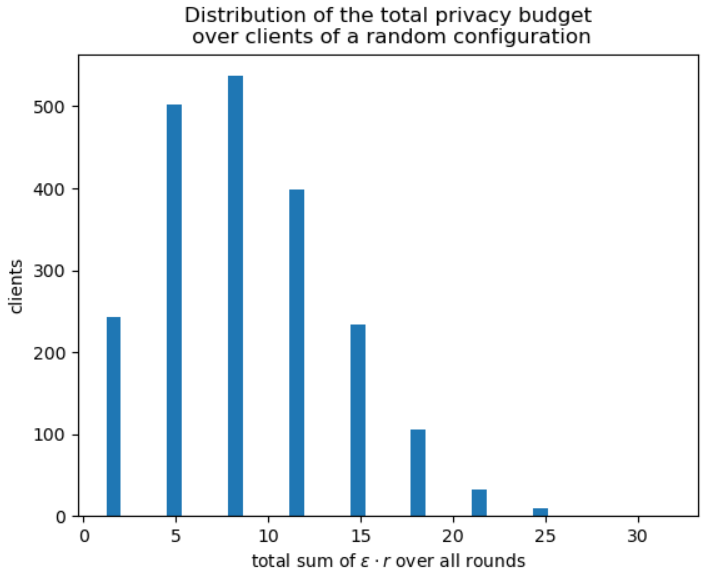


Figure 2.3: For the experiment on hospital charge data, this histogram plots the empirical distribution of the privacy budget over the clients in a particular configuration: $\nu = 3$, 5 initial hypotheses, seed $= 3$, r is the radius of the neighborhood, and the total number of clients is 2062.

latter is evaluated on a held-out set of clients, consisting of 10% of the total number. Validation is performed every 5 training rounds, thus the process terminates after 25 rounds without the model's performance improvement. The optimization process aims to minimize either the RMSE loss or the Cross Entropy loss [Zhang and Sabuncu, 2018] (to further depart from earlier assumptions) between the model's predictions and the target class. Results are presented in Table 2.4. For Cross Entropy, we see a wide range of ν values with comparable average accuracy. In particular, the best-performing model is being trained with a non-zero noise multiplier, which may be explained by a regularizing effect of the additive noise. This is especially true for the RMSE loss, where the best-performing model is trained with $\nu = 3$. For all the runs, we highlight a generally low standard deviation in the results.

Note that with the choice of the range of noise multipliers ν the corresponding value for the privacy leakage $\varepsilon\|\xi_c^{(t)}\|_2 = n/\nu = n_0/\nu$ would be enormous, and would not provide any meaningful guarantee, in theory. As already mentioned in Section 2.4.4, that is true as long as we want to use the Laplace mechanism to be effective against *any* adversary. Still, it is possible to validate, in practice, whether it can protect against a *specific*

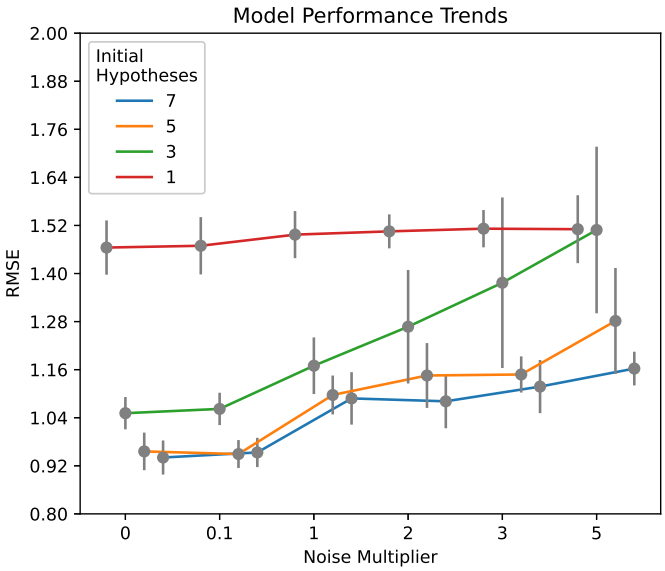


Figure 2.4: RMSE values for models trained with Algorithm 1 on the Hospital Charge Dataset. Error bars show the empirical standard deviation. Lower RMSE values are better for accuracy.

attack: DLG [Zhu et al., 2019]. The threat model for this attack is very fitting for a federated learning scenario. In brief: An honest-but-curious server communicates to a set of clients the parameter vector $\theta_{\bar{j}}^{(t)}$ (among the other $k-1$ hypotheses) at iteration t and receives the updated model parameters $\theta_{\bar{j},c}^{(t)}$ from client c. The server can easily retrieve the true parameter update $\xi_c^{(t)} = \theta_{\bar{j},c}^{(t)} - \theta_{\bar{j}}^{(t)}$ if no sanitization is performed. Under the assumption that the client performs one single optimization step, this results in the gradient scaled down by the local step size. The server then tries to recreate the input samples that generated such a gradient. The process of gradient matching can be cast into a nonlinear minimization problem and be solved itself by gradient descent.

If sanitization is performed, the server is left with matching a corrupted gradient. In [Zhu et al., 2019] the authors evaluate disturbing the gradient with Gaussian and Laplace (with L_1 distance) noise as a privacy mechanism. In the following, we evaluate if the distribution of the Laplace mechanism (under L_2 distance) in Lemma 2 is effective in protecting from the DLG attack. In order to be on the safe side, tests were conducted with the best possible conditions for the attacker: a modified model architecture, so that DLG conditions are met (e.g. all activation functions are replaced with

the sigmoid non-linearity to have a twice-differentiable model); batch size reduced to 1, as the gradient matching optimization problem is easier to solve in this setting; and a single local optimization step. Since the gradient can vary widely for parameters in different layers of the neural network, we apply the Laplace mechanism independently on the parameter vector of each NN layer and communicate to the server the collection of sanitized parameter vectors. The practice of sanitizing each layer independently has already been effectively evaluated in [Liu et al., 2020].

In Figure 2.5 results are reported for application of the noise multiplier values adopted also in Table 2.4. When $\nu = 10^{-3}$ the ground truth image is fully reconstructed. Up to $\nu = 10^{-1}$ we see that at least partial reconstruction is possible. Finally, for $\nu \geq 1$ we see that, experimentally, the DLG attack fails to reconstruct input samples.

	Cross Entropy loss		RMSE loss	
Noise Multiplier	Average Accuracy	Standard Deviation	Average Accuracy	Standard Deviation
0	0.832	$\pm$ 0.012	0.801	$\pm$ 0.001
0.001	0.843	$\pm$ 0.006	0.813	$\pm$ 0.014
0.01	0.832	$\pm$ 0.017	0.805	$\pm$ 0.008
0.1	0.834	$\pm$ 0.026	0.808	$\pm$ 0.019
1	0.834	$\pm$ 0.014	0.814	$\pm$ 0.012
3	0.835	$\pm$ 0.017	0.825	$\pm$ 0.010
5	0.812	$\pm$ 0.016	0.787	$\pm$ 0.003
10	0.692	$\pm$ 0.002	0.687	$\pm$ 0.014
15	0.561	$\pm$ 0.005	0.622	$\pm$ 0.003

Table 2.4: Average classification accuracy and standard deviation of a convolutional neural network over three runs seeded with different values. Experiments tested the effect of increasing noise values on the validation accuracy.

2.5.2 Fairness analysis

In this section, we analyze how group fairness improves with the personalization of the trained models under d-privacy guarantees when there are two groups with different data distributions. Experiments were performed on synthetic data and the FEMNIST image classification dataset that was used in the last Section. To ensure a thorough evaluation, we considered a variety of group fairness metrics in the experiments. In particular, we mea-

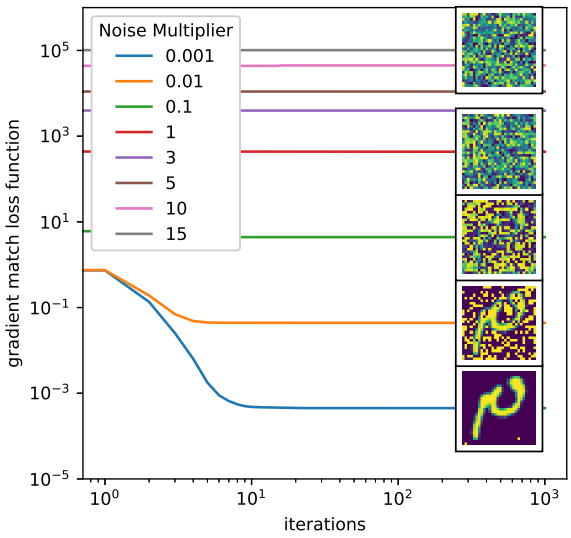


Figure 2.5: Effects of the Laplace mechanism in Lemma 2 with different noise multipliers (ref) as a defense strategy against the DLG attack.

Layer	Properties
2D Convolution	kernel size: (2,2) stride: (1,1) nonlinearity: ReLU output features: 32
2D Convolution	kernel size: (2,2) stride: (1,1) nonlinearity: ReLU output features: 64
2D Max Pool	kernel size: (2,2) stride: (2,2) nonlinearity: ReLU
Fully Connected	nonlinearity: ReLU units: 128
Fully Connected	nonlinearity: ReLU units: 62

Table 2.5: NN architecture adopted in the experiments of Section 2.5.1

sured the fairness with respect to equal opportunity, equalized odds [Hardt et al., 2016], and demographic parity [Dwork et al., 2012] as explained in Section 2.2.4.

In particular, in Figures 2.7 and 2.8, the X-axis denotes the noise multiplier ν representing the amount of d-private noise added to the local updates as explained in Section 2.4.1 and the Y-axis denotes the absolute value of the difference in fairness between the privileged and unprivileged groups with respect to the different metrics of group fairness that we considered.

Synthetic data

Synthetic data was generated in a method similar to that in Section 2.5.1 with the following modifications to enable ourselves to investigate the aspect of group fairness fostered by our method:

- Total number of users is 1000 and each user holds 10 samples. 800 users have data that is generated according to distributions $y = x^T\theta_1 + u$ and $u \sim$ Uniform $[0, 1)$, $i \in \{1, 2\}$, and set as a privileged majority group g_1. The remaining 200 users have data that is generated according to distribution $y = x^T\theta_2 + 15 + u$ and $u \sim$ Uniform $[0, 1)$, $i \in \{1, 2\}$, and set as an unprivileged minority group g_2. In this case, the sensitive attribute considered to evaluate fairness is the group id G where $G \in \{g_1, g_2\}$.

- For binary classification, we set labels by using the $z = \mathrm{Sigmoid}(Y)$, $y, \hat{y} \in Y$. In the case of g_1, we assign the label 1 if the value of z is greater than or equal to 0.5 and assign the label 0 otherwise. On the other hand, in the case of g_2, the label 1 is assigned when the $z = \mathrm{Sigmoid}(Y - 15)$, $\forall\, y, \hat{y} \in Y$ is less than or equal to 0.5, and the label 0 is assigned otherwise. This setting simulates a scenario where discrimination based on sensitive attributes occurs, similar to real-world situations where one group might experience higher loan rejection rates than another group with equivalent qualifications [Bartlett et al., 2022]. Thus, in our experiment, label 1 could be interpreted as "loan approved" and label 0 as "loan denied". The data generated in this way are shown in Figure 2.6.

We compared the fairness for two cases: one with a single hypothesis (without personalization) and the other with 2 of hypotheses as 2 (with personalization) in the framework of Algorithm 1. The results illustrated by Figure 2.7 assert that the personalization of models (i.e., Algorithm 1)

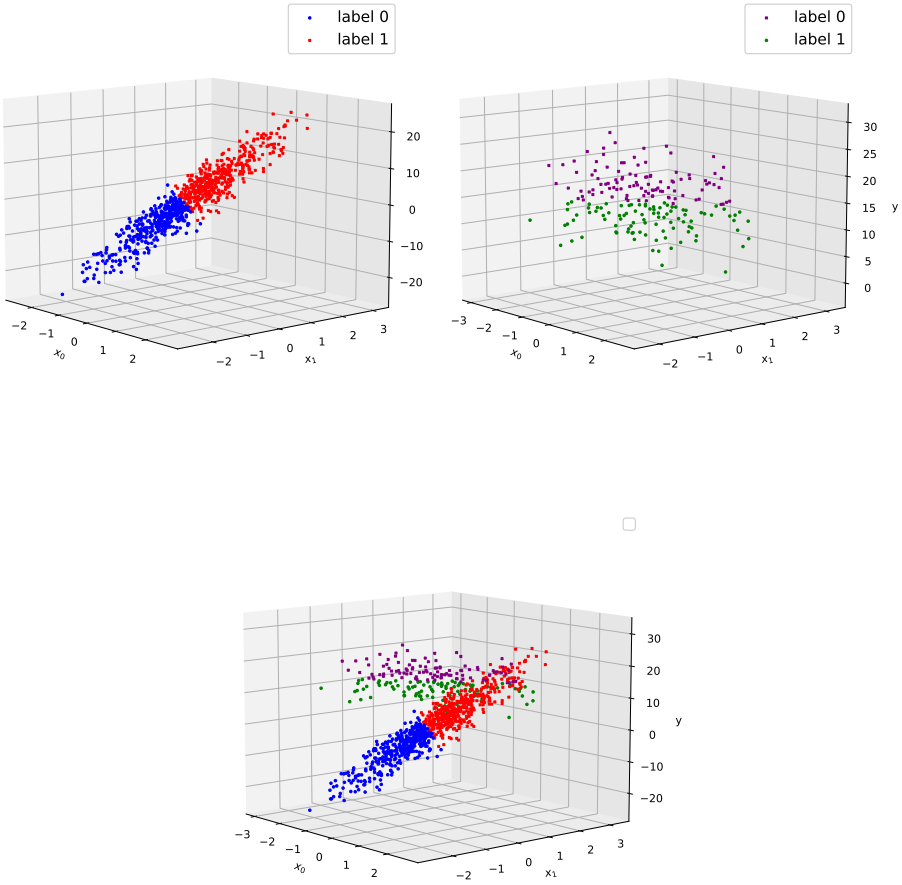
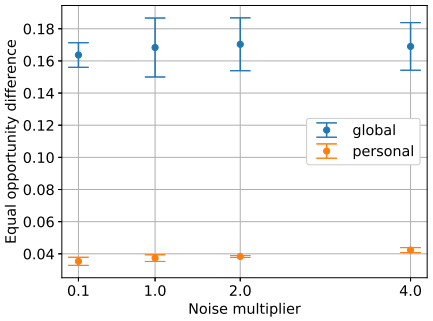
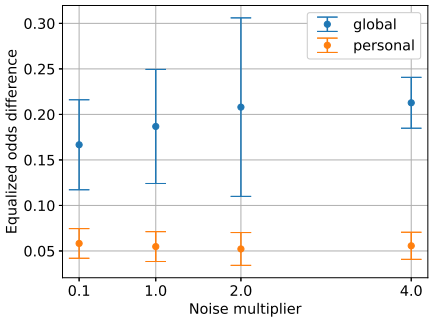


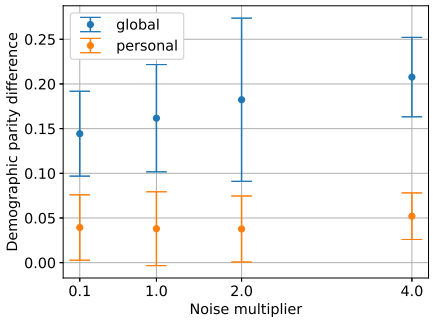
Figure 2.6: The first two plots from the left illustrate the spatial distribution of the samples in g_1 and g_2, respectively, and the third plot shows g_1 and g_2 superimposed together in the same space.



(a) Equal opportunity difference



(b) Equalized odds difference



(c) Demographic parity difference

Figure 2.7: For the experiment with synthetic data, the figure shows the comparison between the personalized and non-personalized models for equal opportunity (a), equalized odds (b), and demographic parity (c), respectively. Experiments were performed for noise multipliers ν of 0.1, 1, 2, and 4. For all the metrics of fairness and the values of the noise multiplier, the personalized model is seen to show improved fairness over the non-personalized model.

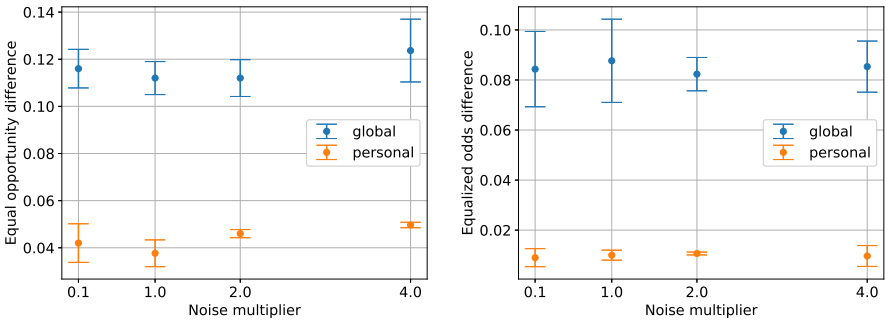
enhances the group fairness under all the metrics and all the levels of formal privacy guarantees, compared to that of the non-personalized model. A major reason behind this significant improvement of fairness by the personalized model is that unlike the non-personalized model, which trains using data from both groups that are biased towards the majority group g_1, the personalized model training optimizes for each group's data distribution without disregarding the effect of the minority group g_2.

We also observe that fairness deteriorates as the value of the noise multiplier increases, as we would expect. This is presumably due to the decreasing influence of the minority group g_2 as the amount of noise insertion increases. This is consistent with the philosophy behind and the definition of DP and its variants. Interestingly we further observe that the personalized model ensures better fairness than the non-personalized model even with the highest level of privacy protection. This shows that personalization in FL under d-privacy can be a comprehensive solution towards privacy-preserving and ethical machine learning as it provides both privacy guarantees and enhanced fairness.

