EPFL

Thèse n° 7216

Data-Aware Privacy-Preserving Machine Learning

Présentée le 21 octobre 2020

à la Faculté informatique et communications
Laboratoire d'intelligence artificielle
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Aleksei TRIASTCYN

Acceptée sur proposition du jury

Prof. E. Telatar, président du jury
Prof. B. Faltings, directeur de thèse
Prof. H. Yu, rapporteur
Prof. C. Dimitrakakis, rapporteur
Prof. M. Jaggi, rapporteur

École polytechnique fédérale de Lausanne

2020

It can scarcely be denied that the supreme goal
of all theory is to make the irreducible basic
elements as simple and as few as possible
without having to surrender the adequate
representation of a single datum of experience.
— Albert Einstein

To my family and friends ...

Acknowledgements

First of all, I am very grateful to my thesis director Prof. Boi Faltings for providing valuable supervision, support, and scientific freedom to pursue my research interests. I would also like to thank Prof. Emre Telatar for insightful discussions on a lot of subjects and for agreeing to serve as the president of my thesis committee. I am thankful to Prof. Martin Jaggi, Prof. Christos Dimitrakakis, and Prof. Han Yu for taking their time to participate in my thesis defence and their knowledgeable comments and helpful suggestions. To current and former LIA and HCI lab members for helping with many aspects of my work, be it teaching or research, and just being great colleagues: Panayiotis, Fei, Naman, Adam, Ljubomir, Onur, Claudiu, Marina, Valentina, Stephane, Aris, Marija, and Diego, who also helped me with the French version of the abstract. A special thanks to Igor K., Goran, and Florent for effortless onboarding in the lab, our profound conversations and all the memorable moments in Switzerland and beyond. To Sylvie Thomet for her tireless help with all the administrative services, and to many students with whom I worked on exciting projects. I am also very appreciative of the feedback given by my friends, Sergey, Roman, and Igor M., which helped me improve my research and this thesis in particular.

My entire experience of the past few years would not have been as enjoyable without many friends who shared with me a great number of unforgettable and delightful moments. My warm gratitude goes to Artem, Arthur, Bill, Chryssa, Denys, Goran, Gorica, Igor K., Igor M., Matt, Miji, Oulfa, Pamela, Patti, Praneeth, Roman, Sergey, Stan, Vassilis, as well as the "mountain gang"—Camille, Davide, Sami, Sam, Yann, Louise—and many others. I am also extremely thankful to my friends from Russia who greatly encouraged and motivated me—Irina, Olga, Dima B., Dima K., Ilya, Anna, Kirill, Olya, Maria, Anastasiya, Marina, Sasha, and others.

Most importantly, I want to thank my family, and especially, my parents Natalia and Nikolai and my sister Inna. I am enormously grateful for their everlasting love, encouragement, and support in all my endeavours.

Lausanne, September 29, 2020 A. T.

Abstract

In this thesis, we focus on the problem of achieving practical privacy guarantees in machine learning (ML), where the classic differential privacy (DP) fails to maintain a good trade-off between user privacy and data utility. Differential privacy guarantee may be influenced by extreme outliers or samples outside of the data distribution to a large extent. For example, when trying to protect a classification model for magnetic resonance imaging (MRI), differentially private mechanisms would add the amount of noise sufficient to hide any image in the space of the same dimensionality. That includes images that do not belong to the intended data distribution (cars, houses, animals, and so on). Such generality inevitably yields poor privacy guarantees. Based on these observations and the ideas of DP, we propose a data-aware approach to privacy in machine learning. We design two novel privacy notions, *Average-Case Differential Privacy (ADP)* and *Bayesian Differential Privacy (BDP)*, which allow to take into account the data distribution information and significantly improve the privacy-utility balance.

First, we present average-case differential privacy, an empirical privacy notion designed for *ex post* privacy analysis of generative models and privacy-preserving data publishing. It relaxes the worst-case requirement of differential privacy to the average case and relies on empirical estimation to deal with undefined distributions. This notion can be regarded as a statistical sensitivity measure – it measures the expected change in the model outcomes given a change in the inputs generated by an observed distribution.

Second, we develop a more rigorous privacy notion, Bayesian differential privacy, based on the same high-level principle of probabilistic sensitivity measure. As the main theoretical contributions of this thesis, we formulate and prove basic properties of Bayesian DP, such as composition, group privacy, and resistance to post-processing, and we develop a novel privacy accounting method for iterative algorithms based on the advanced composition theorem. Furthermore, we show connections between our accountant and the well-known moments accountant, as well as between Bayesian DP and other privacy definitions.

Our practical contributions and evaluation branch into three main areas: (1) privacy-preserving data release using generative adversarial networks (GANs); (2) private classification using convolutional neural networks and other ML models; and (3) private federated learning (FL) for both discriminative and generative models. We demonstrate that both notions allow to achieve considerably higher utility than differential privacy, and that Bayesian DP provides a superior trade-off between privacy guarantees and the output model quality in all settings.

Keywords: privacy-preserving machine learning, privacy-preserving data release, differential privacy, deep learning, federated learning, generative adversarial networks

Résumé

Dans cette thèse, nous nous concentrons sur le problème d'obtention de garanties de confidentialité dans l'apprentissage automatique (ML), où la confidentialité différentielle classique (DP) ne parvient pas à maintenir un bon compromis entre la confidentialité des utilisateurs et l'utilité des données. La garantie différentielle de confidentialité peut être influencée dans une large mesure par des valeurs aberrantes ou des échantillons en dehors de la distribution des données. Par exemple, en essayant de protéger un modèle de classification pour l'imagerie par résonance magnétique (IRM), des mécanismes différentiels privés ajouteraient la quantité de bruit suffisante pour cacher toute image dans un même espace. Cela inclut les images qui n'appartiennent pas à la distribution de données initiales (voitures, maisons, animaux, etc.). Une telle généralité produit inévitablement de mauvaises garanties en termes de confidentialité. Sur la base de ces observations et des idées de DP, nous proposons une approche de la confidentialité basée sur les données d'apprentissage. Nous concevons deux nouvelles notions de confidentialité, Average-Case Differential Privacy (ADP) et Bayesian Differential Privacy (BDP), qui permettent de prendre en compte les informations de distribution des données et d'améliorer significativement l'équilibre confidentialité-utilité.

Premièrement, nous présentons Average-case Differential Privacy, une notion empirique de confidentialité conçue pour l'analyse de la confidentialité des modèles génératifs et la publication de données préservant la confidentialité. Il assouplit l'exigence du pire des cas de confidentialité différentielle au cas moyen et s'appuie sur une estimation empirique pour traiter les distributions non définies. Cette notion peut être considérée comme une mesure de sensibilité statistique – elle mesure le changement attendu des résultats du modèle en cas de changement d'un attribut d'entrée générés selon une distribution observée.

Deuxièmement, nous développons une notion de confidentialité plus rigoureuse, Bayesian differential privacy, basée sur le même principe de mesure de sensibilité probabiliste. En tant que principales contributions théoriques de cette thèse, nous formulons et prouvons les propriétés de base du Bayesian DP, telles que la composition, la confidentialité du groupe et la résistance au post-traitement, et nous développons une nouvelle méthode de mesure de la confidentialité, le comptable bayésien, pour les algorithmes itératifs basée sur le théorème de composition avancée. De plus, nous montrons les liens entre le comptable bayésien et le comptable des moments, ainsi qu'entre Bayesian DP et d'autres définitions de la confidentialité.

Résumé

Nos contributions pratiques et nos évaluations s'articulent autour de trois domaines principaux : (1) la divulgation de données préservant la confidentialité à l'aide de réseaux antagonistes génératifs (GAN) ; (2) classification privée utilisant des réseaux de neurones convolutifs et d'autres modèles ML ; et (3) l'apprentissage fédéré privé (FL) pour les modèles discriminants et génératifs. Nous démontrons que les deux notions permettent d'obtenir une utilité considérablement plus élevée que la confidentialité différentielle, et que Bayesian DP fournit un compromis supérieur entre les garanties de confidentialité et la qualité du modèle dans tous les contextes.

Mots-clés : apprentissage automatique préservant la confidentialité, divulgation de données préservant la confidentialité, confidentialité différentielle, apprentissage profond, apprentissage fédéré, réseaux adverses génératifs

Contents

List of Figures

List of Tables

1 Introduction

Machine learning (ML) and data analytics offer a great number of opportunities for companies, governments and individuals to use the accumulated data for their benefit. At the same time, however, the ability of these technologies to capture fine levels of detail can potentially compromise privacy of data owners. According to recent research (Fredrikson et al., 2015; Shokri et al., 2017; Hitaj et al., 2017), it is possible to infer information about individual records in the training set even in a black-box setting.

Numerous solutions have been proposed to tackle this problem. These solutions vary in the way privacy is achieved and the extent of data and user protection. Moreover, there is no unique way of defining privacy. The research community established a significant variety of formal privacy definitions, ranging from "lightweight" heuristics to rigorous theoretical notions. One of these definitions – *differential privacy (DP)* (Dwork, 2006; Dwork et al., 2006b,a) – stands apart as the gold standard widely accepted by the community.

Differential privacy, in its conventional forms, is independent of the data distribution. This property is being praised as one of the strongest arguments in favour of DP: all users, past and future, independent of their characteristics, are protected by the same guarantee. However, this is not well-matched with the modern machine learning context, where models are specialised and trained on particular kinds of data. As a result, achieving meaningful privacy guarantees in machine learning with DP is often extremely difficult and leads to pronounced reduction of accuracy. This is especially evident in private *data release*, where the task is to publish or provide "unrestricted" access to a dataset containing sensitive information.

On the other hand, privacy-preserving data release provides a large number of advantages over *model release* methods, where data remains a secret and only a model trained on it is released. Perhaps most importantly it offers flexibility. Once the data is sanitised and released, one could freely browse and explore it, perform any desired data analysis, or train any machine learning model on it.

In this thesis, we propose two alternative versions of DP with a bias towards ML: *Average-Case Differential Privacy (ADP)* (in Part I) and *Bayesian Differential Privacy (BDP)* (in Part II). Both take the data distribution into account to quantify privacy in a more meaningful way. Both enable more practical privacy-preserving data release. And crucially, Bayesian DP is a general-purpose definition applicable beyond machine learning.

Our first alternative notion is motivated directly by the challenges of private release for complex, high-dimensional data. One of the main obstacles in this setting is high worst-case sensitivity of the output (in this case, a "summary" dataset of some form, anonymised records, etc.) to changes in the input. Simultaneously, there is little research on how to determine sensitivity for a typical case: how much would a typical output change given the addition or removal of a typical data example at the input. In other words, how much privacy is preserved by the nature of the data itself. For example, a single change in a large set of very homogeneous data is unlikely to noticeably change the output, and thus, unlikely to lead to a privacy leak. These questions give rise to the concept of *average-case differential privacy*. The name is due to the fact that it is defined in a very similar manner to DP, but for typical (average-case) scenarios. However, in its essence, it is rather a statistical measure of sensitivity than a privacy notion, similarly to empirical DP (Abowd et al., 2013; Charest and Hou, 2017), which it is based on. With this concept we define an important abstract idea of factoring the natural data randomness in a privacy definition in the form of probabilistic sensitivity, as opposed to using a traditional deterministic worst-case bound.

In Part II, we take the high-level idea of probabilistic sensitivity and use it as a basis for a new privacy definition, *Bayesian differential privacy*. Unlike ADP, Bayesian DP is not a function of particular past outputs, but a guarantee on future outputs. Therefore, Bayesian DP is not just a measure of sensitivity like ADP, it is a proper privacy notion in the conventional DP sense. This is a very important difference to keep in mind when reading this thesis or deciding what to use in practical applications. In brief, if one needs a theoretical privacy guarantee, BDP is the right choice among the two; on the other hand, ADP can be used if one needs a heuristic statistical measure of how much information is leaked due to the nature of data, and does not necessarily want to change their algorithm and impose any additional privacy protection.

In the remainder of this chapter, we expand on our motivation for improving private data release methods and researching alternative privacy definitions, highlight our contributions, and outline the structure of this thesis.

1.1 Motivation

Our initial motivation for this work was to develop a practical privacy-preserving data release solution. In comparison to more wide-spread model release techniques, it offers a number of benefits. Probably the most valuable one is flexibility. When releasing private models, the trusted data curator has to construct and publish a separate model for each new application or analysis that needs to be performed on the sensitive data. On the other hand, having released the privatised data set, the trusted curator enables any downstream analysis or application without further effort. We outline other advantages of data release over model release in Section 2.2.2.

Previous work on private data release focused on numerical databases and discrete datasets, allowing to use simpler generative models (e.g. Bayesian networks (Bindschaedler et al., 2017)). However, even for these simplified conditions, solutions were scarce due to high sensitivity of the publisheable database to changes in the input (Zhu et al., 2017). The difficulty of generating complex data types and sensitivity of the data generation process lead us to explore the novel concept of *generative adversarial networks (GANs)* (Goodfellow et al., 2014). Similarly to other neural-network-based approaches, this method offers a scalable solution for large and complex datasets. Moreover, provided a correct training process, it can solve the sensitivity problem, because it is designed not to rely on any single input example in particular.