FEMNIST Image Classification

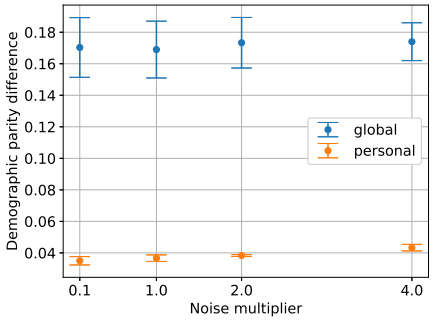
To evaluate the fairness of our method on real datasets, we considered FEMNIST image classification dataset in the same form as in Section 2.5.1. As in experiments performed with the synthetic data in Section 2.5.2, the size of the considered privileged and unprivileged groups were different, denoting the existence of a majority and a minority in the population. In this part, the rotated images are set as the unprivileged group g_2 with a total number of sampled users of 382 forming only 20% of all users. In this case, the total number of sampled users in g_2 is 382. The remaining number of normally-oriented images constitute the privileged group g_1. Like in the case of synthetic data considered before, the group membership was used to denote the sensitive attribute. In the case of g_1, we assign label 1 if the FEMNIST image label is even and 0 if it is odd. And for the g_2, we assign label 0 if the FEMNIST image label is even and assign 1 if it is odd. The experimental results are given by Figure 2.8.

We observe that the personalized model training harbors significantly better group fairness across all metrics compared to its non-personalized counterpart. The change in fairness due to the amount of noise added was not as notable as in the case of the synthetic dataset but it was still observed to deteriorate with an increase in the value of the noise multiplier. Personalized model training in FL under the highest level of privacy is still ob-



(a) Equal opportunity difference

(b) Equalized odds difference



(c) Demographic parity difference

Figure 2.8: For the FEMNIST image classification task, the figure shows the comparison between the personalized and non-personalized models in the same settings described in Figure 2.7. The personalized model again shows improved fairness over the non-personalized model.

served to have better fairness across all the metrics than (non-personalized) models trained in a classical FL framework even with no privacy, similar to what we observed in the experiments with the synthetic data.

2.6 Conclusions

This work presents the challenging task of optimizing federated learning models over the three dimensions of privacy, accuracy, and personalization. The problem of preserving the privacy of individuals is treated under the framework of d-privacy, which provides guarantees of indistinguishability that depend on the distance between any two points, thus eliminating the need to establish an a priori sensitivity threshold. Here, such points lie in the parameter space of machine learning models, which are sanitized and communicated to a central server for aggregation, in order to get closer to the optimal parameters iteratively. Given that the data distribution among individuals is unknown, it is reasonable to assume a mixture of multiple distributions. Clustering the sanitized parameter vectors released by the clients with the k-means algorithm shows to be a good proxy for aggregating clients with similar data distributions. This is possible because d-private mechanisms preserve the topology of the domain of true values. To that end, the Laplace mechanism under Euclidean distance was defined, together with a procedure for sampling from its distribution. Experimental results validate our claims and the limitations of the theory developed here are discussed. In particular, our privacy-preserving mechanism has shown to be promising when machine learning models have a *small* number of parameters. Although formal privacy guarantees degrade sharply with large machine learning models, we show experimentally that the Laplace mechanism under Euclidean distance is still effective at least against the client's data reconstruction by DLG attack. We also evaluated the fairness of machine learning models using personalized federated learning and d-privacy, assessing metrics like equal opportunity, equalized odds, and demographic parity. Our findings show that personalized models enhance group fairness across all metrics and privacy levels, unlike non-personalized models which may be biased towards the majority group. Finally, we highlight how the choice of d-privacy was crucial to achieve these results, as they are a direct consequence of exploiting the distance-based guarantees of metric privacy, which in turn does not require setting a pre-defined clipping threshold.

Chapter 3

Online optimization of the sensitivity

In Chapter 1 we introduced the sensitivity trade-off in the context of the Principal Component Analysis, which stemmed from tuning the clipping threshold of the vectorized data inputs. Chapter 2 focused on the setting of federated learning with averaging of the model parameters and distance-depending privacy guarantees to avoid setting a pre-defined sensitivity. But there are instances where distance-depending privacy guarantees are not enough. In such cases, it is not always possible to rely on the techniques described in Chapter 2, and machine learning models need to be trained within the setting of standard differential privacy, regardless of the centralized or distributed nature of the optimization. In this chapter, we focus on training machine learning models under differential privacy with gradient descent, which involves constraining individuals' contributions by capping the ℓ_2 norm of their gradient at a predetermined threshold. The choice of the clipping threshold significantly hinges on factors such as the dataset, model architecture, and even varies within the same optimization, demanding meticulous tuning usually accomplished through a *grid search*. A grid search methodically tests different combinations of settings for a model's hyperparameters, helping to fine-tune its performance based on specific data and model requirements. To avoid privacy costs from multiple runs during hyperparameter tuning, this chapter introduces a new method for dynamically optimizing the sensitivity in DPML. We consider the clipping threshold as a learnable parameter, creating a direct link between the sensitivity and the cost function. This allows us to optimize it with gradient descent, with minimal repercussions on the overall privacy analysis. Our method is thoroughly assessed against alternative fixed and adaptive strate-

gies across diverse datasets, tasks, model dimensions, and privacy levels. Our results indicate that it performs comparably or better in the evaluated scenarios, given the same privacy requirements expressed at a grid search level. The research presented in this chapter resulted in the following publication:

- Filippo Galli, Catuscia Palamidessi, and Tommaso Cucinotta. Online sensitivity optimization in differentially private learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12109–12117, 2024b

3.1 Introduction

The choice of the gradient clipping threshold C is crucial in gradient-based differentially private machine learning: on the one hand, large values introduce noise levels that may slow down or hinder the optimization altogether; on the other hand, small values introduce a bias in the average clipped gradient with respect to the true average gradient and may leave the optimization stuck in bad local minima. Figure 3.1 exemplifies the issue. Note that the clipping bias is not only directed toward zero (as bounding the ℓ_2 norm may lead to believe), but depends, in general, on the distribution of the per-sample gradients around the expectation [Chen et al., 2020]. Achieving an optimal trade-off remains an ongoing challenge. Historically, researchers have treated the clipping threshold as a parameter to be optimized, often through a grid or random search, in order to assess the performance of privacy-preserving models in the ideal conditions in which an oracle provides the optimal values for the hyperparameters. However, it is worth noting that every additional gradient query to the dataset for optimization purposes introduces a certain degree of privacy leakage. Of late though, the implications of not accounting for privacy leakage over multiple runs of a grid search have drawn more attention, leading to different accounting strategies [Papernot and Steinke, 2022; Mohapatra et al., 2022; Liu and Talwar, 2019]. The inherent challenges of increased privacy leakage and computational overhead resulting from extensive hyperparameter searches persist, necessitating further innovations to encourage broader adoption of differentially private machine learning techniques. Therefore, we set out to provide a strategy for the online optimization of the clipping threshold that is privacy-preserving and computationally inexpensive, while maintaining comparable or better performance on a set of tasks, datasets, and model architectures. This chapter contributes to the state of the art in differentially

private machine learning with the following key contributions:

- It investigates the sensitivity trade-off in differentially private learning in terms of cosine similarity between the sanitized and true gradients, showing that at every iteration it is possible to determine a fairly prominent optimal value
- It elaborates a strategy for the online optimization of the sensitivity, taking from the literature on online learning rate optimization and extending it to optimize the clipping threshold
- It establishes the corresponding techniques for doing so privately, which requires allocating a marginal privacy budget, and
- It provides experimental results to validate this algorithm in multiple contexts and against several relevant state-of-the-art strategies for private hyperparameter optimization.

3.2 Background notions

First, let us formalize gradient-based learning. Gradient-based optimization of (supervised) machine learning models typically implies finding the optimal set of parameters $\theta \in \mathbb{R}^n$ to fit a function $\phi_\theta : \mathcal{X} \to \mathcal{Y}$ to a dataset $D \in \mathcal{D}$ of pairs $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, by minimizing an error function $f : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}_{\geq 0}$. At time t, the iterative optimization process computes the cost of mismatched predictions and updates the parameters towards the nearest local minimum of f by repeated applications of the (Stochastic) Gradient Descent (SGD) algorithm $\theta_{t+1} \leftarrow \theta_t - \rho g_t$, with ρ the learning rate, and

$$g_t = \frac{1}{|B|} \sum_{z_i \in B} \nabla_{\theta_t} f(\phi_{\theta_t}(x_i), y_i) \tag{3.1}$$

being the average gradient of the error function with respect to the parameters, computed over the samples $z_i = (x_i, y_i)$ of the minibatch $B \subseteq D$.

For a broader discussion on differential privacy, we refer the reader to Definition 1 and Section 1.2, but let us recall that to fit machine learning optimization within the definition of a differentially private random mechanism (intended here in a broad sense to also include later generalizations [Dwork and Rothblum, 2016; Mironov, 2017]) the average gradient in Equation (3.1) is sanitized by means of the Gaussian mechanism in Theorem 1. In particular, if $h : \mathcal{X} \to \mathbb{R}^n$ is a function with ℓ_2 norm sensitivity

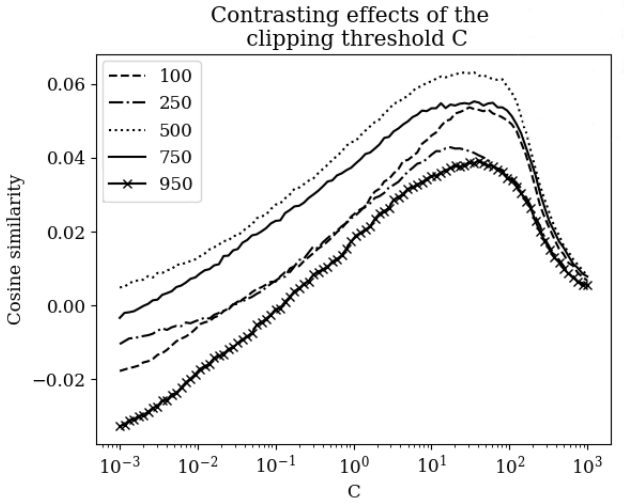


Figure 3.1: The choice of clipping threshold C requires trading off a higher clipping bias at small values, for larger Gaussian noise at large values. Here the clipped, averaged, noised gradient of a CNN for character recognition is compared with the true average gradient at different training iterations $t \in \{100, 250, 500, 750, 950\}$. Note that for some values the sanitized gradient may even have components pointing in the opposite direction w.r.t the true gradient, corresponding to negative cosine similarity. The reported value of cosine similarity is an average over 20 realizations of the Gaussian mechanism.

$S_2(h)$, the DP approximation $\tilde{h}(x)$ of $h(x)$, $x \in \mathcal{X}$, can be found as

$$\tilde{h}(x) = h(x) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2 I = S_2^2(h)\nu^2 I) \tag{3.2}$$

with ν the noise multiplier which depends only on the privacy parameters, and $\mathcal{N}$ being a random normal distribution. Note that although we introduced the notion of a noise multiplier in Chapter 2, and indeed they play similar roles, the definitions are slightly different as they accommodate slightly different DP definitions. Tuning the additive Gaussian noise implies tuning its standard deviation proportionally to the ℓ_2 norm sensitivity of the query g_t over the minibatch B. As, in general, $\|g_t\|_2$ is not bounded *a priori*, the per-sample gradients of the error function are clipped in norm to a certain value C_t [Song et al., 2013; Bassily et al., 2014; Shokri and Shmatikov, 2015b; Abadi et al., 2016] by applying the transformation

$$\bar{g}_t(z_i) = \frac{g_t(z_i)}{\max\left(1, \frac{\|g_t(z_i)\|_2}{C_t}\right)} \tag{3.3}$$

from which follows the sensitivity of the average clipped gradient, allowing for the sanitization of the query at the t^{th} iteration. The Gaussian mechanism lends itself to a refined analysis of the privacy leakage incurred in its repeated application, which is essential in practical machine learning with stochastic gradient descent to keep the overall privacy expenditure to a minimum over multiple training epochs [Abadi et al., 2016; Wang et al., 2019]. A similar procedure can be utilized to account for multiple runs with different configurations in a grid search [Mohapatra et al., 2022].

3.3 Related works

This work draws from two main lines of research, namely hyperparameter optimization in non-private settings and sensitivity optimization in differentially private machine learning. While not exhaustive, the following list of works provides an intuition of the research landscape around the method described in Section 3.4.

Hyperparameter Optimization

Sub-gradient minimization strategies such as SGD iteratively approach the optimal solution by taking steps in the direction of the steepest descent of a cost function. For this heuristic to be effective, the length of each step needs to be tuned by controlling the *learning rate*, which has been considered the "single most important hyperparameter" [Bengio, 2012]. Many works have introduced strategies for its adaptive tuning, such as [Lydia and Francis, 2019; Kingma and Ba, 2015], which adjust the per-parameter value w.r.t. a common value still defined *a priori*. Conversely, other research has exploited automatic differentiation to concurrently optimize the parameters and hyperparameters [Maclaurin et al., 2015] via SGD. In particular, explicitly deriving the partial derivative of the cost function with respect to the learning rate has been demonstrated to be an effective strategy, and it has been discovered independently at different times [Almeida et al., 1999; Baydin et al., 2018]. These works do not explore the private setting and introduce general methods that are almost exclusively applied to learning rate optimization, without addressing the choice of other hyperparameters. In [Mohapatra et al., 2022] instead, the authors study adaptive optimizers in the differentially private setting, by analyzing the estimate of the raw second moment of the gradient at convergence. Their objective is to reduce the privacy cost of tuning the learning rate in a grid search, but the clipping threshold is still treated as an additional hyperparameter.

Sensitivity Optimization

As discussed in Sections 3.1 and 3.2, establishing the value of the clipping threshold C_t is critical in differentially private machine learning, and treating this value as a hyper-parameter has largely been the preferred strategy in the literature [Song et al., 2013; Bassily et al., 2014; Shokri and Shmatikov, 2015b; Abadi et al., 2016]. Grid searching over the candidate values can be tricky as gradient norms may span many orders of magnitude and the effects of more aggressive clipping are not easily predicted before running an optimization. Considering also the increased privacy costs of running multiple configurations, hyperparameter selection under privacy constraints is a thriving research area [Papernot and Steinke, 2022; Liu and Talwar, 2019; Mohapatra et al., 2022].

Adaptive clipping strategies have also been considered. [Andrew et al., 2021] updates C_t during training to match a target quantile of the gradient norms, which is fixed beforehand. Although the optimal quantile is still a hyper-parameter, its domain is limited to the $[0, 1] \subset \mathbb{R}$ interval. Moreover, [Andrew et al., 2021] shows that adaptively updating C_t outperforms even the best fixed-clipping strategy. Additionally, as DP training has been shown to disproportionally favor majority classes in a dataset [Suriyakumar et al., 2021], tuning a target quantile instead of a fixed clipping threshold may help at least in quantifying the issue, if not in solving it. Note that although this strategy was introduced to train differentially private *federated* machine learning models, the attacker is still modeled as an *honest-but-curious* adversary and thus it relies on a central trusted server to provide DP guarantees. Therefore, the clipping strategy in [Andrew et al., 2021] can be used to train *centralized* machine learning models just by switching from user-level to sample-level differential privacy [McMahan et al., 2018b]. To further stress this point, note that although in [Andrew et al., 2021] every single user clips the update and sends statistics to the central server, from a differential privacy point of view this is identical to the server performing these operations itself on the true per-user gradients.

3.4 Method

Inspired by the literature on online hyperparameter optimization discussed in Section 3.3, the idea behind this method is to optimize the clipping threshold based on the chain rule for derivatives, so that we can find what change in C_t will induce a decrease in the cost function $f(\theta_t)$. Although this strategy works in general for sub-gradient methods, we are going to

explicitly derive the results for DP-SGD. Given the SGD update rule with gradient clipping:

$$\theta_{t+1} = \theta_t - \rho \nabla f(\theta_t) \tag{3.4}$$

$$= \theta_t - \rho \frac{1}{|B_t|} \sum_{z_i \in B_t} \frac{g_t(z_i)}{\max(1, \|g_t(z_i)\|_2 / C_t)} \tag{3.5}$$

and we want to find:

$$\frac{\partial f(\theta_t)}{\partial C} = \frac{\partial f(\theta_t)}{\partial \theta_t}^\top \frac{\partial \theta_t}{\partial C_t} \tag{3.6}$$

$$= \nabla f(\theta_t)^\top \frac{\partial \theta_t}{\partial C_t} \tag{3.7}$$

Recall that $\theta_t = \theta_{t-1} - \rho \nabla f(\theta_{t-1})$ and by assuming $C_t \approx C_{t-1}$ we can expand $\frac{\partial \theta_t}{\partial C_t}$ to find

$$\frac{\partial \theta_t}{\partial C_t} \approx \frac{\partial \theta_t}{\partial C_{t-1}} = \frac{\partial \left(\theta_{t-2} - \rho \nabla f(\theta_{t-2}) - \rho \nabla f(\theta_{t-1})\right)}{\partial C_{t-1}}$$

And we have that

$$\frac{\partial \theta_{t-2}}{\partial C_{t-1}} = 0; \quad \frac{\partial \nabla f(\theta_{t-2})}{\partial C_{t-1}} = 0$$

as quantities in the past (at time $t-2$) do not depend on future quantities (at time $t-1$). This results in

$$\frac{\partial \theta_t}{\partial C_t} = -\rho \frac{\partial \nabla f(\theta_{t-1})}{\partial C_{t-1}} \tag{3.8}$$

$$\frac{\partial f(\theta_t)}{\partial C_t} = -\rho \nabla f(\theta_t)^\top \frac{\partial \nabla f(\theta_{t-1})}{\partial C_{t-1}} \tag{3.9}$$

Crucially, we don't assume $C_t \approx C_{t-2}$. We further emphasize how these assumptions are common in optimization theory, as in [Almeida et al., 1999; Baydin et al., 2018]. To find an explicit form for Equation (3.9), we notice that the rightmost term is differentiable almost everywhere, with:

$$\frac{\partial \nabla f(\theta_{t-1})}{\partial C_{t-1}} = q_{t-1} = \frac{1}{|B_{t-1}|} \sum_{z_i \in B_{t-1}} q_{t-1}(z_i) \tag{3.10}$$

and

$$q_{t-1}(z_i) = \begin{cases} \frac{g_{t-1}(z_i)}{\|g_{t-1}(z_i)\|_2} & \text{if } \|g_{t-1}(z_i)\|_2 > C_{t-1} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

where we highlight that $\|q_{t-1}(z_i)\|_2 \in \{0, 1\}, \forall z_i \in B_{t-1}$ by definition of the clipping function. Thus we find:

$$\frac{\partial f(\theta_t)}{\partial C_t} = -\rho \nabla f(\theta_t)^\top \frac{1}{|B_{t-1}|} \sum_{z_i \in B_{t-1}} q_{t-1}(z_i) \tag{3.12}$$

resulting in the gradient descent update rule for the clipping threshold:

$$C_{t+1} = C_t + \rho_c \rho \nabla f(\theta_t)^\top \frac{1}{|B_{t-1}|} \sum_{z_i \in B_{t-1}} q_{t-1}(z_i) \tag{3.13}$$

which is the dot product of the current average gradient with a masked version of the last iteration's average gradient, where all per-sample gradients have either norm 0 or 1.

Notice we use the non-privatized version of the update rule to expand Equation (3.7), i.e. we don't *explicitly* account for the Gaussian noise coming from the DP approximation in Equation (3.2). In fact, to account for the Gaussian noise in our method means to introduce the random vector in Equations (3.7) through (3.9). In particular, we would need to take $\theta_t = \theta_{t-1} - \rho \nabla \ell(\theta_{t-1}) + \gamma$, with $\gamma \sim \mathcal{N}(0, C^2 k^2 I)$, for some privacy-dependent k. Expanding, we find:

$$\begin{aligned} \frac{\partial \theta_t}{\partial C} &= \frac{\partial \left(\theta_{t-1} - \rho \nabla f(\theta_{t-1})\right)}{\partial C} + \frac{\partial \gamma}{\partial C}; \quad \gamma \sim \mathcal{N}(0, C^2 k^2 I) \\ &= \frac{\partial \left(\theta_{t-1} - \rho \nabla f(\theta_{t-1})\right)}{\partial C} + \frac{\partial C k \tilde{\gamma}}{\partial C}; \quad \tilde{\gamma} \sim \mathcal{N}(0, I) \end{aligned}$$

where

$$\frac{\partial C k \tilde{\gamma}}{\partial C} = \frac{\partial C}{\partial C} k \tilde{\gamma} + C k \frac{\partial \tilde{\gamma}}{\partial C} = k \tilde{\gamma}$$

and thus

$$\frac{\partial \theta_t}{\partial C} = \frac{\partial \left(\theta_{t-1} - \rho \nabla f(\theta_{t-1})\right)}{\partial C} + k \tilde{\gamma}$$

Note that the term $k\tilde{\gamma}$: i) has zero mean, ii) it is not needed from a privacy perspective (as it is just a consequence of the DP sanitization of the average gradient and because DP is immune to post-processing) and iii) it disturbs the optimization with a k standard deviation Gaussian noise. For these reasons, we take the expectation (which is $\mathbb{E}[k\tilde{\gamma}] = 0$) and we do not include it in the optimization. Nevertheless, recall that, ultimately, if an increase in the sensitivity will induce a noise large enough to affect the cost function, the clipping threshold will be reduced, by construction of the method itself.

On a separate but related matter, we highlight how having an adaptive clipping strategy may still slow down convergence if the learning rate is kept fixed at the starting value, considering the coupled dynamics of the learning rate and the clipping threshold [Mohapatra et al., 2022]. Thus, we use the same method to derive an update strategy for the learning rate ρ_t, as in [Almeida et al., 1999; Baydin et al., 2018]:

$$\frac{\partial f(\theta_t)}{\partial \rho_t} = \frac{\partial f(\theta_t)}{\partial \theta_t}^\top \frac{\partial \theta_t}{\partial \rho_t} \tag{3.14}$$

$$= \nabla f(\theta_t)^\top \nabla f(\theta_{t-1}) \tag{3.15}$$

which results in the dot product of the current and past clipped gradients, yielding:

$$\rho_{t+1} = \rho_t + \rho_r \rho \nabla f(\theta_t)^\top \nabla f(\theta_{t-1}) \tag{3.16}$$

Note obtaining the quantities in this last result does not require a dedicated procedure, as they are already a byproduct of SGD to optimize θ_t, even in a non-private setting.

3.5 Privacy Analysis

When assuming a time-dependent C_t such as in [Andrew et al., 2021], it is particularly useful to decouple the contributions of the sensitivity from contributions of the privacy parameters (ε, δ) to the variance of the Gaussian mechanism, as in Equation (3.2). Then, within the framework of Rényi DP[1] and given the results in [Mironov, 2017; Wang et al., 2019] one can efficiently determine ahead of training time the values of noise multiplier to be applied at each iteration independently of the current value of C_t. At the t^{th} iteration there may be two sources of differential privacy leakage: the computation of θ_{t+1} in Equation (3.4) and the computation of C_{t+1} in Equation (3.13). Both can be sanitized with the DP approximation already discussed, but the latter needs special attention. To sanitize C_{t+1} with the Gaussian mechanism (for reasons detailed in Proposition 1) we may utilize $\nabla f(\theta_t) \approx \nabla \tilde{f}(\theta_t)$, effectively repurposing the sanitized gradient with respect to θ_t. We focus now on the non-privatized term $\partial \nabla f(\theta_{t-1}) / \partial C_{t-1}$. Naturally, it still involves the sanitization of a sum of vectors, with the fortunate benefit of having all the terms in the summation be of norm either

[1] We stress the use of Rényi DP in this context is limited to achieving stricter results for compositions of *standard* differential privacy queries than would be possible with earlier advanced compositionality Theorems in [Dwork et al., 2014a]. In that sense, we can use it as a more refined *moments accountant* introduced in [Abadi et al., 2016].

0 or 1, as shown in Equation (3.11), resulting in the unit sensitivity of the query. Thus, this step does not introduce the need to develop any further higher-order (adaptive) clipping strategies. With the considerations above, from a privacy perspective, the two privatized parallel queries behave as a single query sanitized with the Gaussian mechanism. This result is formalized in Proposition 1, which follows from the *joint clipping* strategy described in [McMahan et al., 2018a].

Proposition 1. *The Gaussian approximations $\tilde{q}_t$ and $\tilde{g}_t$ of $\sum_{z_i \in B_{t-1}} q_{t-1}(z_i)$ and $\sum_{z_i \in B_t} \bar{g}_t(z_i)$ with noise multipliers, respectively, ν_q and ν_g, is equivalent (as far as privacy accounting is concerned) to the application of a single Gaussian mechanism with noise multiplier ν if $\nu_g = (\nu^{-2} - \nu_q^{-2})^{-1/2}$.*

Compared to Theorem 1 in [Andrew et al., 2021], we lose a factor of 2 in the reduction of the standard deviation $\sigma_q = 1 \cdot \nu_q$ and since ν_q is used here to sanitize a sum of vectors in $\mathbb{R}^n$ (whereas [Andrew et al., 2021] only need to sanitize a scalar quantity) we cannot relegate as much differentially private noise to the computation of $\tilde{q}_t$. Nonetheless, we can derive a rule of thumb, which, together with practical considerations introduced in the next Section, allow to have working estimates of the true $\partial f(\theta_t)/\partial C_t$. In particular, if we allow a 1% increase in ν_g over ν, we can rearrange the result in Proposition 1 to find $\nu_q \approx 7.124 \cdot \nu$. Figure 3.2 shows an example of these trade-offs for the MNIST dataset discussed in Section 3.7.

To complete the privacy analysis, we highlight that from a DP point of view, the updates to the learning rate described in Equation (3.16) come with no additional privacy expenditure with respect to DP-SGD, exploiting the sanitized $\tilde{g}_t$ and $\tilde{g}_{t-1}$.

3.6 The OSO-DPSGD Algorithm

The algorithm keeps track of two sanitized quantities at each iteration, that is:

$$\tilde{q}_t = \frac{1}{|B_t|} \sum_{z_i \in B_t} q_t(z_i) + \eta, \quad \eta \sim \mathcal{N}(0, \nu_q^2 I) \tag{3.17}$$

$$\tilde{g}_t = \frac{1}{|B_t|} \sum_{z_i \in B_t} \bar{g}_t(z_i) + \eta, \quad \eta \sim \mathcal{N}(0, C_t^2 \nu_g^2 I) \tag{3.18}$$

from which one can privately compute the parameter update and $\partial \tilde{f}(\theta_t)/\partial C_t = -\rho \tilde{g}_t^\top \tilde{q}_{t-1}$, which requires to store $\tilde{q}_{t-1}$ from the last iteration. Note that

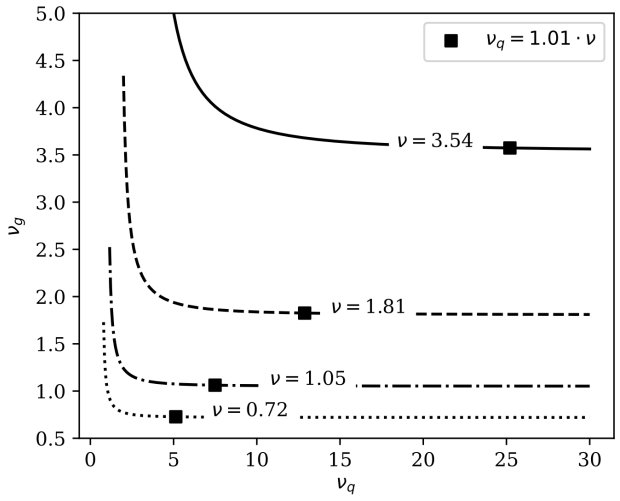


Figure 3.2: The Pareto frontiers of the noise multipliers to sanitize $\tilde{g}_t$ and $\tilde{q}_t$, and the chosen values given the heuristic described in Section 3.5, at different privacy requirements. This particular instance comes from the MNIST experiments described in the Section 3.7.

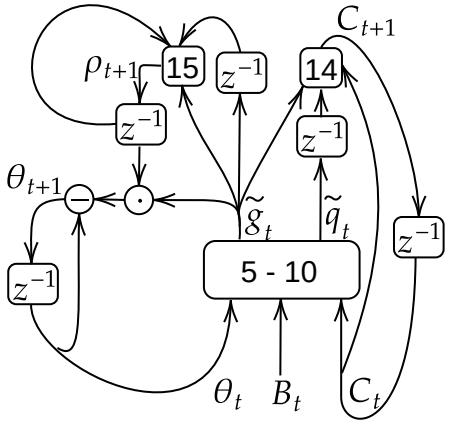


Figure 3.3: Graphical interpretation of Algorithm 3, where z^{-1} represents the time-shift operator and the numbers in boxes the corresponding lines in the Algorithm.

storing vectors from past iterations is a common strategy even in non-privatized learning, as e.g. it is required by every optimizer with momentum(s). In order to cater to the wide range of values C_t might take, spanning orders of magnitude [Andrew et al., 2021], instead of relying on the additive update rule in Equation (3.13), we first consider the scale-invariant Equation (3.19) proposed in [Rubio, 2017], which converges with a logarithmic number of steps, instead of linearly

$$C_{t+1} = C_t \cdot \left(1 + \rho_c \frac{\tilde{g}_t{}^\top \tilde{q}_{t-1}}{\|\tilde{g}_t\|_2 \, \|\tilde{q}_{t-1}\|_2}\right) \tag{3.19}$$

We briefly experimented with Equation (3.19) and found the proportional update step $\tilde{g}_t{}^\top \tilde{q}_{t-1} / \left\|\tilde{g}_t{}^\top \tilde{q}_{t-1}\right\|_2 = \text{sign}(\tilde{g}_t{}^\top \tilde{q}_{t-1})$ to be more robust w.r.t. the Gaussian noise and less dependent on the particular choice of ρ_c. Noticing that $1 + x \approx e^x$ for small values of x, we converge to an exponential update rule for the optimization of both C_t and ρ_t, similar to [Andrew et al., 2021]:

$$C_{t+1} = C_t \cdot \exp(\rho_c \text{sign}(\tilde{g}_t{}^\top \tilde{q}_{t-1})) \tag{3.20}$$

$$\rho_{t+1} = \rho_t \cdot \exp(\rho_r \text{sign}(\tilde{g}_t{}^\top \tilde{g}_{t-1})) \tag{3.21}$$

Although we provide the result for vanilla SGD, deriving the update rule for the case with first-order momentum is trivial and only adds a multiplicative factor to $\partial f / \partial C$, depending on the specific implementation of momentum. The same analysis for Adam is more involved and most importantly it results in the summation in $\partial \nabla f(\theta_{t-1}) / \partial C_{t-1}$ to lose the appealing property of unitary sensitivity. Considering also the disparate results of Adam as a DP optimizer [Mohapatra et al., 2022; Andrew et al., 2021], we leave this analysis for future work. Finally, Algorithm 3 outlines the online optimization strategy presented above, which we call `OSO-DPSGD`, of which Figure 3.3 shows a graphical representation.

In Algorithm 3 we list the learning rates of C_t and ρ as hyperparameters. In practice, especially considering the exponential update rule in Equations 3.20 and 3.21, they can be set to the same value. After a qualitative exploration of reasonable values for both, we settle on $\rho_c = \rho_r = 2.5 \cdot 10^{-3}$ for all the experiments.

3.7 Experiments

In the following section, we proceed to assess Algorithm 3 on a range of experiments on different datasets, tasks, and model sizes. In particular, we

Algorithm 3 Differentially private optimization with `OSO-DPSGD`

Inputs Samples $z_i \in D$; ρ; T; C_0; θ_0; $|B|$; per-iteration noise multiplier ν; ν_q; ρ_c; ρ_r; $\tilde{q}_0 = \tilde{g}_0 = 0$.
$\nu_g \leftarrow (\nu^{-2} - \nu_q^{-2})^{-1/2}$
for $t \in \{1, \dots, T\}$ **do**
  $B_t \leftarrow$ draw $|B|$ samples uniformly from D
  for $z_i \in B_t$ **in parallel do**
    $g_t(z_i) \leftarrow \nabla_{\theta_t} f(\theta_t, z_i)$
    $\bar{g}_t(z_i) \leftarrow g_t(z_i) / \max(1, \frac{\|g_t(z_i)\|_2}{C_t})$
    $q_t(z_i) \leftarrow \frac{g_t(z_i)}{\|g_t(z_i)\|_2}$ if $\|g_t(z_i)\|_2 > C_t$ else 0
  end for
  $\sigma_g \leftarrow \nu_g C_t$
  $\tilde{g}_t \leftarrow \frac{1}{|B|} \left(\sum_{z_i \in B_t} \bar{g}_t(z_i) + \mathcal{N}(0, I\sigma_g^2) \right)$
  $\theta_{t+1} \leftarrow \theta_t - \rho \hat{g}_t$
  $\tilde{q}_t \leftarrow \frac{1}{|B|} \left(\sum_{z_i \in B_t} q_t(z_i) + \mathcal{N}(0, I\nu_g^2) \right)$
  $C_{t+1} \leftarrow C_t \exp(\rho_c \text{sign}(\tilde{g}_t{}^\top \tilde{q}_{t-1}))$
  $\rho_{t+1} \leftarrow \rho_t \exp(\rho_r \text{sign}(\tilde{g}_t{}^\top \tilde{g}_{t-1}))$
end for

explore how online sensitivity optimization can be an effective tool in reducing the privacy and computational costs of running large grid searches. In an effort to draw conclusions that can be as general as possible, we identify three vastly adopted datasets in the literature: MNIST [LeCun et al., 1998], FashionMNIST [Xiao et al., 2017], and AG News [Gulli, 2005] [Zhang et al., 2015]. They are used to train, respectively, a convolutional neural network for image classification (Table 3.5), a convolutional autoencoder (Table 3.6) and a bag-of-words fully connected neural network for text classification (Table 3.7).

Considering the computational burden of benchmarking multiple grid searches, we devise the following pipeline:

- Define the different learning algorithms; to compare `OSO-DPSGD` with relevant strategies, we also include in our experiments the `Fixed Threshold` of [Song et al., 2013] [Shokri and Shmatikov, 2015b] [Abadi et al., 2016] among others and `FixedQuantile` of [Andrew et al., 2021]. As reported by the respective authors, hyperparameter optimization is performed via grid search over the learning rates and threshold values for the former and over the learning rates and the quantiles for the latter. Even though [Mohapatra et al., 2022] introduce `AdamWOSM` for the DP adaptive optimization of the

	AG News	MNIST	Fashion MNIST
Dataset Size	120000	60000	60000
Batch Size	512	512	512
Model Size	113156	551322	48705

Table 3.1: Dataset and model information shared throughout the experiments.

learning rate, it still tackles the challenge of reducing the number of hyperparameters in a privacy-aware grid search, and therefore we include it.

- Establish the corresponding grid search ranges. In all of our experiments, we fix the ranges of the hyperparameters to the same values. Considering the variety of experiments, and without assuming any particular domain knowledge of the task at hand, we opt for large ranges: $C \in [10^{-2}, 10^{2}]$ for the clipping threshold, $\rho \in [10^{-2.5}, 10^{1.5}]$ for the learning rate and $\gamma \in [0.1, 0.9]$ for the target quantile.

- Define grid searches with different granularity. Given the ranges defined in the last step, DP training introduces possibly yet another hyperparameter. In fact, increasing the granularity inevitably results in more candidates, and an additional trade-off to consider is that of increased fine-tuning at the cost of additional privacy leakage. In our experiments, we evaluate 3 grid searches with different granularity, i.e. from the ρ and C ranges in the last step we take $k \in \{5, 7, 9\}$ values uniformly separated in a logarithmic scale. For the experiments with the `FixedQuantile` strategy, we keep the values $\gamma \in [0.1, 0.3, 0.5, 0.7, 0.9]$ defined by the authors in [Andrew et al., 2021], as well as setting the learning rate for the exponential update rule for C to 0.2. The initial value for the clipping threshold in both `FixedQuantile` and `Online` is set to $C_0 = 0.1$.