In the process of developing a solution for the initial problem, the scope of our work has expanded and motivation has evolved. More specifically, training GANs with differential privacy has proven to be unstable and not practical. Due to excessive amounts of noise, necessary to provide reasonable privacy guarantees, the two networks within a GAN (*generator* and *discriminator*) could not be trained in a balanced way, and thus, could not converge to the correct distribution. This practical obstacle motivated us to change our focus to a more general problem of defining and quantifying privacy in ML.

Maintaining a relative, probabilistic guarantee, like differential privacy, is highly desirable. But the conventional DP often requires adding a lot of randomness to achieve a meaningful guarantee. We believe this is not due to inherent difficulty of protecting privacy in machine learning, but rather due to the generality of the DP definition. Not only does DP consider a very broad class of adversaries, it also does not make any assumptions about the data it protects. Consequently, DP algorithms treat all data as equally likely. For example, for a DP mechanism, seeing a landscape photo in the dataset of MRI images is just as likely as seeing another MRI image. Moreover, any random noise image of the same dimensionality is also considered equally likely. This generality makes it difficult to "hide" all the data points. Yet, landscape photos, and especially random noise images, are not of interest to the attacker and do not need the same degree of protection. Hence, we are motivated to develop a "data-aware" privacy definition, which would take into account the fact that some data points are more likely to appear in the dataset than others.

The first privacy definition we propose, average-case differential privacy, is inspired specifically by the GAN-based data release application. While it employs a "data-aware" mindset, it is rather limited due to a number of factors, such as unreliable divergence estimations in high-dimensional spaces and heuristic approach to sampling from data distributions (more on this in Section 3.5). For this reason, we develop the second concept – Bayesian differential privacy. It is a more general-purpose, rigorous notion that can be applied in a wide variety of privacy protection scenarios.

There is a number of challenges that need to be tackled when developing a data-aware definition. How to incorporate the data distribution information in the privacy guarantee? How to deal with the finite data sample size? How to avoid underestimating potential privacy risks of unseen data? We concentrate on all these questions in Chapter 5.

Finally, we also consider the setting of federated learning. It allows to relax assumptions on a trusted central data curator, and thus, further enhance user privacy. More specifically, a model is trained in a decentralised manner, with user data always remaining on their devices. Nevertheless, such a privacy-oriented setup does not provide theoretical privacy guarantees, and it is sensible to augment it with a formal privacy notion. Prior work demonstrated that it is easier to achieve practical DP guarantees in federated scenarios with large numbers of users (McMahan et al., 2017). However, applying DP remains a challenge for smaller user bases and more complex models, such as GANs. Furthermore, prior research focused on *client-level* privacy protection, largely overlooking scenarios where *instance-level* protection might be more important. One example of these scenarios is multiple hospitals collaboratively training a model on patients' data: patient privacy is far more important than hospital privacy. To address these issues, we propose using average-case DP and Bayesian DP in the context of federated learning and show how it can improve the model quality, reduce the number of communication rounds, and impose strong instance-level privacy guarantees.

1.2 Contributions

In order to address the challenges outlined above, we propose a number of novel concepts and techniques. Our main contributions in this thesis are the following:

- We present a privacy-preserving *data release* method based on generative adversarial networks (GANs). This method can be used to create private synthetic datasets for training other machine learning models. Such a solution provides more flexibility for performing downstream tasks involving data compared to more popular *model release* approaches. Unlike similar approaches, developed simultaneously with ours, we forgo the traditional notion of DP in favour of our custom designed notion – Average-Case Differential Privacy, allowing for synthesising higher quality data and achieving better privacy-utility balance.

- We propose *Average-Case Differential Privacy (ADP)*, a novel empirical privacy notion, building upon previous research in empirical differential privacy and on-average KL privacy. This notion allows to analyse the degree of privacy preservation in datasets created by a variety of generative models. We also develop a heuristic that allows to compute privacy estimates without re-training the generative model multiple times, avoiding computationally prohibitive procedure.

- We propose *Bayesian Differential Privacy (BDP)*, a variant of differential privacy tailored specifically for machine learning applications. This novel approach to quantifying privacy enables significant improvements in model accuracy, while still providing strong theoretical privacy guarantees. Despite being developed for ML, this privacy notion is almost as generic as DP and is widely applicable in other areas. Arguably, this is the most important contribution of the thesis, while ADP should be regarded as an intermediate step towards it.

- Along with BDP, we design a novel privacy accounting method for iterative algorithms, such as stochastic gradient descent. This work generalises and encapsulates several previously known and widely used methods, such as the moments accountant, thereby providing a clean, unified framework for accounting differential privacy in iterative algorithms. Moreover, by using a Bayesian approach and the maximum entropy principle, we are able to solve a long-standing problem of quantifying data-distribution-specific privacy guarantees in the absence of distributional information.

- We further adapt the aforementioned techniques to federated learning settings, developing one of the first federated data generation frameworks with GANs and improving privacy analysis of existing algorithms with Bayesian DP.

- Finally, we combine our initial GAN-based data release technique with Bayesian differential privacy. We demonstrate the ability of this approach to generate high-fidelity synthetic data and solve important ML development tasks, such as model debugging and privacy-preserving data annotation and labelling.

1.3 Organisation

This thesis comprises two major parts. The first one is centred around average-case differential privacy and its application to generative models. At a high level, it primarily deals with the private data release setting and ADP is used for *ex post* privacy analysis. The second part focuses on Bayesian differential privacy, developing its theoretical foundation, privacy accounting, and evaluating it in ML and FL applications. Unlike the first part, everything presented in Part II can be applied to both model and data release settings, or even beyond machine learning, and unlike average-case DP, Bayesian DP represents an *ex ante* probabilistic guarantee on private outcomes.

More specifically, the content is organised in the following way:

- Chapter 2 provides some useful background on the topics of privacy and machine learning.
- In Chapter 3, we present and evaluate a GAN-based private data release framework, along with Average-Case Differential Privacy and a heuristic privacy accounting routine.
- In Chapter 4, we extend the previous techniques to federated learning scenarios and show that similar advantages in data quality carry on from the centralised setting.
- Chapter 5 introduces the concept of Bayesian differential privacy. We formulate and prove its main properties, design a generic privacy accounting method, and compare its performance to the state-of-the-art differential privacy techniques.
- In Chapter 6, we show that Bayesian DP seamlessly translates to federated settings and present a FL solution for discriminative models with BDP.
- In Chapter 7, we explore a possibility of training GANs with BDP to solve two problems: (i) low-quality samples, characteristic of differentially private GANs; and (ii) absence of theoretical privacy guarantees in our solution from Chapter 3.
- Chapter 8 concludes the thesis with a summary and future research directions.

2 Background

2.1 Database Privacy

To protect privacy while still benefiting from statistical analysis and machine learning, a number of techniques and privacy notions have been developed over the years. Unfortunately, we cannot realistically cover the entire range of privacy research in the scope of this thesis, and hence, we will focus on a small number of widely accepted privacy definitions.

Technically, the only family of privacy notions necessary for understanding this thesis is *differential privacy (DP)* (Dwork, 2006). However, we also include short overviews of such notions as k-anonymity (Samarati and Sweeney, 1998), l-diversity (Machanavajjhala et al., 2007), and t-closeness (Li et al., 2007). The primary reason for this inclusion is the research motivation. Since differential privacy is a complex concept and is difficult to achieve in practical applications, we believe it is important to understand the underlying rationale behind favouring it over simpler notions.

Privacy concepts in this chapter are presented in the context of relational databases, the initial area of interest for privacy research. Let D denote a database (in the basic case, a table), and let $\{A_1, \ldots, A_n\}$ be a set of attributes. In our examples, each tuple of the database refers to an individual, although it can also represent an organisation or any other entity. Assume all explicit identifiers, such as names, social security numbers (SSNs), phone numbers, and so on, are removed or encrypted. Define a *quasi-identifier* $\{A_i, \ldots, A_j\}$ – a set of attributes that can be linked with external data to uniquely identify at least one individual. The privacy concepts below, prior to DP, rely on the notion of quasi-identifiers and aim to control its release.

2.1.1 k-Anonymity

One of the most intuitive notions of privacy is k-anonymity (Samarati and Sweeney, 1998; Samarati, 2001; Sweeney, 2002). For a data release, it is defined as follows.

Definition 1 (k-Anonymity Requirement)**.** *A release of data is k-anonymous if every combination of values of quasi-identifiers can be indistinctly matched to at least k individuals.*

Similarly, k-anonymity can defined for a table or a database, requiring that every sequence of values of every quasi-identifier occurs at least k times.

In order to enforce k-anonymity, we can apply *generalisation* or *suppression* (Samarati and Sweeney, 1998). The first refers to grouping values into more general categories, implying also the existence of a generalisation hierarchy for each attribute domain. For instance, it can be achieved by combining ZIP codes by first digits, or ages within a range. The second technique refers to removing information from the database. In the case of k-anonymity, removal is applied at the tuple level, meaning that a tuple can only be removed in its entirety. Basically, suppression allows to remove outliers, that would otherwise force an excessive amount of generalisation to achieve k-anonymity. In the context of this thesis, the idea is comparable to our vision of discounting outliers when quantifying privacy guarantees, explained in Chapter 5. Moreover, our Bayesian accountant (Section 5.4.2) provides a mechanism for removing outliers in a more rigorous sense.

The concept of k-anonymity is useful in practical applications and is simple to understand. It is implemented, for example, in *Have I Been Pwned?* service[1] to anonymously check if a searched password was leaked without fully disclosing it[2]. A similar technique is used by Google Chrome's Password Checkup extension[3]. Nevertheless, k-anonymity is vulnerable to relatively simple attacks and does not offer a thorough privacy guarantee.

The first type of attack is a *homogeneity attack*. It exploits the lack of diversity in a sensitive attribute. Imagine that in a k-anonymous database, one group of k records has the same value of a certain sensitive attribute (e.g. a medical diagnosis). Therefore, it is sufficient for an adversary to narrow his search down to this group to recover the value of the sensitive attribute of an individual.

Background knowledge attacks is the second class of attacks that can be performed against k-anonymity. It relies on correlating information in the anonymised database with external data sources to infer sensitive attribute values with some degree of certainty. For example, knowing that some of the sensitive values are much less likely for the

[1] https://haveibeenpwned.com
[2] https://www.troyhunt.com/ive-just-launched-pwned-passwords-version-2/
[3] https://security.googleblog.com/2019/02/protect-your-accounts-from-data.html

individual, and if the k-anonymous group contains a lot of such values, the attacker may infer the real value with high certainty.

Machanavajjhala et al. (2007) further elaborate on these vulnerabilities and partially address them by introducing the notion of l-diversity.

2.1.2 l-Diversity

Consider an *equivalence class* – a set of tuples whose non-sensitive attributes generalise to some value, i.e. these tuples are indistinguishable w.r.t. these attributes. Such blocks of records can be obtained, for instance, via k-anonymity. Machanavajjhala et al. (2007) provide the following definition of l-diversity.

Definition 2 (l-Diversity Principle)**.** *A block of tuples is l-diverse if it contains at least l "well-represented" values for the sensitive attribute S. A table is l-diverse if all blocks are l-diverse.*

In its simplest form, "well-represent" might just mean that there are at least l different values of the sensitive attribute. But the authors define two more instantiations of the notion: *entropy l-diversity* and *recursive (c, l)-diversity*. We refer the reader to the original article to explore these definitions (Machanavajjhala et al., 2007).

Although l-diversity addresses the problem of homogeneity attacks on k-anonymity, it does not fully protects against background knowledge attacks. Furthermore, it has other limitations (Li et al., 2007). First, it may be difficult to achieve, especially due to higher dimensionality in cases where there is more than one sensitive attribute. Second, it is ill-protected against *skewness attacks* exploiting the skewness of the overall data distribution. Finally, *similarity attacks* can also pose a problem, because even if the values in an equivalence class are distinct, they may be semantically similar (e.g. an adversary might learn that the individual has low or high income without knowing the exact salary). These shortcomings led to the development of another concept – t-closeness.

2.1.3 t-Closeness

Let Q denote the distribution of a sensitive attribution in the whole database, and P – the distribution of this attribution within an equivalence class. According to Li et al. (2007), t closeness is defined as follows.

Definition 3 (t-Closeness Principle)**.** *An equivalence class satisfies t-closeness if the distance between P and Q is bounded by a threshold t. A table satisfies t-closeness if all equivalence classes satisfy t-closeness.*

Li et al. (2007) argue that requiring P and Q to be close would effectively reduce the usefulness of the released information due to limiting the correlation between quasi-identifiers and sensitive attributes, but at the same time, they point out that this is a necessary trade-off between utility and privacy to prevent privacy disclosures.

As a measure of closeness, the authors consider *total variation distance* and *Kullback-Leibler (KL) divergence*, but motivate employing *Earth Mover's distance (EMD)* instead, because the former do not reflect semantic distances between values.

While t-closeness improves upon l-diversity and k-anonymity, it also has serious drawbacks. It is challenging to achieve with multiple sensitive attributes, and EMD is insufficient as a similarity measure, as it does not capture information gain from the two equally distant distribution changes (e.g. a change from the distribution $(0.01, 0.99)$ to $(0.11, 0.89)$ might be more informative than from $(0.4, 0.6)$ to $(0.5, 0.5)$). But most importantly, t-closeness still provides only limited protection against an adversary with auxiliary background knowledge. This, and the fact that datasets in machine learning are generally considerably more complex and high-dimensional than the ones considered by the authors of the above privacy definitions, sets *differential privacy (DP)* as the primary privacy concept in machine learning.

2.1.4 Differential Privacy

Differential privacy (DP) (Dwork, 2006; Dwork et al., 2006b,a) relies on an important impossibility result – impossibility of absolute disclosure prevention. The authors prove that a conventional desideratum for statistical database privacy, stating that access to a database should not enable an adversary to learn more about an individual than what could be learned without such access, cannot be achieved due to *auxiliary information* available to the adversary aside from access to the database. This issue prompted the authors to change from considering absolute privacy guarantees to relative ones: the risk of privacy disclosure is present even if an individual does not participate in a database, and it should not substantially increase as a result of participation. In other words, DP captures the increased risk to an individual's privacy incurred by participating in a database.

In order to achieve DP, one needs a source of randomness. Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a random function, mapping sensitive inputs from domain $\mathcal{D}$ to range $\mathcal{R}$ of privatised (or sanitised) outputs. In this context, the input space is a space of possible databases or datasets. We say that two datasets $D, D' \in \mathcal{D}$ are *adjacent*, or *neighbouring*, if they differ in a single data point. The output space can be a space of database query results. In machine learning, it is often a space of learnable model parameters (e.g. neural network weights).

Definition 4 (ε-Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies ε-differential privacy if for any two adjacent datasets*

$D, D' \in \mathcal{D}$ and for any set of outcomes $\mathcal{S} \subset \mathcal{R}$ the following holds:

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right].$$

The above notion is also called *Pure Differential Privacy*, and it is generally difficult to achieve for real datasets. Therefore, a relaxation of differential privacy, called *Approximate Differential Privacy* or (ε, δ)-*Differential Privacy* (Dwork et al., 2014), is more often used in machine learning.

> Throughout the thesis, whenever we refer to DP, we mean approximate DP unless explicitly stated otherwise.

Definition 5 ((ε, δ)-Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$ and for any set of outcomes $\mathcal{S} \subset \mathcal{R}$ the following holds:*

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta.$$

Another relaxation that can be seen in the literature is (ε, δ)-*Probabilistic Differential Privacy (PDP)* by Machanavajjhala et al. (2008). It is conceptually similar to approximate DP, with only a subtle difference in the definition, and is often mistaken for an equivalent of (ε, δ)-DP. However, it is important to distinguish these two definitions, because probabilistic DP does not have the same properties as approximate DP.

Definition 6 ((ε, δ)-Probabilistic Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-probabilistic differential privacy if for any two dataset $D \in \mathcal{D}$ the following holds:*

$$\Pr\left[\mathcal{A}(D) \in \mathrm{Disc}(D, \varepsilon)\right] \leq \delta,$$

where $\mathrm{Disc}(D, \varepsilon) = \{S \in \mathcal{R} \mid \exists D' \in \mathcal{D}, \left|\log \frac{\Pr[\mathcal{A}(D)=S]}{\Pr[\mathcal{A}(D')=S]}\right| > \varepsilon\}$ *is a disclosure set.*

One can show that (ε, δ)-PDP implies (ε, δ)-DP, meaning that PDP is a stricter definition than DP. In other words, a set of PDP algorithms is a subset of (ε, δ)-DP algorithms, but unlike the (ε, δ)-DP set, it is not closed under all operations (in particular, post-processing). It is also worth mentioning that this definition is useful for privacy accounting, as seen in Section 2.2.3 and Chapter 5.

The expression used to defined the disclosure set above is essential for understanding our work. This entity is called *privacy loss*. Before formalising it, let us also note that the privacy mechanism $\mathcal{A}$ can additionally take auxiliary inputs ξ, denoted by $\mathcal{A}(D, \xi)$. In such a case, $\mathcal{A}(\cdot, \cdot)$ satisfies (ε, δ)-DP if $\mathcal{A}(\cdot, \xi)$ is (ε, δ)-DP for every ξ.

Definition 7 (Privacy Loss)**.** *Privacy loss $L_{\mathcal{A}}$ of a randomised algorithm $\mathcal{A} : \mathcal{D} \times \Xi \to \mathcal{R}$ for an outcome $s \in \mathcal{R}$, datasets $D, D' \in \mathcal{D}$, and auxiliary information $\xi \in \Xi$ is given by:*

$$L_{\mathcal{A}}(w; D, D', \xi) = \log \frac{\Pr\left[\mathcal{A}(D, \xi) = w\right]}{\Pr\left[\mathcal{A}(D', \xi) = w\right]}.$$

Since we often are concerned with continuous outcome distributions (i.e. $w \in \mathbb{R}^m$) in ML, this statement is a slight abuse notation. What we actually mean in this case is the ratio of probability density functions $p_{\mathcal{A}}(w|D)$ and $p_{\mathcal{A}}(w|D')$. We also sometimes omit auxiliary information ξ in our notion.

A common way to achieve approximate DP is using Gaussian noise mechanism:

Definition 8 (Gaussian Mechanism)**.** *The Gaussian noise mechanism achieving (ε, δ)-DP, for a function $f : \mathcal{D} \to \mathbb{R}^m$, is defined as*

$$\mathcal{A}(D) = f(D) + \mathcal{N}(0, \sigma^2 \mathbb{I}^m),$$

where $\sigma > C\sqrt{2 \log \frac{1.25}{\delta}}/\varepsilon$ and $C = \max_{D,D'} \|f(D) - f(D')\|$ is the L2-sensitivity of f.

Finally, we have to mention another variation of differential privacy – *local differential privacy (LDP)*, or simply *local privacy* (Dwork et al., 2014). Essentially, LDP is a generalisation of DP. Until now, we considered a *centralised model* (or a *global model*) of privacy, where some central authority holds the data and adds noise to hide sensitive information of individuals. In the *local model*, individuals do not trust the central curator and sanitise their data themselves. Intuitively, it means that one cannot hide in the crowd, because the bound on outcome probabilities should now apply to any pair of individual records.

Definition 9 (ε-Local Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{X} \to \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies ε-local privacy if for any two inputs $x, x' \in \mathcal{X}$ and for any set of outcomes $\mathcal{S} \subset \mathcal{R}$ the following holds:*

$$\Pr\left[\mathcal{A}(x) \in \mathcal{S}\right] \leq e^{\varepsilon} \Pr\left[\mathcal{A}(x') \in \mathcal{S}\right].$$

The local privacy model is especially useful in the context of federated learning (Section 2.3.2), and our Bayesian DP (Chapter 5) is readily convertible to it.

For a more comprehensive overview of differential privacy and DP mechanisms, we refer the reader to (Dwork et al., 2014).

2.2 Machine Learning Privacy

2.2.1 Attacks on Machine Learning Models

In recent years, machine learning applications became a commonplace in many fields. With that, a body of work on security and privacy of ML methods is growing at a rapid pace. Researches discovered a number of important vulnerabilities and related attacks on ML models, and raised the question about developing suitable defences.

Model inversion (Fredrikson et al., 2015) and membership inference (Shokri et al., 2017), in particular, received considerable attention among the attacks that compromise privacy of training data. Both attacks rely on a *passive adversary*. Unlike *active adversaries*, passive adversaries cannot corrupt information or parties participating in a security protocol, they can only eavesdrop.

Model inversion (Fredrikson et al., 2014, 2015) is based on observing the output probabilities of the target model for a given class and performing gradient descent on an input reconstruction. Fredrikson et al. (2015) consider white-box and black-box attacks. In a white-box setting, an adversarial client can download the model, gaining access to its parameters. In a black-box setting, attackers can make prediction queries to the model, but not actually download it. The authors applied both white-box and black-box attacks to decision trees, as well as white-box attacks to face recognition models (softmax regression, multilayer perceptron, and stacked denoising autoencoder), and demonstrated that their approach can infer sensitive responses of survey respondents with high accuracy and extract images from facial recognition models.

Membership inference (Shokri et al., 2017) aims to uncover a simpler fact of presence or absence of a data record in a dataset. It assumes that the attacker has access to the data similar to the ones that were used to train the target model. The attacker then uses this data to train a "shadow" model, which mimics the target, and an attack model, which is used to infer membership. That is, the attack model predicts if a certain example has already been seen by the target model during training based on the output probabilities it elicits. Shokri et al. (2017) performed the attack in a black-box setting against the models trained in the cloud using Google Prediction API and Amazon ML. They showed that on some datasets the attack can reach high accuracy, posing serious privacy risks for participants of sensitive datasets.

An example of an *active* attack is a GAN-based approach by (Hitaj et al., 2017). In a collaborative deep learning setting, where a number of agents train a joint model by submitting local gradient updates to a server, Hitaj et al. (2017) use generative adversarial networks (GANs) to fool other agents into releasing more information about their data. They also demonstrate that differential privacy is effective at thwarting the attack if the ε value is sufficiently small.

2.2.2 Model Release vs. Data Release

Researchers tackle privacy issues in machine learning in two major directions. One approach is to ensure privacy of the model parameters before releasing it. We refer to these techniques as privacy-preserving *model release*. Another way is to sanitise the data itself, somehow removing all sensitive information, such that one could publish a dataset and allow to freely train models without being concerned about privacy. We name this family of algorithms privacy-preserving *data release* methods.

The purpose of this section is to contrast these two directions, outlining their strong and weak sides, and give some examples of techniques within each category. We defer a more detailed discussion of prior research in both of these areas to corresponding chapters.

An example of *model release* is a class of methods that enforce privacy during training. This includes, for example, DP-SGD (Abadi et al., 2016), PATE (Papernot et al., 2016, 2018), and DP-FedAvg (McMahan et al., 2017). These approaches perform well in ML tasks and provide strong privacy protection. Furthermore, they are often significantly easier to implement and tune in practice. However, these methods are often restrictive. First, releasing a specific trained model instead of data provides limited flexibility for future tasks. For instance, it reduces possibilities for integrating models trained on different sources of data. Hyper-parameter tuning and model evaluation is complicated by the additional need to adjust private training parameters. Finally, many of the proposed methods implicitly or explicitly assume access to public data of nature similar to private data, which may not be possible in such areas as medicine.

Data release techniques can range from simple anonymisation, which generally does not guarantee privacy, to seed-based approaches that transform and sanitise the original data points (Bindschaedler et al., 2017; Huang et al., 2017), to "seedless" approaches that generate synthetic data (Beaulieu-Jones et al., 2017; Triastcyn and Faltings, 2019c), to hybrid solutions (Fioretto and Van Hentenryck, 2019). In contrast to *model release*, privacy-preserving *data release* is more difficult to implement on real-world datasets, especially those involving complex data types, and it frequently results in lax privacy guarantees and lower data utility (meaning lower predictive performance of models trained on it). On the other hand, it has many immediate advantages. First of all, any machine learning model could be trained on the released data without additional restrictions. Second, one could pool data from different sources and use it to build stronger models. Third, releasing private data could help solve one of the most prominent obstacles to trading on data markets[4], anonymisation and protection of sensitive information. Moreover, private data publishing could facilitate reproducibility and transparency of research and scientific studies.

[4] https://www.datamakespossible.com/value-of-data-2018/dawn-of-data-marketplace

2.2.3 Differentially Private Machine Learning

Most of the literature on machine learning privacy is focused on privacy-preserving model release. Some researchers tackle this problem by using disjoint datasets and distributed training. Shokri and Shmatikov (2015) suggest such a manner of training, i.e. participants would keep the data locally and communicate sanitised updates to a central authority. This method, however, leads to high privacy losses (Abadi et al., 2016; Papernot et al., 2016). An alternative technique is suggested by Papernot et al. (2016). They also use disjoint training sets, but additionally, they build an ensemble of independently trained "teacher" models to transfer knowledge to "student" models. The knowledge transfer is organised by student models training on some public data labeled by the teachers. The authors expanded their result in (Papernot et al., 2018) and achieved state-of-the-art image classification results for private models with single-digit DP bounds ($\varepsilon < 10$). The disadvantage of these techniques is that they are complicated, and thus, more prone to errors. For instance, if one modifies PATE to propagate errors back through the aggregate teacher, DP guarantee could be violated. Jordon et al. (2018) add such a modification in their PATE-GAN framework, but it is not clear whether they account for the privacy leak resulting from the forward pass caching in backpropagation.