- Execute private hyperparameter optimization at different privacy levels. For the same δ, we explore with increasing values of ε. Following [Mohapatra et al., 2022], the privacy budgets we establish are

per-grid, and not per-run. That is, algorithms that need extra fine-tuning and additional parameters, resulting in more runs, will effectively reduce the per-run privacy budget. Although this setting may not conform to most past literature, we are motivated by approaching DP machine learning from the practitioner's point of view, where an oracle providing the optimal hyperparameters may not be a reasonable assumption. As in [Mohapatra et al., 2022], we utilize the moment accountant to distribute the privacy budget among the configurations, as we do not have a large number of candidates.

On top of comparing DP learning strategies, we provide a baseline in the non-private setting, where we iterate only over the learning rate values and initial weights. To limit the contribution of the Gaussian random noise in the DP setting, each configuration is executed with 5 different seeds, and the results are averaged. Runs with different seeds are not accounted for in terms of privacy budget. Given the large number of runs, we validate each model at training time every 50 iterations on the full test set and pick the model checkpoint at the best value as representative of the corresponding configuration. Each configuration runs for 10 epochs regardless of when the best performance is registered. Given the model size and datasets, the total number of epochs is enough to have most configurations converge. Nevertheless, we don't expect *every* combination of hyperparameters to saturate learning, e.g. when training with C and ρ both set at the lowest value available in the corresponding ranges. In Tables 3.2, 3.3, 3.4, we list the hyperparameters leading to the best results in the grid search with granularity $k = 7$ for the corresponding datasets and models. For improved readability, we discuss the results only for this specific setting and defer the graphical and numerical results to this Chapter's Appendix in Section 3.9 for all the remaining granularity settings. Notice, however, that the reader will see the same trends and draw the same conclusions.

Discussion Figure 3.4 shows the accuracy of the models in the best configurations, among those tested, on the MNIST dataset. Even though at higher privacy levels (low ε) `Online` and `AdamWOSM` appear to be equivalent in terms of results, we can see the former showing better results when the privacy requirements are relaxed. A possible explanation may be found in Table 3.2 by noticing that the best C value for `AdamWOSM` is fairly large compared to the other strategies. We believe that a larger initial value for C may be positive to take long strides towards the direction of the average gradient at the early stages of the optimization, but may be detrimental towards the end when reducing the Gaussian noise may help the optimization. Nevertheless, we consider both strategies to be roughly

	Online	Fixed Threshold		Fixed Quantile		Adam WOSM
ε	ρ	ρ	C	ρ^*	γ	C
3	0.3162	0.01467	1.0	3.162	0.5	21.54
5	1.467	0.003162	4.64	3.162	0.7	21.54
7	1.467	6.812	0.010	3.162	0.7	21.54
9	1.467	6.812	0.010	3.162	0.7	21.54

Table 3.2: Best hyperparameters for the MNIST dataset with grid search granularity $k = 7$. Values with * are scaled $\times 10^3$ for better readability. Best `NoDP` result for $\rho = 0.003162$.

	Online	Fixed Threshold		Fixed Quantile		Adam WOSM
ε	ρ	ρ	C	ρ	γ	C
1	0.3162	0.0681	0.010	-	-	0.01
2	1.467	1.467	0.010	0.3162	0.3	0.01
3	1.467	6.812	0.010	1.467	0.1	0.0464
4	1.467	1.467	0.0464	1.467	0.3	0.01

Table 3.3: Best hyperparameters for the Fashion MNIST dataset with grid search granularity $k = 7$. Best `NoDP` result for $\rho = 0.01467$. All `FixedQuantile` runs diverge for $\varepsilon = 1$.

equivalent in this experiment. The results for `FixedThreshold` and `FixedQuantile` are consistently lower, most likely due to both strategies needing a larger grid search, which in turn limits the per-run privacy budget. Perhaps more surprisingly, the adaptive strategy `FixedQuantile` does not seem to show better results compared to fixing the clipping threshold at the initial value. The improved results that are found in [Andrew et al., 2021] in the federated setting do not seem to translate in centralized learning, with the experiments we conducted.

Figure 3.5 shows the best results in terms of mean squared error on the FashionMNIST dataset, where a model is trained to encode and decode the input images of clothing items. The chosen architecture is based on a convolutional autoencoder and it has the smallest number of parameters among those considered in this work, as in Table 3.1. The privacy regimes are then chosen accordingly. Firstly, we notice that for $\varepsilon = 1$ the `FixedQuantile` strategy does not converge with any of the available

	Online	Fixed Threshold		Fixed Quantile		Adam WOSM
ε	ρ	ρ	C	$\rho*$	γ	C
3	0.06812	1.467	0.01	3.162	0.5	0.01
5	0.06812	1.467	0.010	3.162	0.5	0.01
7	0.06812	1.467	0.010	3.162	0.7	0.01
9	0.06812	0.03162	0.0464	3.162	0.7	0.01

Table 3.4: Best hyperparameters for the AG News dataset. Values with * are scaled $\times 10^3$ for better readability. $k = 7$. Best `NoDP` result for $\rho = 0.003162$.

hyperparameters. To justify this result, we highlight how in Table 3.3 all other strategies adopt aggressive clipping strategies with small C's. We thus believe that for very high privacy regimes even running with $\gamma = 0.1$ (the lowest value for the target quantile) may induce large swings in the exponential updates of C_t, disrupting the optimization. Nevertheless, for $\varepsilon \in \{2, 3, 4\}$ this strategy shows the second best results. Conversely, `AdamWOSM` may be penalized by the choice of the initial $\rho_0 = 10^{-3}$, as suggested by the authors in [Mohapatra et al., 2022]. In fact, we notice from Table 3.4 that the optimal clipping threshold is very small in all competing strategies, and the combination of small C and small ρ_0 may render the optimization excessively slow to converge within the set number of epochs. Further, it may suggest that adapting the learning rate on a per-parameter basis, as in `AdamWOSM`, can be effective as long as the base learning rate is itself carefully selected. Thus, optimizing ρ_t in the grid search, and then adaptively tuning it within the same run, as done in `Online`, seems to show better results.

Figure 3.6 plots the accuracy on the AG News dataset, where a bag of words model with a fully connected neural network is used to classify a selection of news in one of four classes. In this experiment, we notice that `AdamWOSM` performs the best, with `Online` being marginally below. Still, as with the MNIST dataset, we take both strategies to be comparable in these two settings, as the average of one roughly fits within a standard deviation of the other.

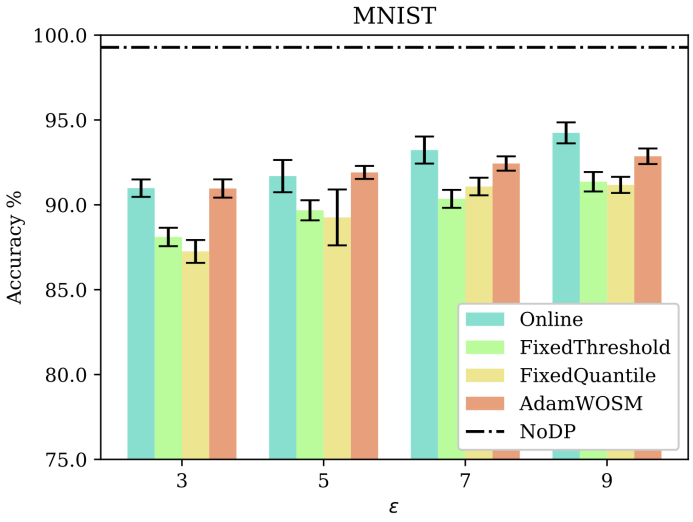


Figure 3.4: Accuracy on the MNIST dataset. Higher is better.

3.8 Conclusions

This chapter studies differentially private machine learning in the context of hyperparameter optimization, where the privacy cost of running a grid search is accounted for. Under these conditions, algorithms that require one less parameter may be preferable. Thus we explore strategies for the adaptive tuning of the sensitivity C, and derive a result inspired by online learning rate optimization. With the proposed strategy, which we incorporate in the `OSO-DPSGD` algorithm, the clipping threshold is updated at each iteration based on the direction of the steepest descent of the cost function. The resulting update rule is particularly clean, and results in the dot product between two sanitized vector queries: the average gradient at time t, and the derivative w.r.t. C of the gradient at time $t-1$. With the former already needed in standard `DP-SGD`, and the latter resulting in a query with unitary sensitivity, the additional computational and privacy burden is minimal. Our range of experiments seems to encourage further research in this area, as online sensitivity optimization shows comparable results with one less parameter when assessed against standard state-of-the-art algorithms, if the privacy guarantees are required at a grid search level, and not just within a single run. Hopefully, a refined analysis and algorithm will possibly achieve better results even in this latter setting of per-run privacy requirements.

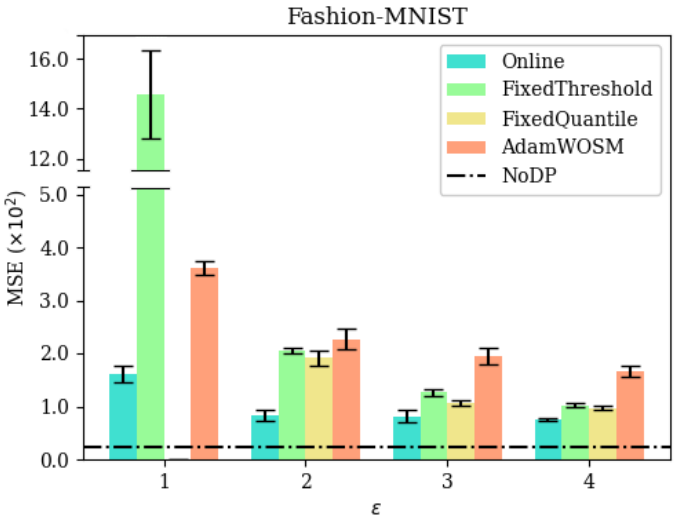


Figure 3.5: Mean Squared Error on the Fashion MNIST dataset. Lower is better. All runs for $\varepsilon = 1$ of `FixedQuantile` result in a diverging optimization and are therefore not included.

3.9 Appendix: further experiments and details

As mentioned in the corresponding Section 3.7, we provide additional details about the model architectures, datasets, and results of experiments at different granularity levels. The model architectures are outlined in Tables 3.5, 3.6, 3.7. All datasets go through minor pre-processing, that is pixel values are mapped to the $[0, 1]$ interval, while the text-based dataset AG News first goes through word embedding, using an embedding size of 50 for up to the first 25 words. To speed up development, we use the pre-trained word embeddings from [Pennington et al., 2014]. Next, we report the results for granularity $k = 5$ in Tables 3.11, 3.12 3.13 and Figures 3.10, 3.11 and 3.12. For $k = 9$, results are presented in Tables 3.8, 3.9, 3.10 and Figures 3.7, 3.8 and 3.9. We reiterate how these tables are postponed to the Appendix because they essentially show the same results and trends of the experiments discussed in the rest of the chapter.

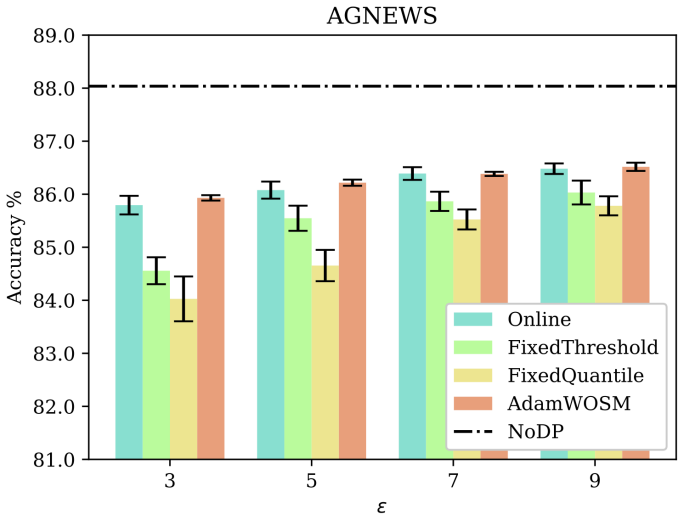


Figure 3.6: Accuracy on the AG News dataset. Higher is better.

CNN			
layer	kernel size output size	stride	non linearity
2D Conv	$16 \times 8 \times 8$	1×1	ReLU
2D MaxPool	2×2	1×1	-
2D Conv	$32 \times 4 \times 4$	1×1	ReLU
Linear	32	-	ReLU
Linear	10	-	ReLU

Table 3.5: CNN

AutoEncoder			
layer	kernel size output size	stride	non linearity
2D Conv	$8 \times 3 \times 3$	1×1	LeakyReLU
2D Conv	$16 \times 3 \times 3$	1×1	LeakyReLU
2D Conv	$32 \times 3 \times 3$	1×1	LeakyReLU
2D Conv	$64 \times 3 \times 3$	1×1	LeakyReLU
2D Transpose Conv	$32 \times 3 \times 3$	1×1	LeakyReLU
2D Transpose Conv	$16 \times 3 \times 3$	1×1	LeakyReLU
2D Transpose Conv	$8 \times 3 \times 3$	1×1	LeakyReLU
2D Transpose Conv	$1 \times 3 \times 3$	1×1	Sigmoid

Table 3.6: AutoEncoder

BagOfWords - FC		
layer	output size	non linearity
Linear	128	LeakyReLU
Linear	128	LeakyReLU
Linear	4	LeakyReLU

Table 3.7: Bag of Words model architecture with a fully connected neural network.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	acc	std dev	ρ	C	acc	std dev	ρ^*	γ	acc	std dev	C	acc	std dev
3	0.316	90.69	0.53	100.0	0.1000	86.83	0.70	3.162	0.5	85.48	0.39	10.0	90.62	0.52
5	1.00	91.62	0.83	3.162	3.162	88.69	0.57	3.162	0.5	88.13	0.59	10.0	91.60	0.39
7	1.00	92.94	0.90	3162.0	0.0100	90.04	0.49	3.162	0.7	90.73	0.72	31.62	92.19	0.37
9	1.00	93.62	0.91	3.162	10.00	90.72	0.47	3.162	0.7	91.09	0.51	31.62	92.53	0.33

Table 3.8: MNIST, $k = 9$, best `NoDP` for $\rho = 0.003162$, * values are scaled $\times 10^3$.

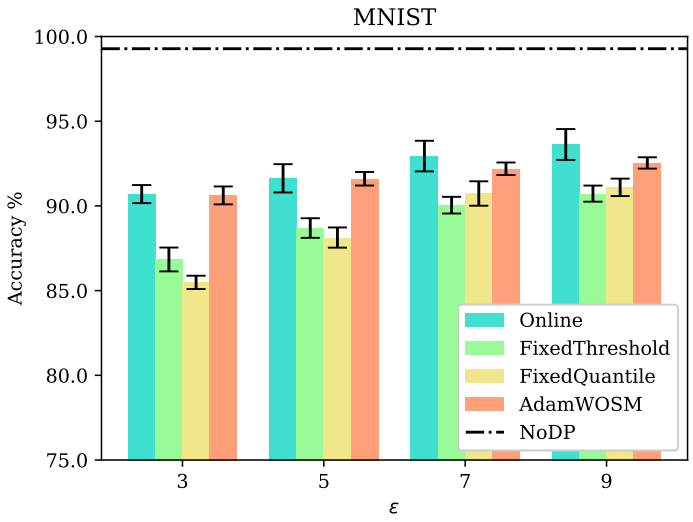


Figure 3.7: $k = 9$. Accuracy on the MNIST dataset. Higher is better. Refer to Table 3.8 for numeric results and optimized hyperparameters.

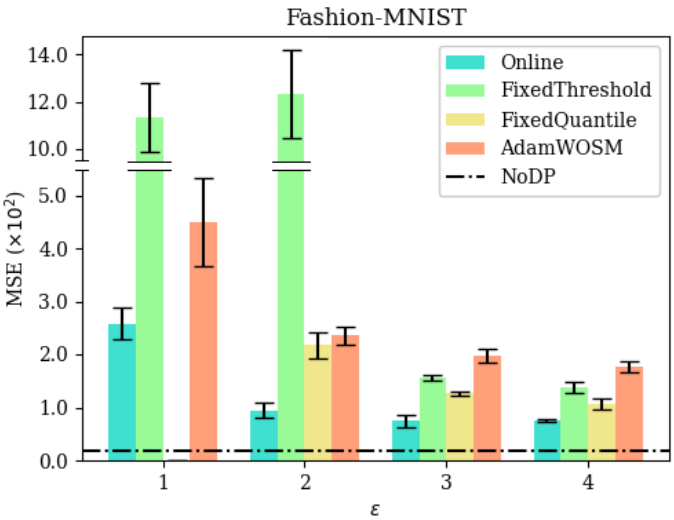


Figure 3.8: $k = 9$. Mean Squared Error on the Fashion MNIST dataset. Lower is better. Refer to Table 3.9 for numeric results and optimized hyperparameters.

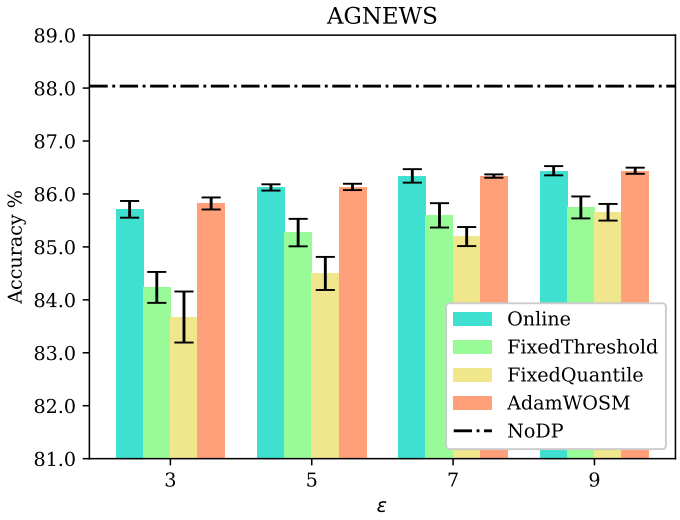


Figure 3.9: $k = 9$. Accuracy on the AG News dataset. Higher is better. Refer to Table 3.10 for numeric results and optimized hyperparameters.