A different approach is taken by Abadi et al. (2016). They propose *differentially private stochastic gradient descent (DP-SGD)* to train deep learning models in a private manner. Their approach reaches high accuracy while maintaining relatively low DP bounds. Importantly, it is relatively simple to implement and understand, and therefore, is less prone to errors. However, all these methods may require access to some public data, which are similar to the sensitive data, in order to achieve acceptable privacy-utility trade-off. These data can be used for training the student model, like in PATE (Papernot et al., 2016), or for pre-training, like in DP-SGD (Abadi et al., 2016).

Abadi et al. (2016) introduced two techniques that are widely employed in modern privacy-preserving machine learning: *DP-SGD* and the *moments accountant (MA)*. They are also used in this thesis, and are important for understanding our contributions. Hence, we briefly describe these techniques below.

Differentially Private SGD

To achieve differential privacy in models trained with gradient descent, Abadi et al. (2016) consider every gradient update as a sensitive output. They apply Gaussian mechanism to bound ε and δ of every SGD iteration, and then use composition properties of DP to compute the overall privacy bound. In order to bound sensitivity (influence) of the gradient update at each iteration, the authors suggest clipping the gradient L2-norm. Algorithm 1 provides a pseudo-code of the algorithm for better clarity.

In order to ensure that each step of the algorithm is (ε, δ)-differentially private, σ is

Algorithm 1 Differentially Private SGD (adapted from Abadi et al. (2016))

Input:
Dataset $D = \{x_1, \cdots, x_N\}$, loss function $\mathcal{L}(w) = \frac{1}{N}\sum_i \mathcal{L}(w, x_i)$.
Parameters: learning rate η_t, noise scale σ, batch size B, gradient norm bound C.
Initialise w_0 randomly
for $t \in [1..T]$ **do**
Sample a random batch of examples $\mathcal{B}_t$ with sampling probability q
Compute gradient
For each $i \in \mathcal{B}_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(w_t, x_i)$
Clip gradient
$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
Add noise
$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
Descent
$w_{t+1} \leftarrow w_t - \eta_t \tilde{\mathbf{g}}_t$
Accumulate privacy loss for (ε, δ) **computation**
end for
Output: w_T, (ε, δ).

chosen to be $\sqrt{2\log\frac{1.25}{\delta}}/\varepsilon$. Using the moments accountant, described below, Abadi et al. (2016) were able to prove that Algorithm 1 is $(O(q\varepsilon\sqrt{T}), \delta)$-differentially private, where q is the probability of each individual example being in a batch, and T is the total number of steps taken by SGD during training.

Moments Accountant

Computing privacy guarantees for DP-SGD with basic DP composition theorems would result in extremely loose bounds. And given the large number of SGD steps, even the advanced composition bounds (Dwork et al., 2014) will not be sufficiently tight. Abadi et al. (2016) solve this problem by designing a new privacy accounting technique, named the *moments accountant (MA)*. The key idea of MA is to consider the privacy loss random variable (see Definition 7), calculate the tail bound on its distribution, and convert this tail bound to DP guarantees. Essentially, the (ε, δ) values obtained in this way correspond to probabilistic DP, but as we discussed earlier, it implies (ε, δ)-approximate DP.

2.3 New Directions in Machine Learning

Finally, we take a brief look at two novel research directions in machine learning that are promising for increasing privacy protection. First, generative adversarial networks (GANs) by Goodfellow et al. (2014) offer extended capabilities for privacy-preserving data synthesis, especially for complex, high-dimensional datasets. Second, *federated*

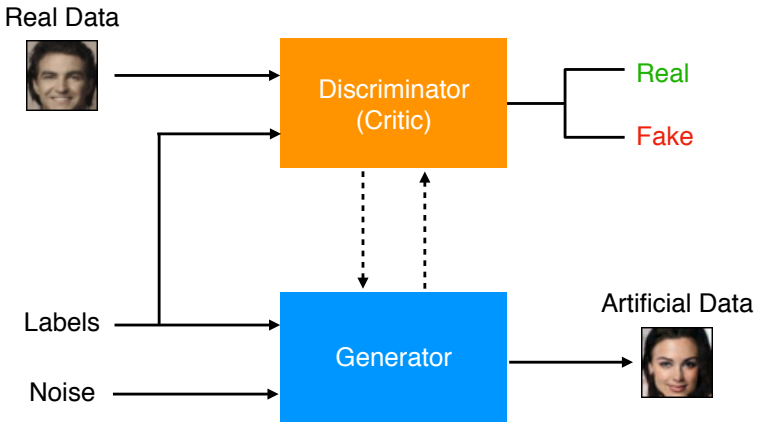


Figure 2.1 – A high level representation of a GAN.

learning (FL) (McMahan et al., 2016) provides an efficient framework for eliminating a trusted centralised data curator in collaborative machine learning, allowing users to keep sensitive data on-device.

2.3.1 Generative Adversarial Networks

In recent years, generative adversarial networks (GANs) by Goodfellow et al. (2014) have received a great deal of attention and pushed the boundaries for deep generative models along with variational autoencoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014; Gregor et al., 2015) and recursive neural networks (e.g. PixelRNN (Oord et al., 2016)). The original work on GANs has been followed by numerous extentions and variations of the concept (Salimans et al., 2016; Radford et al., 2015; Zhao et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Xu et al., 2019) The most successful application for such generative models so far has been realistic image generation, perhaps due to abundance of training data and inherent geometric structure.

In our work, we decided to focus on one type of deep generative models – GANs. There are several reasons for this choice. Firstly, GANs have shown very good results in practice, for example, generating significantly sharper images compared to other generative models. Secondly, the forward pass for generating data is much faster than for some other models, such as RNNs. Thirdly, the *generator* part of the model, the one we eventually interested in, does not interact with the real training data at any point in the learning process, only observing the gradients from the *discriminator*.

On the high level, GANs can be described as follows. The model consists of two separate components: the *generator* $G(z)$ and the *discriminator* $D(x)$. The latter is also called *critic* in the literature, and we use the two names interchangeably. They are independent and can be implemented as different machine learning models. The generator's goal

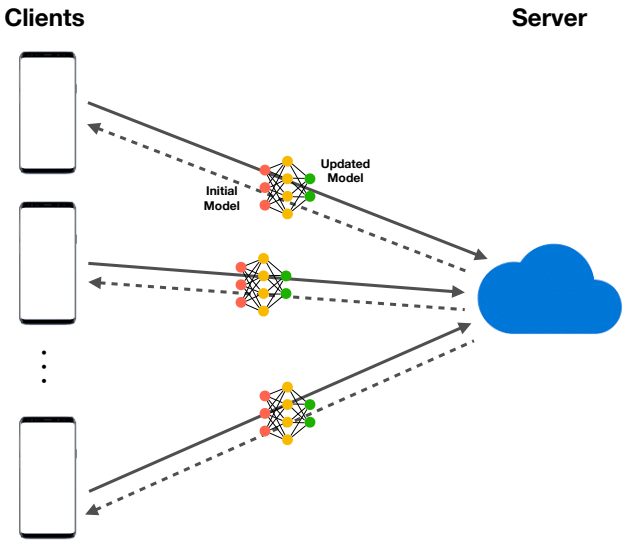


Figure 2.2 – A high level representation of federated learning.

is to produce realistic samples of data based on a random variable $z \sim p_z(z)$, while the discriminator is tasked with distinguishing real data samples $x \sim p_{\text{data}}(x)$ from generated samples $\hat{x} \sim p_g(x)$. These two models are trained in an adversarial fashion, essentially playing a two-player game, with the goal to converge to the Nash equilibrium. The parameters are optimised using simultaneous gradient ascent steps on both the discriminator and the generator to maximise the following functions correspondingly:

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))], \tag{2.1}$$

$$\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))]. \tag{2.2}$$

Some of the proposed variations of GANs modify this objective function in order to improve convergence and training stability, for example, substituting Jensen–Shannon divergence with Wasserstein distance (Arjovsky et al., 2017; Gulrajani et al., 2017)).

Training GANs can be challenging. The common practice is to use Adam optimisation method (Kingma and Ba, 2015) coupled with mini-batch, and utilise such techniques as feature matching, batch normalisation, and one-sided label smoothing to improve the convergence (Salimans et al., 2016). Another strategy is to apply semi-supervised learning that allows to use data labels as inputs in the generator and outputs in the discriminator.

2.3.2 Federated Learning

Federated learning (FL) (McMahan et al., 2016) is a novel machine learning technique for collaboratively training models by multiple parties without exchanging or centrally storing data. The approach is gaining a lot of popularity in recent years and is being actively developed (Konečnỳ et al., 2016; Bonawitz et al., 2017, 2019).

The key idea is that the parties (*clients*) can exchange model updates instead of data. In the context of gradient-based learning, *clients* can receive the initial model from the *server*, locally run the gradient descent on this model using their data, and then send the model update to the server. The server aggregates client updates and applies it to the model, each weighted by a share of client's data, and then sends the updated model to the clients. This process is repeated over a number of *communication rounds*. If the local model updates in each communication round are done once on the full local dataset, the approach is termed *Federated SGD*, or `FedSGD`. A generalisation of this algorithm with multiple local epochs is named *Federated Averaging*, or `FedAvg`.

There are two major advantages of federated learning: enhanced privacy and communication efficiency. The former stems from the fact that the data is kept local on user devices. The latter – from the lower number of communication rounds due to extended local updates. However, it is worth noting that federated learning as such does not provide any theoretical privacy guarantees. Rather, it portects the training process but not its outcome. The privacy attacks discussed earlier (Fredrikson et al., 2015; Shokri et al., 2017) are still applicable to the final model regardless of the training process.

Average-Case Differential Privacy Part I

3 Generating Data with Average-Case Differential Privacy

3.1 Introduction

We start our investigation with a specific instance of privacy-preserving data release problem. Our interest in this particular setting is explained in Section 2.2.2 where we highlight its advantages compared to the model release setting.

In particular, we are interested in solving two problems. First, how to preserve high utility of the released data for machine learning and data analysis algorithms while protecting the sensitive information. Especially in the case where the data is given by a complex continuous process (e.g. images, audio, video, and so on). Second, how to quantify privacy, i.e. the risk of recovering private information from the published dataset, and thus, the trained model.

The main idea of our approach is to use generative adversarial networks (GANs) (Goodfellow et al., 2014) to create artificial datasets to be used in place of real data for training. This method has a number of advantages over the earlier work (Abadi et al., 2016; Papernot et al., 2016, 2018; Bindschaedler et al., 2017). First of all, our solution allows releasing entire datasets, thereby possessing all the benefits of private *data release* as opposed to *model release*. Second, it can achieve high accuracy without pre-training on similar public data. Although it is also possible to pre-train the model, if such data is available, to learn generic low-level features. Third, it is more intuitive and flexible than some other methods, e.g. (Papernot et al., 2016), which have complex architectures and are more susceptible to implementation mistakes and associated privacy leaks, as discussed in Section 2.2.3.

An important observation about GANs is that, unlike many previous approaches, they do not use real data points as seeds when generating the new artificial examples. This

This chapter is based on the paper published in the Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies, AAAI Spring Symposium Series (Triastcyn and Faltings, 2019c).

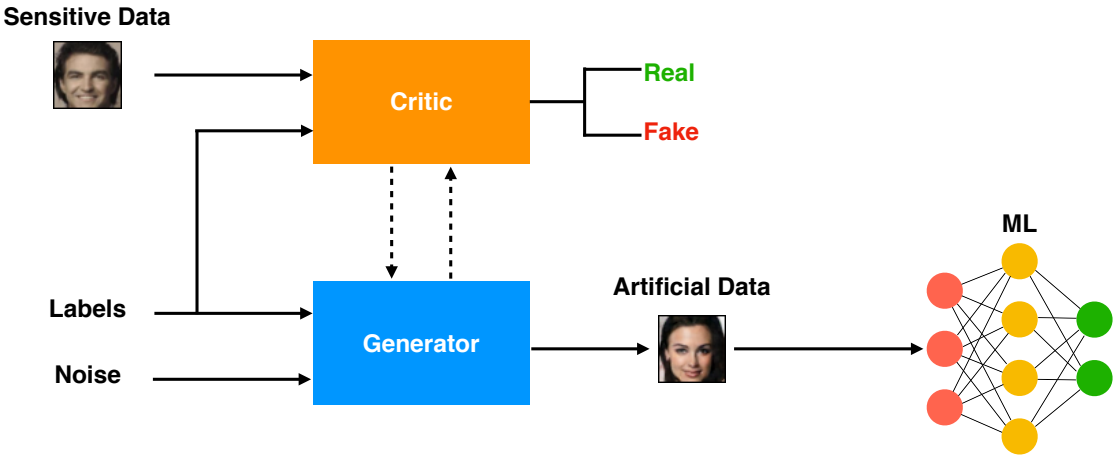


Figure 3.1 – Architecture of our solution. Sensitive data is used to train a GAN to produce a private artificial dataset, which then can be used by any ML model.

observation leads us to believe that there is some degree of privacy inherently present in synthetic datasets created by GANs. However, quantifying the degree of this privacy protection with existing notions, such as DP, is difficult. Therefore, we develop a novel privacy notion *Average-Case Differential Privacy (ADP)*, relaxing the original definition, and design an *ex post* analysis framework for generated data. We use Kullback–Leibler (KL) divergence estimation and Chebyshev's inequality to find statistical bounds on expected privacy loss for a dataset in question.