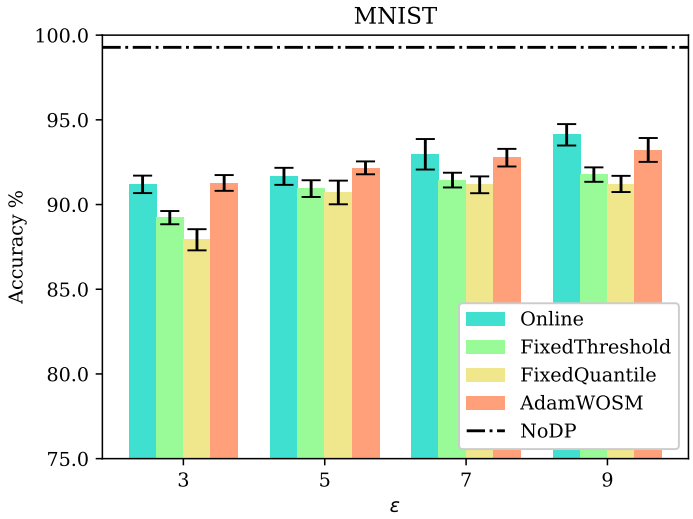


Figure 3.10: $k = 5$. Accuracy on the MNIST dataset. Higher is better. Refer to Table 3.11 for numeric results and optimized hyperparameters.

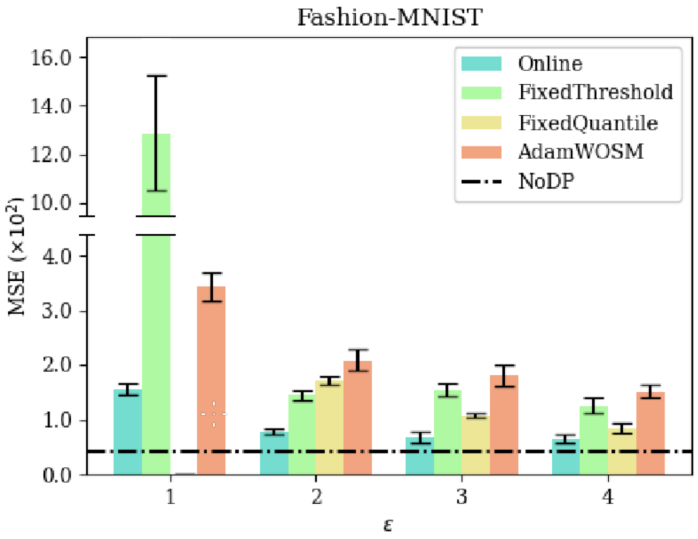


Figure 3.11: $k = 5$. Mean Squared Error on the Fashion MNIST dataset. Lower is better. Refer to Table 3.12 for numeric results and optimized hyperparameters.

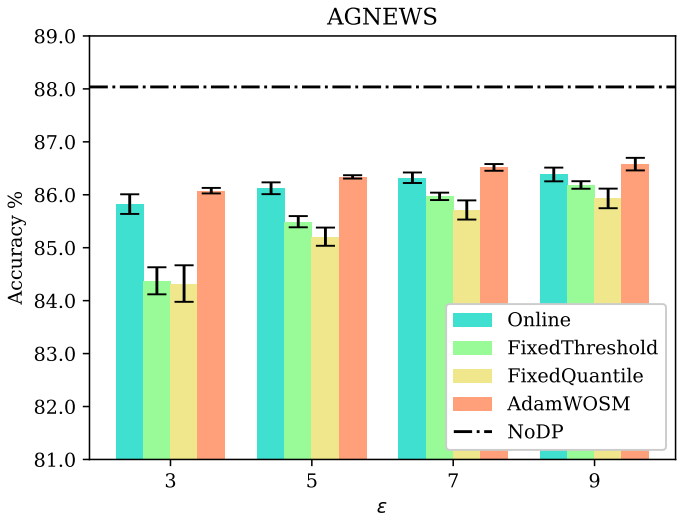


Figure 3.12: $k = 5$. Accuracy on the AG News dataset. Higher is better. Refer to Table 3.13 for numeric results and optimized hyperparameters.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	mse $\times 10^2$	std dev	ρ	C	mse $\times 10^2$	std dev	ρ	γ	mse $\times 10^2$	std dev	C	mse $\times 10^2$	std dev
1	0.100	2.57	0.29	0.0316	0.01	11.32	1.46	-	-	-	-	0.0100	4.48	0.83
2	1.000	0.94	0.13	0.100	0.01	12.35	1.87	0.316	0.3	2.17	0.25	0.0100	2.35	0.16
3	3.162	0.74	0.10	0.3162	0.10	1.54	0.04	1.000	0.1	1.26	0.04	0.0316	1.97	0.13
4	3.162	0.76	0.02	1.000	0.10	1.37	0.10	1.000	0.3	1.05	0.10	0.0316	1.76	0.10

Table 3.9: FashionMNIST, $k = 9$, best `NoDP` for $\rho = 0.003162$.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	acc	std dev	ρ	C	acc	std dev	ρ^*	γ	acc	std dev	C	acc	std dev
3	0.1	85.70	0.15	1.000	0.0100	84.23	0.29	3.162	0.5	83.67	0.48	0.0100	85.82	0.11
5	0.1	86.12	0.05	0.316	0.0316	85.27	0.25	3.162	0.5	84.49	0.31	0.0100	86.13	0.05
7	0.1	86.34	0.12	0.100	0.1000	85.59	0.23	3.162	0.7	85.19	0.17	0.0100	86.33	0.03
9	0.1	86.43	0.08	0.316	0.0316	85.74	0.20	3.162	0.7	85.65	0.15	0.0316	86.43	0.05

Table 3.10: AG News, $k = 9$, * values are scaled $\times 10^3$, best `NoDP` for $\rho = 0.003162$.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	acc	std dev	$\rho*$	C	acc	std dev	ρ^*	γ	acc	std dev	C	acc	std dev
3	0.316	91.19	0.51	3162.0	0.01	89.2	0.39	3.162	0.5	87.91	0.62	10.0	91.27	0.46
5	0.316	91.66	0.50	3.162	10.00	90.9	0.49	3.162	0.7	90.71	0.69	10.0	92.16	0.38
7	3.162	92.96	0.90	3.162	10.00	91.4	0.43	3.162	0.7	91.16	0.49	10.0	92.76	0.52
9	3.162	94.11	0.63	3.162	10.00	91.7	0.42	3.162	0.7	91.21	0.47	10.0	93.22	0.70

Table 3.11: MNIST, $k = 5$, * values are scaled $\times 10^3$, best `NoDP` for $\rho = 0.003162$.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	mse $\times 10^2$	std dev	ρ	C	mse $\times 10^2$	std dev	ρ	γ	mse $\times 10^2$	std dev	C	mse $\times 10^2$	std dev
3	0.316	1.56	0.10	0.0316	0.01	12.88	2.37	-	-	-	-	0.01	3.43	0.26
5	3.162	0.77	0.05	3.162	0.01	1.44	0.08	0.316	0.3	1.71	0.08	0.01	2.09	0.18
7	3.162	0.67	0.09	0.3162	0.10	1.54	0.10	3.162	0.1	1.06	0.03	0.01	1.81	0.18
9	3.162	0.64	0.07	31.62	0.01	1.25	0.14	3.162	0.1	0.84	0.08	0.01	1.51	0.11

Table 3.12: Fashion MNIST, $k = 5$, best `NoDP` for $\rho = 0.003162$.

	Online			Fixed Threshold				Fixed Quantile				Adam WOSM		
ε	ρ	acc	std dev	ρ	C	acc	std dev	ρ	γ	acc	std dev	C	acc	std dev
3	0.0316	85.82	0.18	3.162	0.01	84.37	0.25	3.162	0.5	84.32	0.34	0.01	86.07	0.05
5	0.3162	86.12	0.11	3.162	0.01	85.49	0.10	3.162	0.7	85.20	0.17	0.01	86.33	0.03
7	0.3162	86.32	0.09	3.162	0.01	85.97	0.07	3.162	0.7	85.71	0.18	0.01	86.51	0.06
9	0.3162	86.38	0.12	3.162	0.01	86.18	0.07	3.162	0.7	85.93	0.18	0.01	86.57	0.11

Table 3.13: AG News, $k = 5$, best `NoDP` for $\rho = 0.003162$.

Chapter 4

Foregoing sensitivity with membership inference attacks

Differentially Private Machine Learning (DPML) is a fast-paced research field, leading to foundational improvements to the algorithms currently used to provide mathematical privacy guarantees to the training of machine learning models. Still, although the number of practical use cases is growing and research is edging closer to industry, some key challenges prevent, for the time being, a large-scale adoption of DPML practices. This is especially true for reasonably large models, which are a primary building block of the current AI landscape. Consider, for instance, that ImageNet-scale image classification with DP is roughly at 47.9% accuracy [Kurakin et al., 2022], while comparable non-DP models easily reach 75% and more modern architectures surpass 90% [Dai et al., 2021]. The introduction of DP noise at training time represents a limiting factor that worsens with the size of the model. The issue is even more exacerbated in Large Language Models (LLMs) which have become crucial in various text-centric tasks by leveraging transformer-based architectures. Nevertheless, we recall that, as anticipated in Chapter 1, training machine learning models already assumes a degree of randomness in the dataset collection, in the choice of the parameters, and the whole training procedure, to name a few, that in principle may already provide some degree of privacy by randomness. Without a complete characterization of the sources of randomness though, we cannot derive *a priori* privacy guarantees in DP terms. Still, we can evaluate the indistinguishability of the resulting training procedure *a posteriori* in terms of empirical Differential Privacy (eDP). In brief, eDP tests the model at inference time (e.g. against membership inference attacks) to derive a lower bound on the privacy leakage ε of DP. The advantage is clear: with-

out enforcing DP at every step of the optimization, we can get rid of both the injection of noise and the selection of the optimal sensitivity level. The research contained in this chapter is part of the following work:

- Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against LLMs. In *(Under Review) The Fifth Workshop on Privacy in Natural Language Processing (in Conjuction with ACL 2024)*, 2024a

4.1 Introduction

Advances in machine learning and natural language processing with transformer-based architectures [Vaswani et al., 2017] have increasingly started permeating today's digital landscape and large language models (LLMs) [Radford et al., 2018, 2019; Devlin et al., 2018; Brown et al., 2020] have become integral to many tasks involving text understanding and generation [Sanh et al., 2021; Wei et al., 2022]. However, this technological advancement comes with its challenges and ethical implications as LLMs have raised concerns related to biases [Narayanan Venkit et al., 2023], privacy breaches [Carlini et al., 2021], and model vulnerabilities [Wallace et al., 2021]. One significant issue that emerged in the context of LLMs is the need to protect user privacy and safeguard sensitive information [Lehman et al., 2021]. LLMs are typically trained on vast datasets, which may include personal and sensitive data, posing risks of privacy leakage. These concerns have prompted the adoption of regulatory frameworks and responsible AI principles aimed at ensuring the responsible development of artificial intelligence (AI) technologies. Initiatives such as the General Data Protection Regulation (GDPR) [European Parliament, European Council, 2016] and the California Consumer Privacy Act (CCPA) [State of California, 2018] have set stringent guidelines for handling personal data, including data processed by LLMs. To comply with these regulations and to provide a safe interaction with these technologies, privacy attacks on language models prior to their deployment or public release may prove to be useful auditing mechanisms to ensure that LLMs do not violate the users' rights to data privacy.

Membership inference attacks (MIA) [Shokri et al., 2017] are considered one of such tools. MIAs are a category of privacy attacks that aim to determine whether a specific data point was included in the training dataset used to optimize a machine learning model, by analyzing its output. A successful attack may potentially lead to significant privacy breaches and data

leakage, directly or as a stepping stone to achieving more powerful attacks, given the adversary's newly acquired knowledge about the targeted individual. This attack's success hinges on machine learning models' tendency to be overly confident with familiar training data [Carlini et al., 2019]. By leveraging calibration strategies, accuracy in MIAs can be enhanced without knowledge of specific training samples. Training shadow models mimicking the target model aids in distinguishing between easily predictable non-training samples and those used during model optimization [Shokri et al., 2017; Carlini et al.; Watson et al., 2021]. The effectiveness of this strategy degrades when we deviate from the assumption of knowledge of the training distribution or when the number of shadow models is limited [Carlini et al.]. Naturally, this may incur a large computational cost that may be an obstacle to the practical adoption of effective MIAs for privacy auditing, especially with the ever-increasing size of both models and training datasets [Kaplan et al., 2020]. Therefore, this Chapter will provide the following contributions:

- We will explore membership inference attacks from the standpoint of a privacy auditor
- We will introduce a computationally efficient calibration strategy for MIAs
- We will empirically assess its potential as a substitute for other prevalent strategies

4.2 Background

Auto-regressive transformer-based LLMs are language models that output a probability distribution over their dictionary, conditioned on an input sequence of words that has been tokenized and turned into numerical inputs via an embedding layer, which maps the index of a token in the dictionary to a dense representation that may be learned at training time [Radford et al., 2018, 2019] or adopted from publicly distributed *word embeddings* [Devlin et al., 2018]. For a model f with input sequence x, we define $\mathbb{P}[w|x] = f_w(x)$ as the conditional probability that the token following x is w, and we have that $\sum_w f_w(x) = 1$ when we iterate w over the dictionary of tokens. LLMs are typically trained on large datasets of text to minimize a measure of surprise in seeing the next token. This measure function is called *perplexity* and for a sequence x it is defined as the average negative

log-likelihood of a sequence:

$$ppx(f, x) = -\frac{1}{|x|}\sum_{t=1}^{|x|}\log(f_{x_t}(x_{<t})) \tag{4.1}$$

with $|x|$ the number of tokens in the sequence.

Membership inference attacks [Shokri et al., 2017; Watson et al., 2021; Carlini et al.] aim to determine whether a particular data record x was used in the training dataset D_{train} of a machine learning model. These methods leverage model outputs like confidence scores or prediction probabilities to compute a score for the targeted sample. For LLMs, the typical assumption is to grant the adversary access to the output probabilities $f(x)$, which may be used to estimate the perplexity on the targeted samples as a score. Given a sample x, the goal of the attacker is to learn a thresholding classifier to output 1 when the perplexity is lower than a certain value γ:

$$A_\gamma(f, x) = \mathbb{1}[ppx(f, x) < \gamma] \tag{4.2}$$

MIA is a simple and effective tool to measure the privacy risk in a trained machine learning model and it has interesting connections with other privacy frameworks. In particular, it is known to have a success rate bounded by the privacy parameters of Differential Privacy (DP) [Dwork et al., 2006]. A randomized mechanism $\mathcal{M}$ is said to be ε-DP if for any two datasets D, D' that differ in at most one sample, and for any $R \subseteq \text{range}(\mathcal{M})$, we have that:

$$\mathbb{P}[\mathcal{M}(D) \in R] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(D') \in R] \tag{4.3}$$

Notably, DP quantifies the worst-case scenario of the privacy risk, and it is therefore a fundamental tool in privacy assessment. From the performance of the thresholding classifier $\tilde{A}_\gamma(f, x)$ one can obtain a lower bound to the *empirical* ε-DP [Kairouz et al.]:

$$e^{\varepsilon} \geq \frac{TPR}{FPR} \tag{4.4}$$

with TPR and FPR being, respectively, the true and false positive rates, given a certain threshold.

4.3 Related works

Privacy attacks against language models is an active area of research and different refinements have been proposed. Some works have focused on an

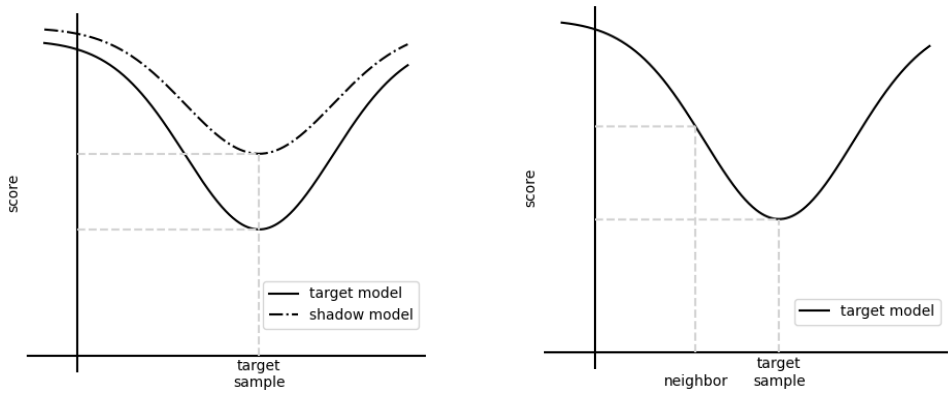
attacker where data poisoning is allowed, granting the adversary write access to the training dataset, to increase memorization [Tramèr et al., 2022] or in general to induce malicious behaviors [Xu et al., 2023; Wallace et al., 2021; Yan et al., 2023; Shu et al., 2024; Huang et al., 2020] and improve property inference attacks [Mahloujifar et al., 2022]. Other works have adopted similar techniques to achieve actual training data extraction from the training set, with only query access to the trained model [Carlini et al., 2021, 2023].

In the context of MIAs with query access to the target model, most research focused on strategies to improve the calibration of the per-sample scores, i.e. techniques to improve the precision and recall in distinguishing members from non-members of the training set. In principle, if we can assert that an out-of-distribution non-member of the training set will induce a high perplexity in a target LLM, there are a number of scenarios where the distinction is not as clear cut and a thresholding classifier essentially ends up distinguishing between in-distribution and out-of-distribution samples. A refined MIA then employs calibration strategies to tune the scoring function based on the difficulty of classifying the specific sample, as in [Watson et al., 2021]. Thus, a relative membership score is obtained by comparing $f(x)$ with one of two results based on whether the adversary is assumed to have access to *neighboring models* $\tilde{f}(x)$ [Carlini et al.; Watson et al., 2021] or *neighboring samples* $f(\tilde{x})$ [Mattern et al., 2023]. Regardless of the specifics of the calibration strategies, we can define the new classifier as

$$\tilde{A}_\gamma(f, x) = \mathbb{1}[ppx(f, x) - p\tilde{p}x(f, x) < \gamma] \tag{4.5}$$

where $p\tilde{p}x(f, x)$ is the calibrated score over a set of neighboring models $ppx(\tilde{f}, x)$ or over a set of neighboring samples $ppx(f, \tilde{x})$. For a visual understanding of the two scenarios, refer to Figure 4.1.

Neighboring models can be obtained by an adversary who is assumed to have some degree of knowledge of the training data distribution and training a number of shadow models to mimic the behavior of the target LLM. For instance [Carlini et al.] trains multiple instances of the same architecture on different partitions of the training set, [Carlini et al., 2021] uses smaller architectures trained on roughly the same data, [Watson et al., 2021] leverages catastrophic forgetting of the target model under the assumption of white-box access. Neighboring samples do not require this assumption nor additional training and only need a strategy to craft inputs that are similar to the target sample under a certain distance metric. For instance, [Mattern et al., 2023] crafts neighboring sentences by swapping a number of words with their synonyms, showing good results but limited to



(a) (b)

Figure 4.1: An intuitive comparison of computing the score with different calibration strategies: using a shadow model (a) and using a neighboring sample (b).

the case of an adversary with limited knowledge of the training data distribution. The authors then base the neighboring relationship on the *semantic* space, which is hard to quantify and fix, resulting in the need to generate a large number of neighbors to reduce the effects of these random fluctuations. Additionally, we emphasize how [Mattern et al., 2023] requires the use of an additional BERT-like model to generate synonyms, thus increasing the computational and memory cost of the attack. In [Tramèr et al., 2022] instead, calibration is done by comparing scores of the true inputs with scores of the lower-cased inputs. These strategies are known to be under-performing when knowledge of the training distribution is available, and are therefore proposed as an effective calibration mechanism when training shadow models is not possible.

4.4 Method

The intuition behind the use of neighboring samples for score calibration is rather simple: if a model is over-confident in a specific data point because it was trained on it, we would see a prominent dip in the perplexity compared to data points that are only marginally distant from that sample. Conversely, for samples not seen at training time, we would expect roughly the same perplexity for the target sample and its neighbors. Considering that we are then thresholding this difference, it is of utter importance to fix the distance at which we evaluate the neighbors.

Noisy neighbors If we describe a language model as a composition of layers $f(x) = g(e(x))$ where e is an embedding layer and g is the rest of the network, one can artificially create neighbors in the n-dimensional embedding space by directly injecting random noise at the output of $e(x)$. In particular, if we create noisy neighbors by injecting Gaussian noise such that

$$f(x'_\sigma) = g(e(x) + \rho), \quad \text{with } \rho \sim \mathcal{N}(0, \sigma I_n) \tag{4.6}$$

we will have that the Euclidean distance between the true and randomized input in the embedding space will be

$$\mathbb{E}[\|e(x) - e(x) - \rho\|] = \mathbb{E}[\|\rho\|] = \sigma\sqrt{n} \tag{4.7}$$

thus fixing, in expectation, the distance from the true sample at which the perplexity of the models will be evaluated. In practice, this step requires generating multiple neighbors for each sample, to mitigate the inherent randomness of using stochastic noise, which effectively only amounts to running the LLM inference multiple times.

The choice of the standard deviation σ is an additional free parameter that may lead to thinking of a challenging search, introducing a large number of queries to the model. Instead, we find the performance of the strategy as a function of σ does i) present a prominent peak corresponding to the optimal value, and ii) this optimal value can be found with a binary search, as shown in Figure 4.2.

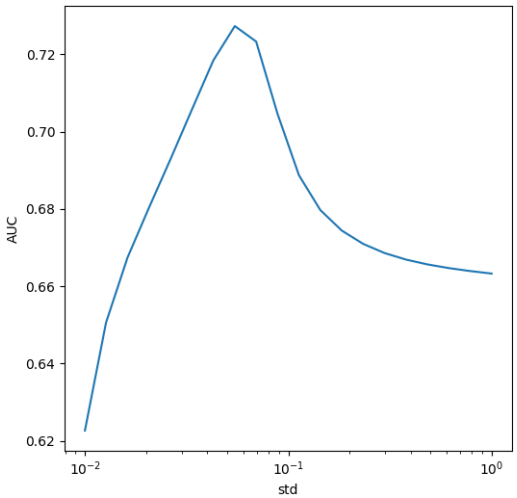


Figure 4.2: The AUC of the thresholding classifier for MIA shows a single and prominent peak at the optimal σ value in the *noisy neighbors* strategy.

We emphasize the challenge of isolating the embedding layer from the remainder of the network in an LLM when considering a scenario where an attacker has only black box access to the model. However, when this limitation doesn't apply, we think it is still within the capacity of an auditor to utilize a slightly stronger attacker model, where the first embedding layer is exposed, to save computational resources in simulating an adversary without access to the model architecture. Most importantly, in fact, we are inclined to explore this option as a more computationally efficient substitute for training shadow models for calibration, particularly in the context of auditing, rather than viewing it as a novel, realistic attack. Thus, we depart from other MIA research using neighboring samples in that we do not make limiting assumptions about the auditor's access to the training data distribution, but we do not leverage this knowledge for computational efficiency.

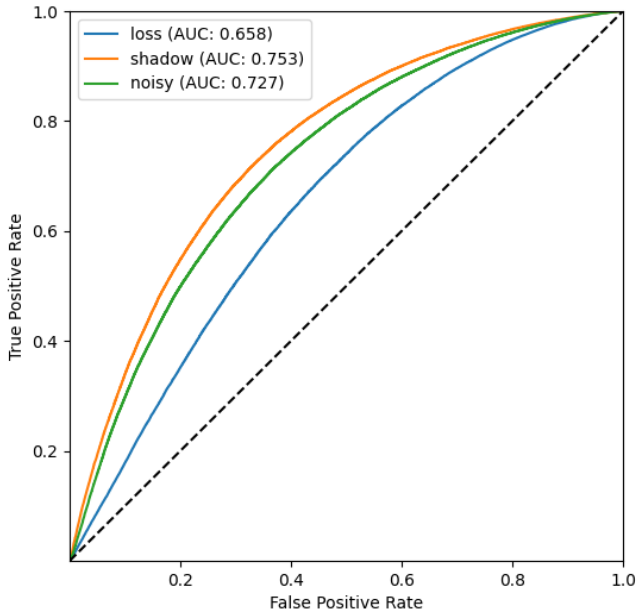
4.5 Experiments

To validate the noisy neighbor strategy in implementing a calibrated MIA, we run a series of preliminary experiments on an LLM to gauge the risk of memorization of training data. The chosen architecture is GPT-2 [Radford et al., 2019] which offers ample learning capabilities while at the same time being a model with a relatively small memory and computational footprint by today's standards, which allows for easier and faster prototyping. The model was pre-trained on OpenWebText [Gokaslan and Cohen, 2019], an open reproduction of the undisclosed WebText in [Radford et al., 2019]. The model was then fine-tuned on 60% of the full WikiText corpus [Merity et al., 2016], a large collection of Wikipedia articles. The same data split was then partitioned into 10 subsets used to train 10 shadow models for score calibration, as in [Carlini et al.]. Note that Wikipedia articles are filtered out of the OpenWebtext corpus, to avoid data leakage in common benchmarks, such as ours. The remaining portion of 40% of WikiText is thus used as a source of non-member, 126-token long samples to analyze the performance of the attack. We generate only 10 synthetic neighbors for each sample. Given a sample and its score, the thresholding classifier yields a binary decision on whether it was part of the training dataset or not. To determine how good the best possible classifier may be, we need to evaluate its accuracy at different thresholds. As is common for binary classification problems, though, the accuracy does not give a complete picture of the confidence at which the classifier can tell apart members and non-members of the dataset. Thus Figure 4.3a shows the complete

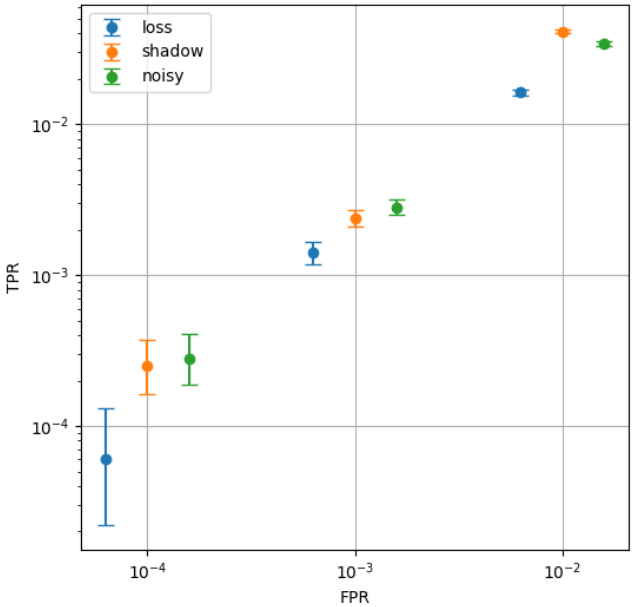
range of TPRs versus FPRs for the three main strategies we included in this comparison: score by perplexity (*loss*), by shadow model calibration (*shadow*), and by noisy neighbor (*noisy*) calibration. We have opted not to incorporate the *lowercasing* strategy [Tramèr et al., 2022] and the *semantic neighbor* approach [Mattern et al., 2023] in our study. These methods have, however, shown lower performance levels when information about the training data distribution is accessible, which is contemplated from the auditor's point of view. Additionally, concerning the findings of [Mattern et al., 2023], we encountered challenges in effectively reproducing some of their results. This difficulty may stem from the synonym generation technique outlined in [Zhou et al., 2019], which may not be fully comprehensive. Figure 4.3a also notes the Area Under the Curve (AUC), which for *noisy* and *shadow* amounts to 0.727 and 0.753 respectively, thus showing a discrepancy of only $\sim 3.4\%$. The AUC is an important metric for binary classifiers as it abstracts from the specific threshold, thus giving an average case idea of the strength of the attacker. Still, as highlighted in [Carlini et al.], special care should be given to what happens at low FPRs, that is when the attacker can confidently recognize members of the training set. This is what Figure 4.3b focuses on, again showing a strong overlap of the *shadow* and *noisy* strategies. Following Equation 4.4, we also provide the perspective of empirical DP, as the privacy community pushes to adopt this framework to comply with regulatory frameworks such as the GDPR [Cummings and Desai, 2018]. Empirical DP measures the extent to which individual data points can be inferred or re-identified from the output of the system, and contrary to DP, it is a *post-hoc* measurement, not an *a-priori* guarantee. Figure 4.4 reports the results, where we see a strong consistency between the *noisy* and *shadow* strategies, especially for FPRs lower than 10^{-2}.

4.6 Conclusion

This chapter set out to elaborate an efficient strategy for membership inference attacks. Past research in the area focused on improving the strength of the attacker, especially under the black-box access model granted to the attacker. Conversely, we develop a technique to try and match the efficacy of state-of-the-art strategies, while reducing the computational burden for an auditor trying to assess the privacy risk of exposing the query access to a trained LLM. In particular, MIAs are typically improved with the use of shadow models, that need training on a data distribution similar to that of the target model training set. We propose the use of noise injection in the



(a) ROC curve of the MIA classifier.



(b) Performance of the attacker at low FPRs.

Figure 4.3: Efficacy of different strategies for MIA.

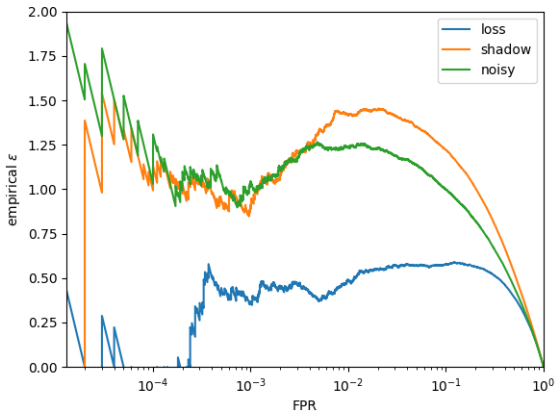


Figure 4.4: Empirical differential privacy measured downstream of training.

embedding space of the LLM to create synthetic neighbors of the targeted sample, to shift the comparison from the perplexity scored by different models on one sample, to the comparison of different samples by the same model. This approach allows to only use the model in inference mode, thus inherently reducing the time and cost of running an MIA. With several experiments we assess how our strategy results converge to the results of using shadow models, showing remarkable alignment. This is also looked at from the perspective of empirical differential privacy, which is considered a modern tool for assessing privacy risks and regulatory compliance. Within the general context of the thesis, we highlight how it allows us to connect membership inference attacks with another take on the issue of selecting the clipping threshold: by considering a post-training assessment instead of pre-training guarantees, the whole problem is entirely avoided.

Chapter 5

Adaptive sensitivity in the bounded model for differential privacy

Moving towards the remaining considerations on the topic of sensitivity in differential privacy, at least for what concerns this thesis, we reason on its definition, the role of the adversary, and the promise we make to the individual who decides to take part in a dataset. In particular, we highlight how most DPML research focuses on the "addition/removal" of a record from the database to define the sensitivity, and thus the tuning of the sanitization noise and following privacy guarantees. We argue here that defining sensitivity with respect to the "replacement" of a record, as was the initial definition of indistinguishability, may play a favourable role in certain machine learning settings. Therefore, we discuss how to switch from one definition to the other, what the implications are in terms of sensitivity and sanitization noise and eventually, highlight what scenarios may benefit from using either definition. We present this research as a constructive critique of the current DMPL research, but we acknowledge limitations in our position as well. Mostly, although we can provide a theoretical justification and preliminary results for this position, a thorough experimental evaluation is still to be carried out. Part of the following chapter was presented as:

- Filippo Galli, Saya Biswas, Kangsoo Jung, Catuscia Palamidessi, and Tommaso Cucinotta. On the adaptive sensitivity of differentially private machine learning. In *The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence (in Conjuction with AAAI 2023)*, 2023a

5.1 Introduction

We have thoroughly discussed differential privacy throughout this document, providing its definition in Chapter 1, along with the definition of one of its generalizations in Chapter 2. However, given our focus on the fine-grained details of its terminology and the specific representation of datasets upon which sensitivity is defined, we find it beneficial to revisit these definitions here. Let us define the domain of databases of cardinality n as $\mathcal{D}^n$ and call the mechanism $\mathcal{M}$ the random function querying the elements (that is the databases) ${}_mD \in \mathcal{D}^n$. Let us refer to this notation for databases as the multiset representation, whence we have the left subscript m in ${}_mD$. Thus, we can introduce the following definition from [Dwork et al., 2006], slightly modified to accommodate for approximate (ε, δ) differential privacy[1].

Definition 10 (Bounded model of (ε, δ)-DP [Dwork et al., 2006])**.** *A random mechanism $\mathcal{M}$ is (ε, δ) differentially private if for all pairs of datasets ${}_mD, {}_mD' \in \mathcal{D}^n$ which differ in only 1 entry, and for all $T \subseteq$ Range$(\mathcal{M})$:*

$$\mathbb{P}\left[\mathcal{M}({}_mD) \in T\right] \leq e^{\varepsilon}\mathbb{P}\left[\mathcal{M}({}_mD') \in T\right] + \delta \tag{5.1}$$

Definition 10 for (ε, δ)-DP, although being called *indistinguishability* in [Dwork et al., 2006], is commonly taken as coinciding with that of differential privacy. This is especially true in what is known today in the community as the *bounded model* for differential privacy [Kifer and Machanavajjhala, 2011], where "bounded" comes from including the cardinality of the dataset in the definition. Multiple instances of works focused on this framework, notably those on the shuffle model for DP [Feldman et al., 2020; Balle et al., 2019; Koskela et al., 2021a], which all assume a constant cardinality of the databases involved in the definition.

In subsequent work, [Dwork et al., 2014a] defined the role of databases in terms of their histogram representation. Let $\mathcal{X}$ be the universe of all possible records with a given ordering. Then ${}_hD \in \mathbb{N}^{|\mathcal{X}|}$ is the histogram representation of a database[2], where ${}_hD(x)$ is the number of entries of type $x \in \mathcal{X}$. We can, then, introduce the definition of ℓ_1 distance between databases as:

Definition 11 (ℓ_1 distance [Dwork et al., 2014a])**.** *The ℓ_1 norm of a database*

[1] Historically, DP meant pure differential privacy ($\varepsilon = 0$) when it was first introduced.

[2] Where we redefine $\mathbb{N} \triangleq \mathbb{N} \cup \{0\}$

${}_hD \in \mathbb{N}^{|\mathcal{X}|}$ is denoted with $\|{}_hD\|_1$ and is defined as

$$\|{}_hD\|_1 = \sum_{x \in \mathcal{X}} |{}_hD(x)| \tag{5.2}$$

The ℓ_1 distance between two databases ${}_hD, {}_hD'$ is $\|{}_hD - {}_hD'\|_1$.