Our main contributions in this chapter are the following:

- we propose a novel, yet simple, approach for private data release, and to the best of our knowledge, this is the first practical solution for continuous, high-dimensional data, such as images;
- we introduce a new framework for statistical estimation of the expected privacy loss of the released data;
- we show that our method achieves learning performance of model release methods and is resilient to model inversion attacks.

The rest of this chapter is structured as follows. In Section 3.2, we give an overview of related work. Section 3.3 contains some preliminary information, such as a reminder on KL divergence and Chebyshev's inequality. In Section 3.4, we describe our notion of *Average-Case Differential Privacy* and the privacy estimation routine, as well as discuss its limitations. Experimental results and implementation details are presented in Section 3.6; and Section 3.7 concludes the chapter.

3.2 Related Work

In this section, we focus on the research related specifically to creating synthetic datasets, especially using generative adversarial networks, as well as relaxations of differential privacy relevant for our work. For the overview of related research on privacy attacks and protections in machine learning in general, we refer the reader to Chapter 2.

Despite most of the research in the area of privacy-preserving ML being concentrated on private model release, some researchers start to direct more attention to private data release, including privacy protection via generating synthetic data. Some recent examples are by Bindschaedler et al. (2017); Zhang et al. (2017); Huang et al. (2017); Beaulieu-Jones et al. (2017) and Fioretto and Van Hentenryck (2019). These approaches fall into a wider category of *non-interactive privacy mechanisms* for data publishing, but in this thesis, we are only going to focus on one instance of such mechanisms – synthetic data release (although the methods designed in Chapters 5 and 6 apply broadly, to both interactive and non-interactive settings). We also consider a narrower definition of *data publishing*, only referring to the methods that release an entire dataset, and not including, for instance, batch query publishing. For a more extensive overview of non-interactive methods, and in a wider sense of the term, we refer the reader to Zhu et al. (2017).

In non-interactive, data release scenarios, differential privacy is hard to guarantee, and thus, the proposed techniques tend to either relax the DP requirements or remain limited to simpler data (typically discrete and low-dimensional). Let us consider some of the more recent examples in more detail.

First, Bindschaedler et al. (2017) develop an alternative, formal notion of privacy, called *plausible deniability*, specifically designed for releasing sensitive datasets. The main idea of this notion is the following. Given some sanitisation mechanism, an output point of this mechanism can be released only if a pre-defined number of input points are indistinguishable, up to a privacy parameter. Bindschaedler et al. (2017) integrate the ideas of k-anonymity and differential privacy and prove that under certain conditions, plausible deniability yields DP. The authors then use a graphical probabilistic model to learn an underlying data distribution and transform real data points (*seeds*) into synthetic data points, which are then filtered by a privacy test based on a plausible deniability criterion. Unfortunately, this procedure would be rather expensive for complex, high-dimensional data, such as images, audio and video recordings, etc. Nevertheless, the authors provide an interesting and useful real-world application with location traces (Bindschaedler and Shokri, 2016).

Another method that works well for discrete data is a hybrid model/data release solution by Fioretto and Van Hentenryck (2019). It employs decision trees to simultaneously perform classification/regression and generate a synthetic dataset that can be published. Moreover, it guarantees a stronger ε-differential privacy, which is rare in realistic applica-

tions. However, like in the previous case, this approach is less suitable for high-dimensional and continuous data.

Alternatively, Huang et al. (2017) introduce the notion of *generative adversarial privacy* and use GANs to obfuscate real data points with respect to pre-defined sensitive attributes, enabling privacy for more complex, continuous data types. The downside of this approach is that it only hides a respective attribute and provides privacy against a specific adversary.

Finally, borrowing from the model release literature in machine learning, a natural approach to try is training GANs using DP-SGD or some other DP algorithm. This direction has gained a lot of traction in the last years, starting with the work by Beaulieu-Jones et al. (2017), performed in parallel with ours, and followed later by a number of papers based on this idea, extending it, and applying in different contexts (Xie et al., 2018; Zhang et al., 2018; Jordon et al., 2018; Long et al., 2019; Augenstein et al., 2019). However, it proved extremely difficult to stabilise training with the necessary amount of noise, which scales as $\sqrt{m}$ w.r.t. the number of model parameters m. It makes these methods inapplicable to more complex datasets without resorting to unrealistic (at least for some areas) assumptions, like access to public data from the same distribution.

Similarly, our approach uses GANs, but unlike the former approaches we do not restrict ourselves to the differential privacy guarantee, and unlike (Huang et al., 2017), the data is generated without real seeds and with the goal to hide all attributes. We verify empirically that out-of-the-box GAN samples can be sufficiently different from real data, and average-case privacy loss can be approximately bounded by single-digit numbers. To achieve this, we build upon the notions of *empirical differential privacy (EDP)* (Abowd et al., 2013) and *On-Average KL-Privacy* (Wang et al., 2016b).

Empirical DP was introduced in Abowd et al. (2013) for Bayesian linear mixed models. The main idea is to substitute the data-independent notion of DP with a data-dependent analogous notion. In other words, instead of bounding the maximum probability ratio for any two adjacent datasets D and D', EDP bounds the maximum probability ratio for the original dataset D and any D' obtained by removing a single example from D. The method has been later applied to Bayesian generalised linear mixed models, as well as zero-inflated Poisson models (Schneider and Abowd, 2015). The advantage of such approach is that it is more forgiving and contextual, because the guarantee takes into account the actual data. Besides, it can be easily computed for a wide range of Bayesian models. However, there are important limitations and conceptual differences from DP, investigated by Charest and Hou (2017). In Section 3.5.2 of this chapter, we elaborate some more on the limitations of this privacy definition in the context of (non-Bayesian) machine learning, and in particular, why it cannot be readily applied to the considered problem.

Another relaxation of DP that inspires our solution is On-Average KL-Privacy (Wang

et al., 2016b). In Section 2.1.4, we pointed out that the expectation of the privacy loss over the outcomes distribution is nothing but a KL divergence. While the traditional ε-DP bounds the worst-case privacy loss, i.e. taking a maximum over both the outcomes and the adjacent dataset pairs, ε-on-average KL-privacy is defined by the equivalent bound on the expectation (over the datasets distribution) of the KL divergence between two adjacent datasets. Therefore, it relaxes the worst-case bound to an expected-case bound. In the remainder of this chapter, we develop the notion of *average-case differential privacy*, borrowing ideas from both EDP and On-Average KL-Privacy to construct an ML-friendly relaxation of DP, primarily with data synthesis applications and post hoc privacy analysis in mind.

3.3 Preliminaries

This section contains a refresher on some mathematical notions used in this chapter. For more details on differential privacy, see (Dwork et al., 2014) and Chapter 2 of this thesis.

In our privacy estimation routine, we use the notion of Kullback-Leibler divergence:

Definition 10. *The Kullback–Leibler (KL) divergence between two continuous probability distributions P and Q with corresponding densities p, q is given by:*

$$D_{KL}(P\|Q) = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} dx. \tag{3.1}$$

Note that KL divergence between the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$ is nothing but the expectation of the privacy loss random variable $\mathbb{E}[L_{\mathcal{A}}(w, D, D')]$.

Additionally, we will use Chebyshev's inequality to obtain tail bounds:

$$\Pr(|x - \mathbb{E}[x]| \geq k\sigma) \leq \frac{1}{k^2}. \tag{3.2}$$

In particular, as we expect the distribution to be asymmetric, we use the version with semi-variances (Berck and Hihn, 1982) to get a sharper bound:

$$\Pr(x \geq \mathbb{E}[x] + k\sigma) \leq \frac{1}{k^2} \frac{\sigma_+^2}{\sigma^2}, \tag{3.3}$$

where $\sigma_+^2 = \int_{\mathbb{E}[x]}^{+\infty} p(x)(x - \mathbb{E}[x])^2 dx$ is the upper semi-variance.

3.4 Our Approach to Generating Private Data

In this section, we describe the key idea of our approach to generating private data – using generative adversarial networks (GANs). We also discuss one further improvement that can boost not only privacy, but also the quality of the generated data. Finally, we outline the limitations of the method.

The main idea of our approach is to use artificial data for learning and publishing instead of real (see Figure 3.1 for a general workflow). The intuition behind it is the following. Since it is possible to recover training examples from ML models (Fredrikson et al., 2015), we need to limit the exposure of real data during training. While this can be achieved by DP training (e.g. with DP-SGD), it would have the limitations mentioned earlier. Besides, certain attacks can still be successful if DP bounds are loose (Hitaj et al., 2017). Removing real data from the training process altogether would add another layer of protection and limit the information leakage to artificial samples. What remains to show is that the artificial data is sufficiently different from the real one.

3.4.1 Differentially Private Critic

Despite the fact that the generator does not have access to real data in the training process, one cannot guarantee that generated samples will not repeat the input. To alleviate this problem, we propose to enforce differential privacy on the output of the discriminator (*critic*). This is done by employing the Gaussian noise mechanism (Dwork et al., 2014) at the second-to-last layer: clipping the $L2$ norm of the input and adding Gaussian noise. To be more specific, activations $a(x)$ of the second-to-last layer become $\tilde{a}(x) = a(x)/\max(\|a(x)\|_2, 1) + \mathcal{N}(0; \sigma^2)$. We refer to this version of the critic as *DP critic*. It is important to keep in mind that only the critic outputs are differentially private, not the critic parameters.

If the chosen GAN loss function was directly differentiable w.r.t. generator output, i.e. if critic could be treated as a black box, this modification would enforce the same DP guarantees on generator parameters, and consequently, all generated samples. Unfortunately, probably the only way to achieve it in practice is using finite differences instead of backpropagation, which is not feasible.

As our evaluation shows, this modification has a number of advantages. First, it improves diversity of samples and decreases similarity with real data. Second, it allows to prolong stable training, and hence, obtain higher quality samples. Finally, in our experiments, it significantly improves the ability of GANs to generate samples conditionally.

3.4.2 Limitations

The major drawback of this solution for privacy-preserving data release is that all the existing limitations of GANs (or generative models in general), such as training instability or mode collapse, will apply to this method. Hence, at the current state of the field, our approach may be difficult to adapt to inputs other than image data. Yet, there is still a number of privacy-sensitive applications, e.g. medical imaging or facial analysis, that could benefit from our technique. And as generative methods progress, new uses will be possible.

3.5 Average-Case Differential Privacy

Because of the restrictions that the conventional differential privacy poses in ML context, especially for generative models, we propose a novel privacy definition – *Average-Case Differential Privacy (ADP)* – a relaxed version of differential privacy that aims to provide empirical expected guarantees rather than worst-case guarantees. We build upon the ideas of *empirical DP* (EDP) (Abowd et al., 2013) and *on-average KL privacy* (Charest and Hou, 2017). The first can be viewed as a measure of sensitivity of the outcomes to (in our case, generated data distributions) to changes in the inputs. And as we explain below, so is our definition. The second relaxes DP to the average-case notion.

It is worth mentioning that, in the context of this thesis, ADP should be viewed as the first attempt of incorporating the data distribution information in a privacy definition, in a way that is tuned for generative models. In Part II, we build upon the same high-level idea and move towards a more universal, practical, and rigorous notion of Bayesian differential privacy.

3.5.1 Definiton

Let us formally define average-case differential privacy.

Definition 11 (Average-Case Differential Privacy)**.** *A randomised mechanism $\mathcal{A}$ is said to be (μ, γ)-average-case differentially private if for two neighbouring datasets D, D', where data points are identically distributed, and a set of outputs S, s.t. $|S| \approx |D|$, it holds that*

$$\Pr\left(\bar{L}(S, D) > \mu\right) \leq \gamma, \tag{3.4}$$

where $\bar{L}(S, D)$ is an estimator of the expected privacy loss $\mathbb{E}_{s\sim\mathcal{A}(\mathcal{D})}\left[|L_{\mathcal{A}}(s, D, D')|\right]$ (defined in Section 3.5.2).

One may notice that this definition is more akin to (ε, δ)-probabilistic DP (see Definition 6), which in short can be written as

$$\Pr(L > \varepsilon) \leq \delta,$$

rather than (ε, δ)-DP (Definition 5). Moreover, similarly to EDP (Charest and Hou, 2017), it is more appropriate to regard the ADP bound as a measure of sensitivity rather than a privacy definition in the traditional DP sense, because it is a function of outputs S. However, while EDP concerns a specific dataset D, with ADP we attempt to generalise the bound to the data distribution, and thus, call it a statistical sensitivity measure.