The corresponding definition of differential privacy in [Dwork et al., 2014a] for algorithms querying databases in their histogram representation can be reported as follows:

Definition 12 (Unbounded model for (ε, δ)-DP [Dwork et al., 2014a])**.** *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ) differentially private if for all $T \subseteq$ Range$(\mathcal{M})$ and for all adjecent ${}_hD, {}_hD' \in \mathbb{N}^{|\mathcal{X}|}$ (i.e. such that $\|{}_hD - {}_hD'\| \leq 1$):*

$$\mathbb{P}\left[\mathcal{M}({}_hD) \in T\right] \leq e^{\varepsilon} \mathbb{P}\left[\mathcal{M}({}_hD') \in T\right] + \delta \tag{5.3}$$

One might argue that the presence of n in Equation (5.1) of Definition 10 implies that the adversary knows the number of elements (rows) in the datasets considered in the bounded model of differential privacy. This is unlike the unbounded model, which may lead us to believe that the latter provides a stronger privacy guarantee. To address this issue, we propose an alternative, more general definition of DP as follows.

Definition 13 ((ε, δ)-DP for attribute inference attacks)**.** *Let $\mathcal{D} = \{{}_hD : {}_hD \in \mathbb{N}^{|\mathcal{X}|}\}$ be the space of all datasets (represented as histograms whose entries are from the domain $\mathcal{X}$). Any pair of datasets ${}_hD, {}_hD' \in \mathcal{D}$ are* adjacent*, denoted as ${}_hD \simeq {}_hD'$, if either:*

- *${}_hD = {}_hD'$, or*
- *$\exists!\ x_1, x_2 \in \mathcal{X}$ s.t.*
  - *${}_hD(x_1) - {}_hD'(x_1) = 1$ and*
  - *${}_hD(x_2) - {}_hD'(x_2) = -1$*

A randomizing algorithm $\mathcal{M}$ is (ε, δ) differentially private if, for all pairs of adjacent datasets ${}_hD,\ {}_hD' \in \mathcal{D}$ and for any $T \subseteq \mathrm{Range}(\mathcal{M})$, we have:

$$\mathbb{P}\left[\mathcal{M}({}_hD) \in T\right] \leq e^{\varepsilon} \mathbb{P}\left[\mathcal{M}({}_hD') \in T\right] + \delta \tag{5.4}$$

Note that Definition 13 is in the same spirit as the bounded model of DP (i.e. adjacency of datasets implies they are the same size) with the difference that in this case there is no explicit bound and, thus, the size of the dataset is not known to the attacker. In particular, the case of ${}_hD \simeq {}_hD'$ with $\sum_{x\in\mathcal{X}} {}_hD(x) = \sum_{x\in\mathcal{X}} {}_hD'(x) = n$ for some fixed $n \in \mathbb{N}$ represents the bounded model of DP as in Definition 10. To aid the reader in dealing with the new notational elements of this Chapter, Table 5.1 includes a summary of the symbols used here and their meaning.

Table 5.1: Table of notations for Chapter 5

Notation	Description
$x \in \mathcal{X}$	Element of a database D
${}_mD \in \mathcal{D}^n$	Database D of size n in multiset representation
${}_hD \in \mathbb{N}^{\lvert\mathcal{X}\rvert}$	Database D in histogram representation
${}_hD(x)$	Number of elements of type $x \in \mathcal{X}$ in database ${}_hD$
${}_mD_i$	The i-th element of the multiset ${}_mD$
$D \simeq D'$	Adjacent databases
${}_mS_2(f)$	ℓ_2 sensitivity for a function f of database ${}_mD$
${}_hS_2(f)$	ℓ_2 sensitivity for a function f of database ${}_hD$

5.2 Sensitivity in the bounded and unbounded models of differential privacy

A legitimate question would be: Are Definition 10 and Definition 12 equivalent? To provide an answer, we can consider the databases ${}_mD, {}_mD' \in \mathcal{D}^n$ differing in at most 1 entry and translate them into their histogram representation ${}_hD, {}_hD' \in \mathbb{N}^{\lvert\mathcal{X}\rvert}$. We know they have at least $n-1$ records in common, which leaves us with the n^{th} records ${}_hD_n, {}_hD'_n$ be of any two (possibly different) types from $\mathcal{X}$. This gives, in general, $\|{}_hD - {}_hD'\|_1 \leq 2$. Therefore, when we are using Definition 10 we are considering privacy guarantees, with parameters (ε, δ), that are valid for databases ${}_hD, {}_hD'$ at a distance of up to 2 on the Hamming graph of their adjacency relation [Chatzikokolakis et al., 2013]. The representation of the databases under Definition 10 and Definition 12 has been illustrated in Figure 5.1.

Sanitizing a deterministic query with a DP mechanism requires knowledge of the sensitivity of the query itself, as we extensively discussed. Recall additive mechanisms for (ε, δ) differential privacy use the ℓ_2 sensitivity of a function to tune the variance of the random noise distribution, refer to

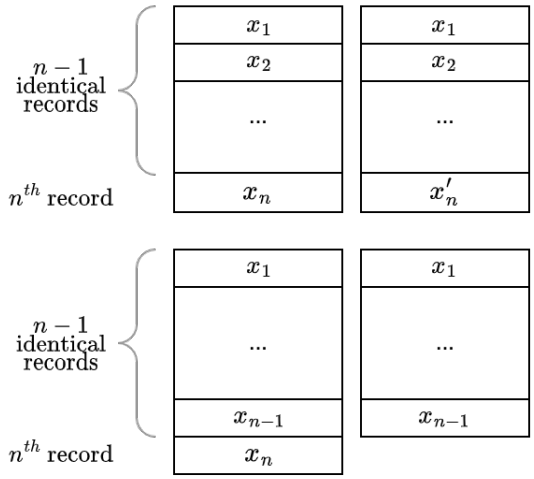


Figure 5.1: Indistinguishability requirements are defined for different pairs of databases: in the upper half according to Definition 10, and at the bottom according to Definition 12

Chapter 1 for a broader discussion. Let $f : \mathcal{D}^n \to \mathbb{R}^d$ and $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^d$ be the deterministic function under consideration, where we overload the definition of f depending on its argument. The ℓ_2 sensitivity is defined on corresponding pairs of neighboring databases. Therefore, introducing the terms ${}_mS_2(f)$ and ${}_hS_2(f)$ to denote the corresponding ℓ_2 sensitivities of a function f, applied to either the histogram or multiset representations of its input, we have that:

$$
\begin{aligned}
{}_mS_2(f) &= \max_{\substack{{}_mD,{}_mD' \in \mathcal{D}^n \\ {}_mD \simeq_m D'}} \|f({}_mD) - f({}_mD')\|_2 \\
&= \max_{\substack{{}_hD,{}_hD' \in \mathbb{N}^{|\mathcal{X}|} \\ \|{}_hD - {}_hD'\|_1 \leq 2}} \|f({}_hD) - f({}_hD')\|_2 \\
&\leq \max_{\substack{{}_hD,{}_hD',{}_hD'' \in \mathbb{N}^{|\mathcal{X}|} \\ \|{}_hD - {}_hD''\|_1 \leq 1 \\ \|{}_hD'' - {}_hD'\|_1 \leq 1}} \|f({}_hD) - f({}_hD'')\|_2 + \|f({}_hD'') - f({}_hD')\|_2 \\
&= 2 \max_{\substack{{}_hD,{}_hD'' \in \mathbb{N}^{|\mathcal{X}|} \\ \|{}_hD - {}_hD''\|_1 \leq 1}} \|f({}_hD) - f({}_hD'')\|_2 \\
&= 2 \max_{\substack{{}_hD,{}_hD'' \in \mathbb{N}^{|\mathcal{X}|} \\ {}_hD \simeq {}_hD''}} \|f({}_hD) - f({}_hD'')\|_2 \\
&= 2 \cdot {}_hS_2(f)
\end{aligned}
\tag{5.5}
$$

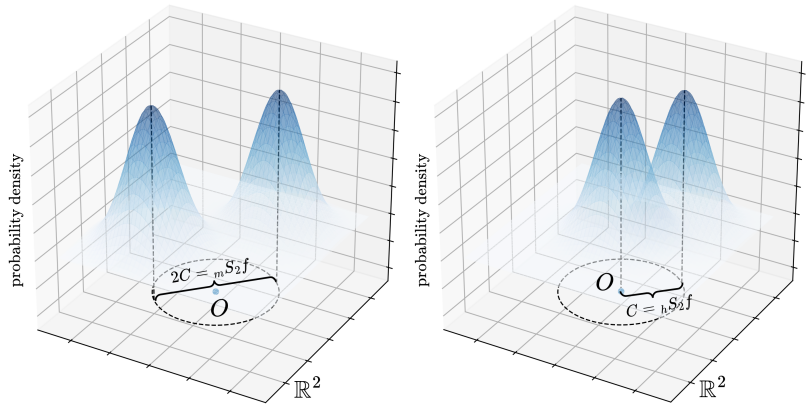
Thus, we see that mechanisms for sanitizing a query with the same

privacy parameters satisfying Definition 10 may require up to twice the amount of noise (in terms of variance of the random distribution) needed to satisfy Definition 12. Let us now take the special case of ${}_mS_2f = 2{}_hS_2f$. For instance, from the Gaussian mechanism introduced in 1 we know that $\mathcal{N}(\mu = 0, \sigma^2 = \frac{2\ln(1.25/\delta){}_mS_2f^2}{\epsilon^2})$ will give (ε, δ)-DP in Definition 10 and imply $(\varepsilon/2, \delta)$-DP in Definition 12, since $\mathcal{N}(\mu = 0, \sigma^2 = \frac{2\ln(1.25/\delta){}_mS_2f^2}{\epsilon^2}) = \mathcal{N}(\mu = 0, \sigma^2 = \frac{2\ln(1.25/\delta){}_hS_2f^2}{(\epsilon/2)^2})$. Therefore, under the assumptions that ${}_mS_2f = 2{}_hS_2f$, we can not only directly compare the two definitions, but also derive the corresponding privacy parameters.

5.3 The unbounded model of DP in machine learning

The authors in [Abadi et al., 2016], whose work we have frequently discussed in this thesis, use Definition 12 of differential privacy to sanitize the gradient-based optimization of the parameter vector $\theta \in \mathbb{R}^d$. Instead of computing the average gradient on some database ${}_hD$, `DP-SGD` first computes the "sum of gradients", and by clipping the 2-norm of each individual addend to C, they reduce the ℓ_2 sensitivity of the sum of gradients to be ${}_hS_2f = C$. Considering the first $n-1$ contributions to the sum of gradients are identical whether it is computed on ${}_hD$ or ${}_hD'$, with ${}_hD \simeq {}_hD'$, we can see the indistinguishability parameters are to be evaluated as the ratio between the p.d.f.'s of a Gaussian centered in 0 and a Gaussian centered in any point on the surface of the ball of radius C in $\mathbb{R}^d$, with d the dimension of θ. Figure 5.2b shows an example of the above for the case with $d = 2$. Conversely, Figure 5.2a shows what would happen, in terms of indistinguishability, if we were to use the same Gaussian mechanism with the same variance to sanitize according to Definition 10. Being exactly in the special case with ${}_hS_2f = {}_mS_2f/2$ we would end up with twice the ε.

In a similar fashion to centralized machine learning, Federated Learning (FL) is a distributed learning paradigm where a set of clients, each holding private data, trains a global machine learning model collaboratively by exchanging a series of local optimization updates. See Chapter 2 for a more extensive introduction on the topic. The work in [Konečny et al., 2016b; McMahan et al., 2017a] introduced the `FedAvg` algorithm where the server broadcasts the current global model parameters $\theta^{(t)}$ to the clients involved in the training, who perform local optimization over their private data and communicate back the local update vector. These vectors are then averaged by the server and used to update the global model. Al-



(a) (b)

Figure 5.2: Normal distributions with the same variance yield different indistinguishability parameters when used in Definition 10 (a) and Definition 12 (b), up to twice as conservative.

though in general local updates are different objects to gradients, from the point of view of the server, they behave exactly in the same way, i.e. taking the model from $\theta^{(t)}$ to $\theta^{(t+1)}$. In DP FL, too, we see Definition 10 being the prevalent choice for providing differentially private guarantees [McMahan et al., 2017b; Andrew et al., 2021]. By clipping the addends of the "sum of local updates" query at C and sanitizing it with the Gaussian mechanism, it is possible to learn global models that are differentially private with respect to the clients' local data. The choice of the clipping threshold is made *a priori* as in `DP-SGD`, and no clear strategy is put forward to define how to pick a particular value. A step in this direction is made by [Andrew et al., 2021], who introduce an adaptive clipping strategy for providing DP guarantees under the same model of adversary, in the context of Federated Learning. Still, a small value for C would slow down the optimization, whereas a large value would introduce a lot of noise, since ${}_hS_2f = C$. Concurrently optimizing for the best C is thus crucial throughout training.

The tension between the *magnitude* of the clipping threshold and the *sensitivity*, we argue here, is due to the adoption of Definition 12 for differential privacy. Consider $f : \mathcal{D}^n \to \mathbb{R}^d$ and $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^d$ to be the query "sum of gradients", each defined for the matching representation of their database inputs. Assuming, without loss or generality, that ${}_hD \in \mathbb{N}^{|\mathcal{X}|}$ has an additional record (i.e. a gradient) of type $g \in \mathcal{X}$ with respect to

${}_hD' \in \mathbb{N}^{|\mathcal{X}|}$, we find:

$$\begin{aligned} {}_hS_2f &= \max_{\substack{{}_hD,{}_hD' \in \mathbb{N}^{|\mathcal{X}|} \\ {}_hD \simeq {}_hD'}} \|f({}_hD) - f({}_hD')\|_2 \\ &= \max_{\substack{{}_hD,{}_hD' \in \mathbb{N}^{|\mathcal{X}|} \\ {}_hD \simeq {}_hD'}} \|\sum_{x \in \mathcal{X}} {}_hD(x) \cdot x - \sum_{x \in \mathcal{X}} {}_hD(x)' \cdot x\|_2 \\ &= \max_{\substack{{}_hD,{}_hD' \in \mathbb{N}^{|\mathcal{X}|} \\ {}_hD \simeq {}_hD'}} \|g\|_2 \end{aligned} \tag{5.6}$$

Otherwise assuming that ${}_mD, {}_mD' \in \mathcal{D}^n$ differ only in the n^{th} record we find:

$$\begin{aligned} {}_mS_2f &= \max_{\substack{{}_mD,{}_mD' \in \mathcal{D}^n \\ {}_mD \simeq {}_mD'}} \|f({}_mD) - f({}_mD')\|_2 \\ &= \max_{\substack{{}_mD,{}_mD' \in \mathcal{D}^n \\ {}_mD \simeq {}_mD'}} \|\sum_{j=1}^{n} {}_mD_j - \sum_{j=1}^{n} {}_mD'_j\|_2 \\ &= \max_{\substack{{}_mD,{}_mD' \in \mathcal{D}^n \\ {}_mD \simeq {}_mD'}} \|{}_mD_n - {}_mD'_n\|_2 \end{aligned} \tag{5.7}$$

The result in Equation (5.6) highlights how the only way to reduce the sensitivity in Definition 12, and in turn the additive noise, is to reduce the 2-norm of (any) gradient $g \in \mathcal{X}$, thus effectively slowing down training, as each iteration allows only small updates to the parameters θ. Conversely, Equation (5.7) shows how each gradient or local update can be as large as needed, and the sensitivity only depends on the 2-norm of the largest difference between any two vectors. Therefore, we conjecture that with the adoption of the bounded model for DP for learning differentially private global models in (federated) machine learning, on top of adaptively adjusting the (upper) clipping threshold $C^{(t)}$, one could also introduce lower and angular clipping thresholds $c^{(t)}$ and $a^{(t)}$ to the local updates, as in Figure 5.3.

At the first round of a (federated) machine learning optimization, the domain $\mathcal{X}$ of local updates can be allowed to be the full ball of radius $C^{(0)}$ in $\mathbb{R}^n$, where according to Definition 10 we would have twice the sensitivity needed in Definition 12 to satisfy (ε, δ) differential privacy with the same privacy parameters. As training progresses, though, and clients start driving the optimization in the same direction with a certain degree of consensus, the domain $\mathcal{X}$ can be forced to be reduced to only a sector of a hollow sphere, where sensitivity is greatly reduced, bounded by $C^{(t)}, c^{(t)}$ and $a^{(t)}$. We highlight the focus on federated learning and clients, instead

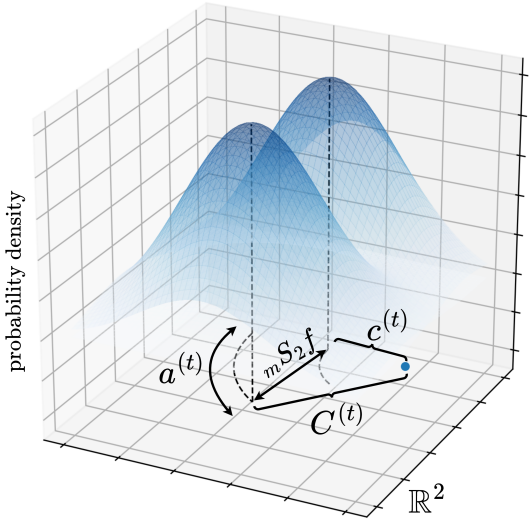
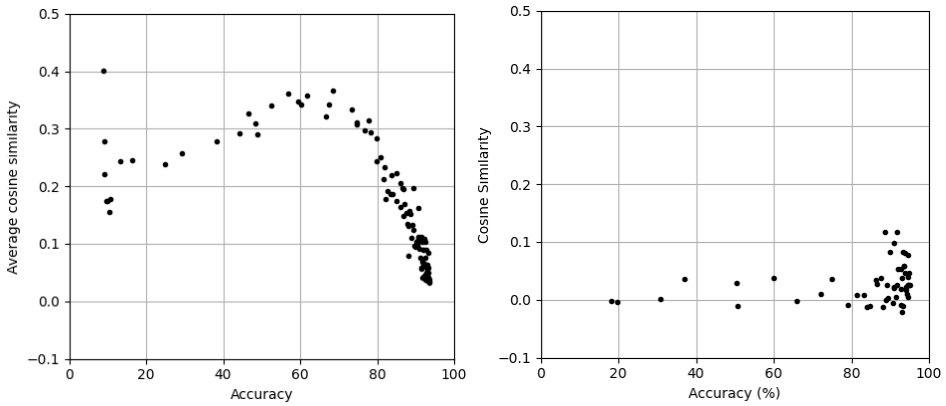


Figure 5.3: Adopting Definition 10 for differential privacy allows to adaptively reduce the sensitivity ${}_m S_2 f$ as in Equation (5.7) while not reducing the average length of gradients and local update vectors. To do so, these vectors can be clipped from above ($C^{(t)}$), below ($c^{(t)}$) and in direction ($a^{(t)}$).

of centralized machine learning and single records, as local updates usually present higher alignment as opposed to actual gradients. The reason is easily found in the nature of the two types of vectors. Although they are indeed similar objects, local updates are already essentially average gradients of the local datasets of the clients. Considering that clients can indeed have different dataset distributions, they still preserve some degree of similarity that is reflected in the local updates *roughly* pointing in the same direction, which is essential to reducing the sensitivity according to the results in Equation 5.7 and as shown in Figure 5.3. The same conditions are harder to verify in centralized machine learning, where single gradient vectors are sparser and present a less prevalent alignment. We report a simple instance of the above happening in Figure 5.4 where we train a simple CNN to fit the MNIST dataset, in a centralized and federated setting. When we evaluate the cosine similarity between local updates in a federated optimization, we see a considerably larger alignment compared to what we see in centralized training among gradients. In this specific instance, for the distributed setting, we are comparing the local updates of 20 clients, each holding a different partition of the dataset and training on the local samples for 5 steps, then followed by server aggregation before starting the next it-

eration. The centralized setting simply compares the per-sample gradient of a batch of 20 samples. Interestingly, we see diminishing agreement even in the federated setting at later stages of the optimization, suggesting that indeed the final stages focus on local dataset-specific features that are not necessarily shared with other clients.