For the sake of example, let each data point in D, D' represent a single user. Then, $(0.01, 0.001)$-ADP could be interpreted as follows: with probability 0.999, a user from the same distribution submitting their data will change outcome probabilities of the private algorithm on average by 1% (because $e^{0.01} \approx 1.01$.).

3.5.2 Privacy Estimation

In the case of many generative models, and in particular GANs, we don't have access to exact posterior distributions which are used to compute the empirical DP bounds by Abowd et al. (2013). Hence, a straightforward EDP procedure in our scenario would be the following:

1. train GAN on the original dataset D;
2. remove a random sample from D;
3. re-train GAN on the updated set;
4. estimate probabilities of all outcomes and the maximum privacy loss value;
5. repeat *(1)–(4)* sufficiently many times to approximate μ, γ.

If the generative model is simple, this procedure can be used without modification. Otherwise, for models like GANs, it becomes prohibitively expensive due to repetitive re-training (steps *(1)–(3)*). Another obstacle is estimating the maximum privacy loss value (step *(4)*). To overcome these two issues, we propose the following.

First, to avoid re-training, we imitate the removal of examples directly on the generated set $\widetilde{D}$. We define a similarity metric $sim(x, y)$ between two data points x and y that reflects important characteristics of data (see Section 3.6 for details). For every randomly selected real example i, we remove k nearest artificial neighbours to simulate absence of this example in the training set and obtain $\widetilde{D}^{-i}$. Our intuition behind this operation is the following. Removing a real example would result in a lower probability density in the corresponding region of space. If this change is picked up by a GAN, which we assume is properly trained (e.g. there is no mode collapse), the density of this region in the generated examples space should also decrease. The number of neighbours k is a

hyper-parameter, we defined it by the ratio of artificial and real examples, to keep the densities approximately normalised.

Second, we relax the worst-case privacy loss bound in step *(4)* by the expected-case bound, in the same manner as on-average KL privacy. This relaxation allows us to use a high-dimensional KL divergence estimator (Pérez-Cruz, 2008) to obtain the expected privacy loss for every pair of adjacent datasets $\widetilde{D}$ and $\widetilde{D}^{-i}$ (we denote it by $\mathcal{D}_{KL}^{-i}$, where $i = 1..m$). There are two major advantages of this estimator: it converges almost surely to the true value of KL divergence (see Definition 10); and it does not require intermediate density estimates to converge to the true probability measures. Also since this estimator uses nearest neighbours to approximate KL divergence, our heuristic described above is naturally linked to the estimation method.

Finally, after obtaining sufficiently many samples of different pairs $(\widetilde{D}, \widetilde{D}^{-i})$, we use Chebyshev's inequality to bound the probability $\gamma = \Pr\left(\bar{L}(\tilde{D}) \geq \mu\right)$ of the expected privacy loss estimator exceeding a predefined threshold μ. To deal with the problem of insufficiently many samples, one could use a sample version of inequality (Saw et al., 1984) at the cost of looser bounds.

3.5.3 Limitations

The advantage of ADP, as a definition, over the traditional DP is that it incorporates more information about the data and provides a tighter analysis for typical data points. The disadvantage, however, is that the guarantee is so relaxed that it is difficult to reason about the breadth of the privacy loss distribution.

The second serious drawback of this approach is the empirical privacy estimator. It simulates the removal of training examples using a heuristic approach and the chosen similarity metric. However, if the GAN hasn't properly converged to the data distribution, or if the similarity metric does not reflect privacy-inducing characteristics of the data, the algorithm will yield unrepresentative samples and poor estimation. Furthermore, although the KL divergence estimator is consistent and does not require the data density approximation, it is not robust in high-dimensional spaces with a small number of data points.

Finally, the meaning of our empirical guarantees provided by the *ex post* analysis of the artificial dataset is not equivalent to the traditional formulation of DP, or related notions, and has certain conceptual differences discussed by Charest and Hou (2017) and in the earlier sections. Nevertheless, it may be useful in the situations where strict privacy guarantees are not required or cannot be achieved by existing methods, or when one wants to get a better idea about the expected privacy loss rather than the highly unlikely worst-case.

All these limitations are addressed in Chapter 5, where we design a data-aware privacy notion in a way that enables tight guarantees, a better analysis of the privacy loss distribution, robust estimation, and is more conceptually close to DP.

3.6 Evaluation

In this section, we describe the experimental setup and implementation, and evaluate our method on MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), and CelebA (Liu et al., 2015) datasets.

3.6.1 Experimental Setting

We evaluate our method in four major ways. First, we show that it is possible not only to train ML models purely on generated data, but also achieve high learning performance (Section 3.6.3). Second, we demonstrate an even stronger result: generated data can be used as a validation set for tuning model hyper-parameters (Section 3.6.4). Third, we report Fréchet Inception Distance (FID) (Heusel et al., 2017) between real and generated datasets to underline advantages of the DP critic (Section 3.6.5). Finally, we compute average-case DP bounds for the given datasets and evaluate the artificial data effectiveness against model inversion attacks (Section 3.6.6).

Learning performance experiments are set up as follows:

1. Train a generative model (*teacher*) on the original dataset using only the training split.
2. Generate an artificial dataset by the obtained model and use it to train ML models (*students*).
3. Evaluate students on a held-out test set.

Note that there is no dependency between teacher and student models. Moreover, student models are not constrained to neural networks and can be implemented as any type of machine learning algorithm.

We choose three commonly used image datasets for our experiments: MNIST, SVHN, and CelebA. MNIST is a handwritten digit recognition dataset consisting of 60000 training examples and 10000 test examples, each example is a 28x28 size greyscale image. SVHN is also a digit recognition task, with 73257 images for training and 26032 for testing. The examples are coloured 32x32 pixel images of house numbers from Google Street View. CelebA is a facial attributes dataset with 202599 images, each of which we crop to 128x128 and then downscale to 48x48.

3.6.2 Implementation Details

For our experiments, we use Python and Pytorch[1] framework. We implement, with some minor modifications, a Wasserstein GAN with gradient penalty (WGAN-GP) by Gulrajani et al. (2017). More specifically, the critic consists of four convolutional layers with SELU (Klambauer et al., 2017) activations (instead of ReLU) followed by a fully connected linear layer which outputs a d-dimensional feature vector ($d = 64$). For the DP critic, we implement the Gaussian noise mechanism (Dwork et al., 2014) by clipping the $L2$-norm of this feature vector to $C = 1$ and adding Gaussian noise with $\sigma = 1.5$ (we refer to it as *DP layer*). Finally, it is passed through a linear classification layer. The generator starts with a fully connected linear layer that transforms noise and labels into a 4096-dimensional feature vector which is then passed through a SELU activation and three deconvolution layers with SELU activations. The output of the third deconvolution layer is downsampled by max pooling and normalised with a `tanh` activation function. Both networks are trained using Adam (Kingma and Ba, 2015) with learning rate 10^{-4}, $\beta_1 = 0$, $\beta_2 = 0.9$, and a batch size of 64.

Similarly to the original paper, we use a classical WGAN value function with the gradient penalty that enforces Lipschitz constraint on a critic. We also set the penalty parameter $\lambda = 10$ and the number of critic iterations $n_{\text{critic}} = 5$. Furthermore, we modify the architecture to allow for conditioning WGAN on class labels. Binarised labels are appended to the input of the generator and to the linear layer of the critic after convolutions. Therefore, the generator can be used to create labelled datasets for supervised learning.

The student network is constructed of two convolutional layers with ReLU activations, batch normalisation and max pooling, followed by two fully connected layers with ReLU, and a `softmax` output layer. Note that this network does not achieve state-of-the-art performance on the used datasets, but we are primarily interested in evaluating the relative performance drop compared to a non-private model.

To estimate privacy loss, we carry out the procedure presented in Section 3.4. Specifically, based on recent ideas in qualitative evaluation of images, such as FID and Inception Score, we compute image features with the pre-trained InceptionV3 network (Szegedy et al., 2016) and use inverse distances between these features as the *sim* function. We implement the KL divergence estimator (Pérez-Cruz, 2008) and use k-d trees (Bentley, 1975) for fast nearest neighbour searches. For privacy evaluation, we implement the model inversion attack (Fredrikson et al., 2015).

[1] http://pytorch.org

Table 3.1 – Accuracy of student models for non-private baseline, PATE (Papernot et al., 2016), and our method.

Dataset	Non-private	PATE	Our approach
MNIST	99.2%	98.0%	98.3%
SVHN	92.8%	82.7%	87.7%

3.6.3 Learning Performance

First, we evaluate the generalisation ability of a student model trained on artificial data. More specifically, we train a student model on generated data and report test classification accuracy on a held-out real set.

As noted above, most of the work on privacy-preserving ML focuses on *model release* methods and assumes (explicitly or implicitly) access to similar "public" data in one form or another (Abadi et al., 2016; Papernot et al., 2016, 2018; Zhang et al., 2018). On the other hand, existing *data release* solutions struggle with high-dimensional data (Zhu et al., 2017). It limits the choice of methods for comparison.

We chose to compare learning performance with the current state-of-the-art model release technique, PATE by Papernot et al. (2018), which uses a relatively small set of unlabelled publicly available data. Since our approach does not require any public data, in order to make the evaluation more appropriate, we pick the results of PATE corresponding to the least number of labelling queries. It is worth noting, however, that one should keep in mind the difference in the privacy guarantee strength when interpreting these learning performance results.

Table 3.1 shows test accuracy for the non-private baseline model (trained on the real training set), PATE, and our method. We observe that artificial data allows us to achieve 98.3% accuracy on MNIST and 87.7% accuracy on SVHN, which is comparable or better than corresponding results of PATE. These results demonstrate that our approach does not compromise learning performance, and may even improve it, while enabling the full flexibility of data release methods.

Additionally, we train a simple logistic regression model on artificial MNIST samples, and obtain 91.69% accuracy, compared to 92.58% on the original data, confirming that student models are not restricted to a specific type.

3.6.4 Validation Performance

In the previous section, we demonstrated that ML models trained on artificial data can generalise well enough and achieve high accuracy on unseen real data. However, there is another important aspect of training: choosing the right hyper-parameters. In scenarios

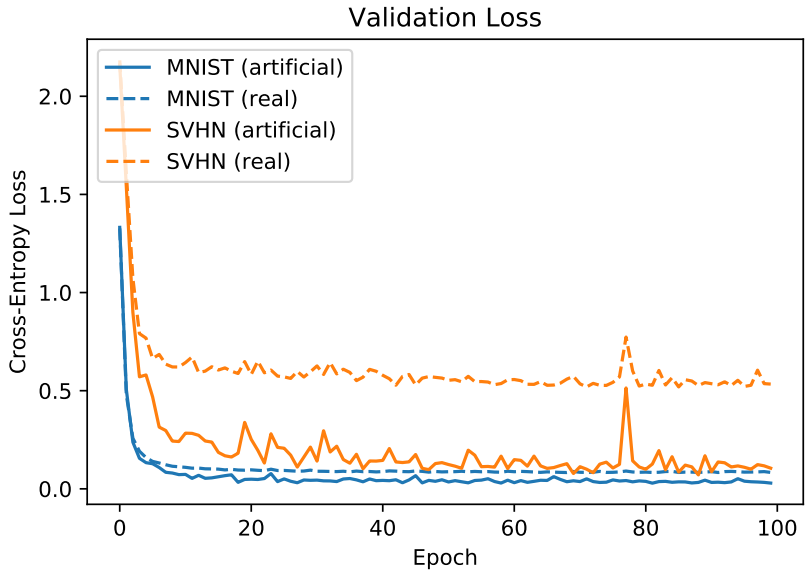


Figure 3.2 – Cross-entropy loss for real and artificial validation sets (SGD with learning rate 0.001).

where public data is not available, validation has to be done on private artificial data, and to the best of our knowledge, the question of using artificial data for validation has not been well covered in the machine learning literature.

We evaluate the validation power in the following way. While training a student model, we compute validation loss on a real held-out set and on an artificial held-out set. We then compute correlation between the two sequences. Figure 3.2 shows two pairs of validation loss curves, for MNIST and SVHN datasets. We observe that, indeed, artificial validation loss closely follows the real one, despite being generally lower and fluctuating more. Note that lower validation loss does not imply better test performance, but high correlation is important for hyper-parameter tuning. We ran experiments for a number of different learning rates, and correlation coefficients range from 0.7197 to 0.9972 for MNIST and from 0.8047 to 0.9810 for SVHN.

3.6.5 Visual Quality of Generated Samples

While the main purpose of this work is to evaluate and improve privacy of generated data, we observe that addition of DP layer in the critic has a beneficial side effect: improving image quality and diversity, as well as providing regularisation effect which allows for much longer stable training.

Figure 3.3 shows FID values for every 10-th epoch of training with and without DP layer. For both SVHN and CelebA, WGAN-GP with DP critic achieves better performance

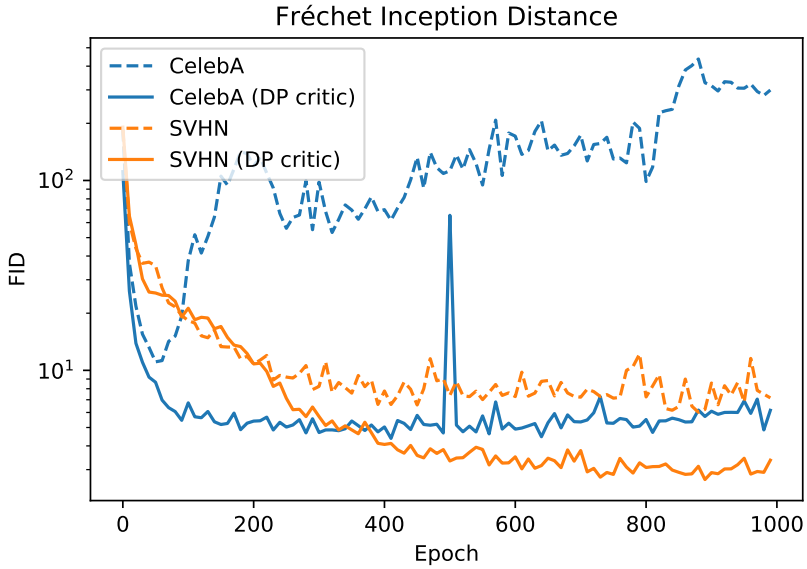


Figure 3.3 – Fréchet Inception Distance between real and generated data for WGAN-GP with and without DP critic.

and converges more stably. At the same time, the quality of CelebA samples for vanilla WGAN-GP significantly degrades after 100 epochs indicating overfitting (note that for privacy evaluation we chose the epoch with the best FID score). Moreover, GANs with DP critics achieve better FID scores for given datasets than the best ones reported in (Heusel et al., 2017).

3.6.6 Privacy Analysis

Using the privacy estimation framework (see Section 3.5.2), we fix the probability γ of exceeding the expected privacy loss bound μ in all experiments to 10^{-5} and compute the corresponding μ for each dataset and two versions of WGAN-GP (vanilla and with DP critic). Table 3.2 encapsulates our findings. It is worth noting, that our μ should not be viewed as an empirical estimation of ε of DP, since the former bounds *expected* privacy loss, while the latter *maximum*. These two quantities, however, in our experiments turn out to be similar to deep learning DP bounds found in recent literature (Abadi et al., 2016; Papernot et al., 2018). This may be explained by tight concentration of privacy loss random variable (Dwork and Rothblum, 2016) or loose estimation. Additionally, DP critic helps to bring down μ values in all cases.

The lack of theoretical privacy guarantees for our method neccesitates assessing the strength of provided protection. We perform this evaluation by running the *model inversion attack* (Fredrikson et al., 2015) on a student model. Note that we also experimented with another well-known attack on machine learning models, the membership

Table 3.2 – Empirical privacy parameters: expected privacy loss bound μ and probability γ of exceeding it.

Dataset	Method	μ	γ
MNIST	WGAN-GP	5.80	10^{-5}
	WGAN-GP (DP critic)	5.36	
SVHN	WGAN-GP	13.16	
	WGAN-GP (DP critic)	4.92	
CelebA	WGAN-GP	6.27	
	WGAN-GP (DP critic)	4.15	

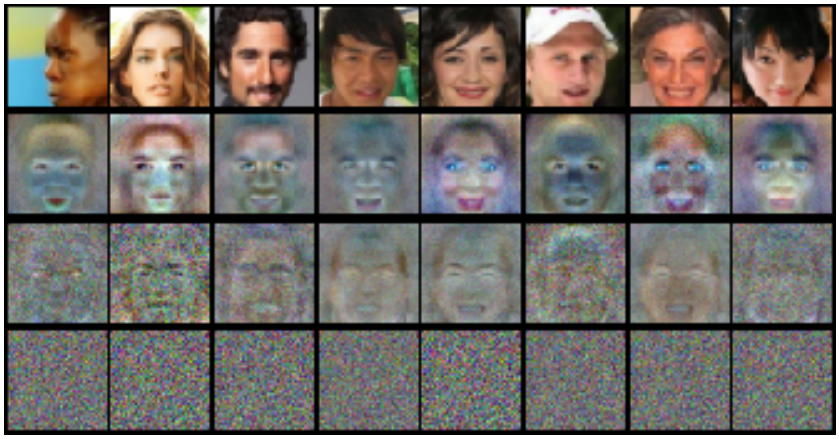


Figure 3.4 – Results of the model inversion attack. Top to bottom: real target images, reconstructions from non-private model, our method, and DP model.

inference (Shokri et al., 2017). However, we did not include it in the final evaluation, because of the poor attacker's performance in our setting (nearly random guess accuracy for given datasets and models even without any protection). Both attacks are performed by *passive adversaries*. Apart from that, one could evaluate our method against *active adversaries* (e.g. Hitaj et al. (2017)), but we do not investigate such attacks in this thesis, instead aiming for more rigorous theoretical guarantees in Part II.

In order to run the attack, we train a student model (a simple multi-layer perceptron with two hidden layers of 1000 and 300 neurons) in three settings: real data, artificial data generated by GAN (with DP critic), and real data with differential privacy (using DP-SGD with a small $\varepsilon < 1$). As facial recognition is a more privacy-sensitive application, and provides a better visualisation of the attack, we picked CelebA attribute prediction task to run this experiment.

Figure 3.4 shows the results of the model inversion attack. The top row presents the real target images. The following rows depict reconstructed images from a non-private model, a model trained on GAN samples, and DP model, correspondingly. One can

Table 3.3 – Face detection and recognition rates (pairs with distances below 0.99) for non-private, our method, and DP.

	Non-private	Our approach	DP
Detection	63.6%	1.3%	0.0%
Recognition	11.0%	0.3%	–

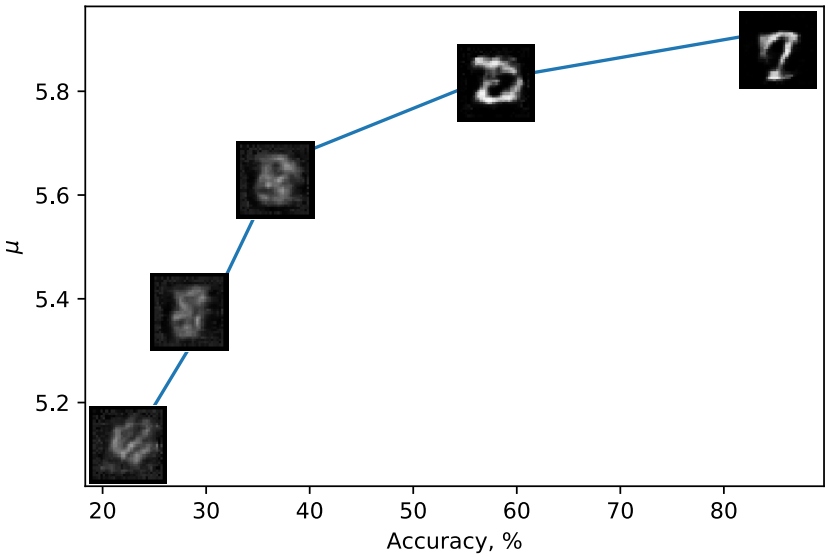


Figure 3.5 – Privacy-accuracy trade-off curve and corresponding image reconstructions from a multi-layer perceptron trained on artificial MNIST dataset.

observe a clear information loss in reconstructed images going from non-private model, to artificial data, to DP. The latter is superior in decoupling the model and the training data, and is a preferred choice in the model release setting and/or if public data is accessible for pre-training. The non-private model, albeit trained with abundant data ($\sim 200K$ images) reveals facial features, such as skin and hair colour, expression, etc. Our method, despite failing to conceal general shapes in training images (i.e. faces), seems to achieve a trade-off, hiding most of the specific features. The obtained reconstructions are either very noisy (columns 1, 2, 6, 8), much like DP, or converge to some average feature-less faces (columns 4, 5, 7).

We also analyse real and reconstructed image pairs using OpenFace (Amos et al., 2016) (see Table 3.3). It confirms our initial findings: in images reconstructed from a non-private model, faces were detected (recognised) 63.6% (11%) of the time, while for our method, detection succeeded only in 1.3% of cases and recognition rate was 0.3%, well within state-of-the-art error margins. For DP both rates were at 0%.

To evaluate our privacy estimation method, we look at how the privacy loss bound μ correlates with the success of the attack. Figure 3.5 depicts the privacy-accuracy

(a) Generated (b) Real

Figure 3.6 – Generated and closest real examples for SVHN.

trade-off curve for an MLP (64-32-10) trained on artificial data. In this setting, we use a stacked denoising autoencoder to compress images to 64-dimensional feature vectors and facilitate the attack performance. Along the curve, we plot examples of the model inversion reconstruction at corresponding points. We see that with growing μ, meaning lower privacy, both model accuracy and reconstruction quality increase. However, the value of μ does not change dramatically, indicating potentially loose estimates.

Finally, as an additional measure, we perform visual inspection of generated examples and corresponding nearest neighbours in real data. Figures 3.6 and 3.7 depict generated and the corresponding most similar real images from SVHN and CelebA datasets. We observe that, despite general visual similarity, generated images differ from real examples in details, which is normally more important for privacy. For SVHN, digits vary either in shape, colour or surroundings. A lot of pairs come from different classes. For CelebA, the pose and lighting may be similar, but such details as gender, skin colour, facial features are usually significantly different.

3.7 Conclusions

We investigate the problem of private data release for complex high-dimensional data. In contrast to commonly studied model release setting, this approach enables important advantages and applications, such as data pooling from multiple sources, simpler development process, and data trading.

We employ generative adversarial networks to produce artificial privacy-preserving

(a) Generated (b) Real

Figure 3.7 – Generated and closest real examples for CelebA.

datasets. The choice of GANs as a generative model ensures scalability and makes the technique suitable for real-world data with complex structure. Unlike many prior approaches, our method does not assume access to similar publicly available data. In our experiments, we show that student models trained on artificial data can achieve high accuracy on MNIST and SVHN datasets. Moreover, models can also be validated on artificial data.

We propose a novel privacy definition and a technique for post hoc privacy analysis of the released data by bounding an estimator of the expected privacy loss. Our privacy notion equates to a statistical measure of sensitivity of the synthetic data to changes in the original data. We compute privacy bounds for samples from WGAN-GP on MNIST, SVHN, and CelebA, and demonstrate that expected privacy loss is bounded by single-digit values. To evaluate the provided protection, we run a model inversion attack and show that training with GAN samples reduces information leakage and that attack success correlates with estimated privacy bounds. For instance, in the face reconstruction example, face detection rates drop from 63.6% to 1.3%.

Additionally, we introduce a simple modification to the critic: differential privacy layer. Not only does it improve privacy loss bounds and ensures DP guarantees for the critic output, but it also acts as a regulariser, improving stability of training, and quality and diversity of generated images.

4 Federated Generative Privacy

4.1 Introduction

Aside from differntial privacy, another method that tackles privacy issues in machine learning is rapidly gaining popularity—the recent concept of *federated learning (FL)* (McMahan et al., 2016). In the FL setting, a central entity (*server*) trains a model without actually collecting user data. Instead, users (*clients*) update models locally, and the *server* aggregates these models. One popular approach is the federated averaging, `FedAvg` (McMahan et al., 2016), where *clients* do on-device gradient descent using their data, then send these updates to the *server* where they get averaged. Privacy can be enhanced by using secure multi-party computation (MPC) to disallow the server access individual updates before averaging (Bonawitz et al., 2017).

Despite many advantages, federated learning does have a number of challenges. First, the result of FL is a single trained model (therefore, we categorise it as a *model release* method), which reduces its flexibility. For instance, it would limit possibilities for further aggregation of information from different sources in hierarchical scenarios, e.g. different hospitals trying to combine federated models trained on their patients data. Second, this solution requires data to be labelled at the source, which is not always possible, because users may not be qualified to label their data or unwilling to do so. A good example is again a medical application where users are unqualified to diagnose themselves but at the same time would want to keep their medical condition and other information private. Third, in spite of a popular opinion circulating in non-scientific and even some scientific publications, federated learning does not offer formal privacy guarantees and *is vulnerable to attacks*, such as the model inversion attack (Fredrikson et al., 2015).

Some papers propose to augment FL with differential privacy (DP) (McMahan et al., 2017) to alleviate this issue and provide rigorous privacy guarantees. While these approaches

This chapter is based on the paper published in IEEE Intelligent Systmes (Triastcyn and Faltings, 2020b).