(a) (b)

Figure 5.4: Average cosine similarity among local updates in federated learning (a) and gradients in centralized machine learning (b).

5.4 Conclusions

In this chapter, we address the problem of adaptive sensitivity in differentially private (federated) machine learning. We suggest the adoption of the bounded model for differential privacy, contrary to seminal research in the field [Andrew et al., 2021; Abadi et al., 2016]. Despite the validity of both approaches, we advocate a theoretical advantage of the former in the specific context of sanitizing sums of gradients or local updates in the multi-dimensional real set $\mathbb{R}^d$, when there is an expected degree of agreement in the direction of the local minimum among individuals. In particular, the advantage lies in the possibility of bounding the sensitivity without necessarily reducing the average 2-norm of the vectors, thus decoupling the variance of the additive noise from the expected magnitude of the parameter updates.

Chapter 6

Conclusion

Widespread data harvesting has been a major drive in the rise of ubiquitous machine learning systems, and vice-versa, as the appeal for better-performing predictive and generative models have fueled data collection at a massive scale. The use and retention of user information to assemble large datasets may come with different degrees of informed consent from the users, and regardless, the risk of privacy breaches offers a compelling opportunity for companies and academics to push the boundaries of what machines can learn while protecting private information. We have thoroughly discussed why privacy concerns are real and why privacy attacks are practical. Still, beyond the economic and legal incentives of protecting user privacy for companies, a compelling reason to study Privacy-Preserving Machine Learning (PPML) lies in it being an *interesting* problem. Citing [Raskhodnikova et al., 2008], the question essentially boils down to: "What can we learn privately"? We started with the idea of learning global distributions while rejecting individual information and found the ideal candidate framework in differential privacy. Remarkably, it offers a structural foundation to define privacy, and the necessary tools to quantify the privacy risk associated with statistical queries on a dataset. Additionally, it provides tools to sanitize non-private queries to the required privacy levels introducing randomness in the results. These strategies, as one should expect, prompt the data practitioner with an additional dilemma: what are we willing to give up, in terms of data utility, to protect data privacy? Although a general answer would not make sense and it ultimately depends on the user's perceived risk of privacy leakage, what we can do, as PPML researchers, is to optimize the trade-off to offer the best privacy guarantees at a fixed utility loss and the most utility for a prescribed privacy level. Within this mandate, this thesis focused specifically on the issue of sensitivity in differentially private learning. Since priva-

tizing the dataset queries used in gradient-based machine learning most often involves a degree of choice in selecting the sensitivity, we studied the implications of this choice, whether to even make it, and how to optimize it. The delimited nature of this research should not divert from its broader scope. Tackling the problem at its roots, that is optimizing privacy in the learning algorithm itself, inherits the applicability that is characteristic of gradient-based learning: throughout the chapters, we have applied these concepts to a myriad of settings (images, locations, prices, natural language), tasks (classification, regression, generation), models (convolutional neural networks, multilayer perceptrons, large language models, autoencoders), without making special assumptions. Yet, we can establish an order of what may be the expectation of practical viability of the research in this work. To the benefit of the reader, that is exactly the rationale behind the specific ordering of the chapters, which, to a certain extent, reflects the development of the research.

Results

The Introduction to the thesis in Chapter 1 discussed the context and background to the general topic of this thesis, i.e. how to privatize vector averages in machine learning, focusing on the impact of sensitivity and providing a practical example of where it plays a major role. In Chapter 2 we aimed to provide a solution to many coupled problems, optimizing over different metrics, i.e. privacy, fairness, and utility. The resulting contribution is thus with limited applicability to modern, large models, as already discussed, although it proved to be an effective defensive strategy against specific types of attacks. Additionally, we highlighted how the theoretical contributions introduced the use of metric-privacy for model sanitization in machine learning, extended the Laplace mechanism, and applied these novelties to personalized federated learning, as recognized by a number of subsequent works, e.g. [He et al., 2023; Zong et al., 2023], among others. Chapter 3 instead focused on the more established setting of approximate differential privacy, specifically targeting a common practice in current DPML research: not accounting for the privacy leakage incurred in optimizing for the sensitivity over a grid search. The current trend, for practical reasons, assumes that this selection does not consume privacy. When accounting for it, instead, we showed how optimizing the sensitivity while training the models is an effective strategy to make the best of the available privacy budget, improving utility at a fixed privacy level. We introduced a method to do so by drawing inspiration from a concurrent area of research, i.e. hyper-parameter optimization. Chapter 4 effectively

tackled a problem that companies are in fact already encountering to satisfy privacy regulations. When deploying large language models, the risk of privacy leakage can be minimized by regularly performing privacy audits. Our contribution then introduced a novel strategy to test these models under more stringent computational requirements, encouraging a larger adoption of these methods. With a minor departure from the theoretical-to-practical ordering, Chapter 5 challenged the current majority of DPML research, offering a different perspective on adaptive sensitivity for improved differentially private federated learning, and opening new avenues of research to future experimentation.

Future works

On top of experimenting with the theoretical speculations described in Chapter 5, we wish to highlight here additional paths this research could take. One of the possible areas to improve in metric privacy for distributed machine learning, as discussed in Chapter 2, lies in better results for computing the overall budget of a composition of queries. The best results for pure d-privacy (that is only involving a single privacy parameter ε) account for successive queries by summing the single privacy leakages. It is a limiting factor that in (ε, δ)-DP has been solved with numerous advanced compositionality theorems, and allowed for the current expansion of results in DPML. Conversely, although we experimented with extending our research to include (ε, δ) metric privacy guarantees, we struggled to derive meaningful theoretical results. With the results presented in Chapter 3, we presented a principled attempt at setting the clipping threshold, by addressing the sensitivity optimization along with parameter optimization. Admittedly, future work could focus on deriving the consequences of this coupled optimization in terms of the convergence rate of (DP)-SGD, which may present deviations from the already established theoretical analyses. Considering that this research focuses on the more standard definition of DP, it would be interesting to evaluate further experimental results to push the accuracy of large-scale DPML, following the considerations already discussed of being an area where there have not been major advancements, contrary to its non-private counterpart. In Chapter 4 we discussed how the use of noisy neighbors of the target model can be used as a surrogate for shadow models in privacy-auditing large language models. Naturally, one could extend this research to different architectures, different noise types, and most importantly, an automatic calibration of the standard deviations, to further support alignment with privacy regulatory frameworks.

Beyond iterative improvements on current research, one of the major

advancements in this field could be achieved by finding models which learn with fixed-sensitivity queries from the dataset. This could be achieved by architectures that, by construction, present a gradient that is bounded in norm, instead of requiring clipping. Naturally, this finding would also have a major impact on other areas that, for different reasons, require handling the gradient with respect to the parameters, such as explainable AI [Sundararajan et al., 2017]. Otherwise, finding learning paradigms that use fixed-sensitivity function queries from a dataset may require ditching gradient-based learning altogether, although the progress made thus far in standard neural networks trained with stochastic gradient descent and its significant advancements may have inadvertently led to a relative underestimation and diminished engagement in alternative machine learning paradigms. Ultimately, whether the advancement comes from one approach or the other, we underscore that unintended consequences on the fairness, privacy, and utility of records may still hinge on the sensitivity tradeoff discussed throughout this thesis, requiring careful consideration of all the variables involved.

Bibliography

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

Luís B. Almeida, Thibault Langlois, José D. Amaral, and Alexander Plakhov. *Parameter Adaptation in Stochastic Optimization*, pages 111–134. Cambridge University Press, USA, 1999. ISBN 0521652634.

Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security*, pages 901–914, 2013.

Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, pages 638–667. Springer, 2019.

Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 657–676, 2020.

Robert Bartlett, Adair Morse, Richard Stanton, and Nancy Wallace. Consumer-lending discrimination in the fintech era. *Journal of Financial Economics*, 143(1):30–56, 2022.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. Practical locally private heavy hitters. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3d779cae2d46cf6a8a99a35ba4167977-Paper.pdf.

Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *International Conference on Learning Representations*, 2018.

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 437–478. Springer, 2012.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459, 2017.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 21th ACM Conference on Computer and Communications Security (CCS 2014)*, 2014.

Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. URL http://arxiv.org/abs/2112.03570.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 82–102. Springer, 2013.

Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

Albert Cheu and Maxim Zhilyaev. Differentially private histograms in the shuffle model from fake users. *arXiv preprint arXiv:2104.02739*, 2021.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

Tommaso Cucinotta, Giacomo Lanciano, Antonio Ritacco, Fabio Brau, Filippo Galli, Vincenzo Iannino, Marco Vannucci, Antonino Artale, Joao Barata, and Enrica Sposato. Forecasting operation metrics for virtualized network functions. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 596–605. IEEE, 2021.

Tommaso Cucinotta, Luigi Pannocchi, Filippo Galli, Silvia Fichera, Sourav Lahiri, and Antonino Artale. Optimum VM placement for NFV infrastructures. In *2022 IEEE International Conference on Cloud Engineering (IC2E)*, pages 205–212. IEEE, 2022.

Rachel Cummings and Deven Desai. The role of differential privacy in gdpr compliance. In *FAT'18: Proceedings of the Conference on Fairness, Accountability, and Transparency*, page 20, 2018.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, 2013. doi: 10.1109/FOCS.2013.53.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

Cynthia Dwork and Guy Rothblum. Concentrated differential privacy. 2016.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014a.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014b.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv preprint arXiv:2001.03618*, 2020.

European Parliament, European Council. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. *arXiv preprint arXiv:2012.12803*, 2020.

Natasha Fernandes, Annabelle McIver, and Carroll Morgan. The laplace mechanism has optimal utility for differential privacy over continuous queries. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021.*, pages 1–12. IEEE, 2021. doi: 10.1109/LICS52264.2021.9470718. URL https://doi.org/10.1109/LICS52264.2021.9470718.

Filippo Galli, Sayan Biswas, Kangsoo Jung, Tommaso Cucinotta, and Catuscia Palamidessi. Group privacy for personalized federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

Filippo Galli, Saya Biswas, Kangsoo Jung, Catuscia Palamidessi, and Tommaso Cucinotta. On the adaptive sensitivity of differentially private machine learning. In *The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence (in Conjuction with AAAI 2023)*, 2023a.

Filippo Galli, Kangsoo Jung, Sayan Biswas, Catuscia Palamidessi, and Tommaso Cucinotta. Advancing personalized federated learning: Group privacy, fairness, and beyond. *SN Computer Science*, 4(6):831, 2023b.

Filippo Galli, Biswas Sayan, Jung Kangsoo, Tommaso Cucinotta, Palamidessi Catuscia, et al. Group privacy for personalized federated learning. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP)*, volume 1, pages 252–263. SciTePress, 2023c.

Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against LLMs. In *(Under Review) The Fifth Workshop on Privacy in Natural Language Processing (in Conjuction with ACL 2024)*, 2024a.

Filippo Galli, Catuscia Palamidessi, and Tommaso Cucinotta. Online sensitivity optimization in differentially private learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12109–12117, 2024b.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2021.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Antonio Gulli. The anatomy of a news search engine. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 880–881, 2005.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12, 2009.

Rema Hanna and Leigh Linden. Measuring discrimination in education. Technical report, National Bureau of Economic Research, 2009.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Zaobo He, Lintao Wang, and Zhipeng Cai. Clustered federated learning with adaptive local differential privacy on heterogeneous iot data. *IEEE Internet of Things Journal*, 2023.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7(10):9530–9539, 2020.

W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12080–12091. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8ce6fc704072e351679ac97d4a985574-Paper.pdf.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. URL http://arxiv.org/abs/1311.0776.

Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR, 2016.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jakub Konečny, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

Jakub Konečny, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

Antti Koskela, Mikko A Heikkilä, and Antti Honkela. Tight accounting in the shuffle model of differential privacy. *arXiv preprint arXiv:2106.00477*, 2021a.

Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 3358–3366. PMLR, 2021b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.

Giacomo Lanciano, Filippo Galli, Tommaso Cucinotta, Davide Bacciu, and Andrea Passarella. Predictive auto-scaling with openstack monasca. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*, pages 1–10, 2021.

Daniel Le Métayer and Sourya Joyee De. PRIAM: a Privacy Risk Analysis Methodology. In G. Livraga, V. Torra, A. Aldini, F. Martinelli, and N. Suri, editors, *Data Privacy Management and Security Assurance*, Heraklion, Greece, September 2016. Springer. URL https://hal.inria.fr/hal-01420983.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. Does bert pretrained on clinical notes reveal sensitive data? *arXiv preprint arXiv:2104.07762*, 2021.

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Seppo Linnainmaa. The representation of the cumulative roundoff error of an algorithm as a taylor expansion of the local error. Master's thesis, University of Helsinki, 1970.

Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

Xiaoyuan Liu, Hongwei Li, Guowen Xu, Sen Liu, Zhe Liu, and Rongxing Lu. Padl: Privacy-aware and asynchronous deep learning for iot applications. *IEEE Internet of Things Journal*, 7(8):6955–6969, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Agnes Lydia and Sagayaraj Francis. Adagrad—an optimizer for stochastic gradient descent. *Int. J. Inf. Comput. Sci*, 6(5):566–568, 2019.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):3–es, 2007.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

Saeed Mahloujifar, Esha Ghosh, and Melissa Chase. Property inference from poisoning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1120–1137. IEEE, 2022.

Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. On the applicability of machine learning fairness notions. *ACM SIGKDD Explorations Newsletter*, 23(1):14–23, 2021.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017a.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.

H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018a.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018b.

Casey Meehan, Amrita Roy Chowdhury, Kamalika Chaudhuri, and Somesh Jha. A shuffling framework for local differential privacy. *arXiv preprint arXiv:2106.06603*, 2021.

Sebastian Meiser. Approximate and probabilistic differential privacy definitions. *Cryptology ePrint Archive*, 2018.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

Shubhankar Mohapatra, Sajin Sasy, Xi He, Gautam Kamath, and Om Thakkar. The role of adaptive optimizers for honest private hyperparameter selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7806–7813, 2022.

Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Ting-Hao Huang, and Shomir Wilson. Nationality bias in text generation. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–122, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.9. URL https://aclanthology.org/2023.eacl-main.9.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

NIST. Nist privacy framework core. URL https://www.nist.gov/system/files/documents/2021/05/05/NIST-Privacy-Framework-V1.0-Core-PDF.pdf.

Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-70L8lpp9DF.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sofya Raskhodnikova, Adam Smith, Homin K Lee, Kobbi Nissim, and Shiva Prasad Kasiviswanathan. What can we learn privately. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, pages 531–540, 2008.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečny, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

David Martınez Rubio. Convergence analysis of an adaptive method of gradient descent. *University of Oxford, Oxford, M. Sc. thesis*, 2017.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015a.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015b.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. doi: 10.1109/SP.2017.41.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

David M Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *Proceedings on privacy enhancing technologies*, 2019(2):245–269, 2019.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013.

State of California. California Consumer Privacy Act (CCPA), 2018. URL https://oag.ca.gov/privacy/ccpa.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

Vinith M Suriyakumar, Nicolas Papernot, Anna Goldenberg, and Marzyeh Ghassemi. Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 723–734, 2021.

Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. Truth serum: Poisoning machine learning models to reveal their secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2779–2792, 2022.

Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

Alan Mathison Turing. Computing machinery and intelligence. *Mind*, 49: 433–460, 1950.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pages 1–7, 2018.

Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2021.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *arXiv preprint arXiv:2305.14710*, 2023.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Virtual prompt injection for instruction-tuned large language models. *arXiv preprint arXiv:2307.16888*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. Local differential privacy-based federated learning for internet of things. *IEEE Internet of Things Journal*, 8(11):8836–8853, 2020.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. BERT-based lexical substitution. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1328. URL https://aclanthology.org/P19-1328.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

Zixiao Zong, Mengwei Yang, Justin Ley, Carter T Butts, and Athina Markopoulou. Privacy by projection: Federated population density estimation by projecting on random features. In *Proceedings on Privacy Enhancing Technologies. Privacy Enhancing Technologies Symposium*, volume 2023, page 309. NIH Public Access, 2023.