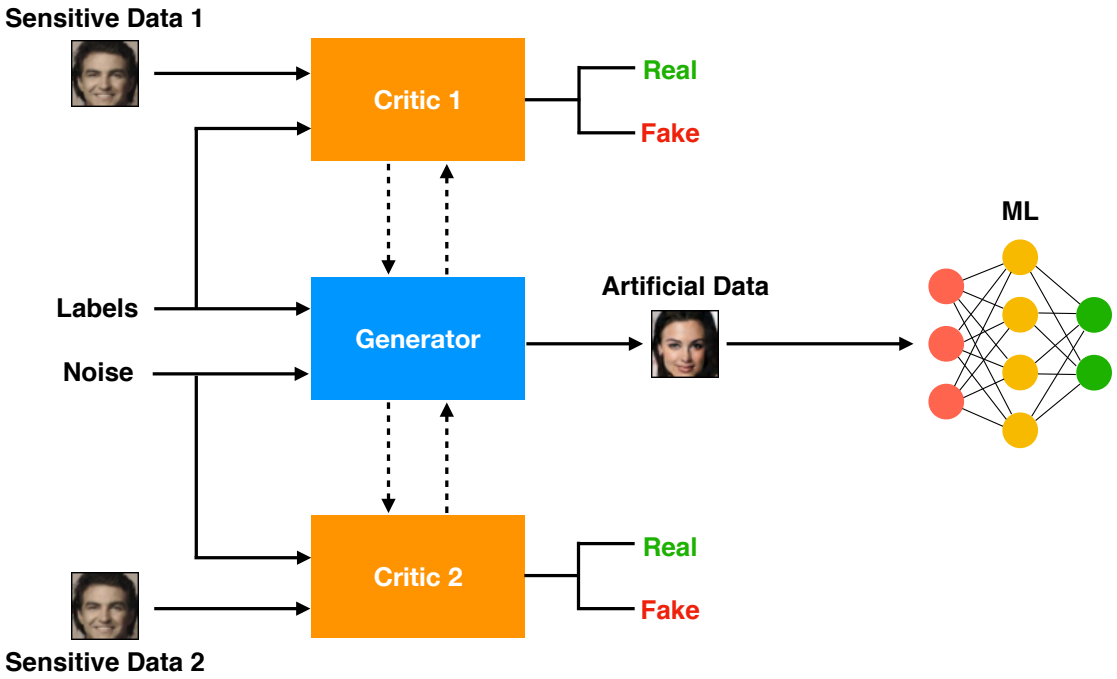


Figure 4.1 – Architecture of our solution for two clients. Sensitive data is used to train a GAN (local critic and federated generator) to produce a private artificial dataset.

perform well in ML tasks and provide theoretical privacy guarantees, they are often restrictive. The major issue of this approach, exacerbated by the fact that we task ourselves with building generative models as opposed to discriminative, is that obtaining reasonable DP guarantees requires a noticeable downgrade in model quality. This is evident in a recent work by Augenstein et al. (2019), which we consider in more detail in Chapter 7. As mentioned in the previous chapter, in order to alleviate this issue, many research papers in this area assume, implicitly or explicitly, access to public data of similar nature or abundant amounts of data, which is not always realistic.

We address these problems by combining the strengths of federated learning with our GAN-based framework for privacy-preserving data release introduced earlier. The main idea of our approach, named `FedGP`, for *federated generative privacy*, is to train generative adversarial networks (GANs) (Goodfellow et al., 2014) on clients to produce artificial data that can replace clients real data. Since some clients may have insufficient data to train a GAN locally, we instead train a federated GAN model. This way, user data always remain on their devices. Moreover, the federated GAN will produce samples from the common cross-user distribution and not from a single user, which adds to overall privacy. Figure 4.1 depicts the schematics of our approach.

Similarly to its centralised variant, this approach allows releasing entire datasets, enabling many immediate advantages compared to the model release techniques. First, the released data could be used to train any ML model (we refer to it as *downstream task* or *downstream*

model) without additional assumptions. Second, data from different sources could be easily pooled, allowing for hierarchical aggregation and building stronger models. Third, labelling and verification can be done later down the pipeline, relieving some trust and expertise requirements on users. Fourth, released data could be traded on data markets[1], where anonymisation and protection of sensitive information is one of the biggest obstacles. Finally, data publishing would facilitate transparency and reproducibility of research.

To evaluate potential privacy risks, we adapt the Average-Case Differential Privacy (ADP) notion and the improved version of the post hoc privacy estimation routine from Section 3.5. Its key idea is to estimate Kullback-Leibler (KL) divergence between pairs of synthetic data distributions produced by GANs with one-point difference in the original dataset.

Our contributions in this chapter are the following:

- we extend our approach for private data release to the federated setting, broadening its applicability and enhancing privacy;
- we tune the federated learning protocol for GANs to allow a range of benefits, such as reduced communication costs;
- we demonstrate that downstream models trained on artificial data achieve high accuracy while maintaining good average-case privacy and resilience to model inversion attacks.

4.2 Related Work

McMahan et al. (2016) proposed federated learning as one possible solution for privacy protection, among other issues, such as scalability and communication costs. In this setting, privacy is enforced by keeping data on user devices and only submitting model updates to the server. Two of the most popular approaches are the federated stochastic gradient descent (`FedSGD`) and federated averaging (`FedAvg`) (McMahan et al., 2016), where *clients* do local on-device gradient descent using their data, then send these updates to the *server*, which applies a weighted average update to the model. Federated learning offers significant advantages in certain scenarios, and thus, is being rapidly developed and further improved by both industry and academia (see, for example, Konečnỳ et al. (2016); Bonawitz et al. (2019)). Yang et al. (2019) offers a more comprehensive overview of different aspects of FL.

It is crucial to note that, in spite of the privacy-oriented design, federated learning in itself does not offer any theoretical privacy guarantees. For this reason, it is often combined with privacy-preserving mechanisms. Bonawitz et al. (2017), for example, propose to

[1] https://www.datamakespossible.com/value-of-data-2018/dawn-of-data-marketplace

enahnce privacy by an efficient secure aggregation algorithm based on secure multi-party computation (MPC) (Yao, 1982). This allows the server to access only average updates of a big group of users and not individual ones. Unfortunately, methods like MPC and homomorphic encryption do not necessarily guarantee robustness against privacy attacks on machine learning models, such as model inversion or membership inference (Fredrikson et al., 2015; Shokri et al., 2017), due to the fact that these attacks operate on the final model which remains the same. Essentially, these methods would protect the training process, but not the product of this process. It is also possible to employ differential privacy in this context, and it has been done, but we defer this discussion until Chapter 6.

So far, there has been little work on GANs, or even generative models in general, in federated settings. In one of the earliest papers, Chen et al. (2019) train a character-level RNN using FL to generate out-of-vocabulary words to help track temporal changes in the word usage frequencies. Their model is used by Augenstein et al. (2019) to evaluate their federated debugging solution, based on GANs and providing differential privacy. This work is perhaps the closest in spirit to what we want to achieve, but in order to preserve the chronological order, we leave a more detailed discussion of this paper to Chapter 7, where we augment their solution with more practical privacy guarantees.

Finally, Hardy et al. (2019) present a distributed scheme for training GANs and evaluate it against the naïve implementation of federated GANs. The distinctive characteristic of their algorithm is that discriminators (critics) are being exchanged between clients in a peer-to-peer fashion, in contrast to the conventional FL scheme of building averaged models on the server. The authors claim better learning results and improved computation complexity on the clients compared to the naïve solution. The downside of this approach is a larger possibility of privacy leaks due to discriminator exchange. Moreover, in Section 4.5, we show that the increased risk translates to only marginal improvements in accuracy and only in i.i.d. settings, and thus, might not be justified.

4.3 Preliminaries

As this chapter largely relies on the concepts introduced earlier in this thesis, we encourage the reader to consult Chapters 2 and 3 for preliminary information.

Additionally, we will use the Bayesian perspective on estimating mean from the data to get sharper bounds on the expected privacy loss compared to the original privacy estimation framework from Section 3.5.2.

Proposition 1. *Let $[l_1, l_2, \ldots, l_m]$ be a random vector drawn from a real-valued distribution $p(L)$, defined on $(-\infty, +\infty)$, with the existing common mean and variance, and let $\overline{L}$ and S be the sample mean and the sample standard deviation of the random variable*

L. Then, in the absence of any other information about $p(L)$, we can claim

$$\Pr\left(\mathbb{E}[L] > \overline{L} + \frac{F_{m-1}^{-1}(1-\gamma)}{\sqrt{m-1}}S\right) \leq \gamma, \tag{4.1}$$

where $F_{m-1}^{-1}(1-\gamma)$ is the inverse CDF of the Student's t-distribution with $m-1$ degrees of freedom at $1-\gamma$.

We defer a more detailed analysis and the proof of this proposition to Chapter 5. However, the sketch of the proof is as follows. Assuming the existence of the common mean and variance, we can use the maximum entropy principle for the likelihood function of these samples to ensure the highest uncertainty, and thus, conservativeness of the estimate. Combined with a flat prior, this likelihood function gives us the marginal distribution of the true mean $\mathbb{E}[L]$, and we observe that the random variable $\frac{\mathbb{E}[L]-\overline{L}}{S/\sqrt{m-1}}$ follows the Student's t-distribution with $m-1$ degrees of freedom (Oliphant, 2006). We can then use the inverse of the Student's t CDF to arrive to Proposition 1.

4.4 Federated Generative Privacy

In order to keep participants data private while still maintaining flexibility in downstream tasks, our algorithm produces a federated generative model. This model can output artificial data, not belonging to any real user in particular, but coming from the common cross-user data distribution.

Let $\{u_1, u_2, \ldots, u_n\}$ be a set of *clients* holding private datasets $\{d_1, d_2, \ldots, d_n\}$. Before starting the training protocol, the *server* is providing each *client* with generator G_i^0 and critic C_i^0 models, and *clients* initialise their models randomly. Like in a normal FL setting, the training process afterwards consists of communication rounds. In each round t, *clients* update their respective models performing one or more passes through their data and submit generator updates $\triangle G_i^t$ to the *server* through MPC while keeping C_i^t private. In the beginning of the next round, the *server* provides an updated common generator G^t to all *clients*.

This approach has important advantages:

- Data do not physically leave user devices.
- Only generators (that do not come directly into contact with data) are shared, and critics remain private.
- Using artificial data in downstream tasks adds another layer of protection and limits information leakage to artificial samples.

4.4.1 Privacy Estimation

What remains to assess is how much information would an attacker gain about the original data. We do so by employing average-case differential privacy (Definition 11) and the corresponding empirical estimation framework (see Section 3.5.2). However, we modify the last step of the routine, which was using Chebyshev's inequality to obtain the privacy parameters estimates.

In particular, having obtained sufficiently many sample pairs $(\widetilde{D}, \widetilde{D}^{-i})$, we use Proposition 1 to determine the ADP parameters μ and γ. We fix γ at the desired level (generally, inversely proportional to the number of data points), and then compute

$$\mu = \overline{L} + \frac{F_{m-1}^{-1}(1-\gamma)}{\sqrt{m-1}} S, \tag{4.2}$$

where $\overline{L}$ and S are the sample mean and the sample standard deviation of $\{\mathcal{D}_{KL}^{-i}\}$.

It is worth noting that this modification leads to a somewhat different interpretation of the definition and parameters μ and γ. More specifically, expectation is now taken not only over the outcomes, but also over the data, altering the meaning of the parameters. In the example of Section 3.5.1, $(0.01, 0.001)$-ADP would now be interpreted as follows: with probability 0.999, an average user from the same distribution submitting their data will change outcome probabilities of the private algorithm on average by 1% (because $e^{0.01} \approx 1.01$.). As we demonstrate in our evaluation, this revision, effectively discounting data outliers in the privacy quantification process, yields significant improvements in the bounds for more ordinary data points.

4.4.2 Limitations

All the limitations of the centralised version of the framework, discussed in Sections 3.4.2 and 3.5.3, still apply to `FedGP`. The only area where it is different is the probability bound obtained using a Bayesian approach instead of Chebyshev's inequality.

Additionally, there is a potentially negative effect on the model quality due to federation. Since critics remain private and do not leave user devices their performance can be hampered by a small number of training examples. Nevertheless, we observe that even in the settings where some users have smaller datasets the overall discriminative ability of all critics is sufficient to train good generators.

4.5 Evaluation

We evaluate two major aspects of our method: downstream learning performance and privacy. Similarly to the previous chapter, we first show that training ML models on

Table 4.1 – Accuracy of student models trained on artificial samples of `FedGP` compared to non-private centralised baseline and `CentGP`. In parenthesis we specify the average number of data points per client.

Setting	Dataset	Baseline	MD-GAN	CentGP	FedGP
i.i.d.	MNIST (500)	98.10%	64.30%	97.35%	79.45%
	MNIST (1000)	98.55%	93.46%	97.39%	93.38%
	MNIST (2000)	98.92%	97.47%	97.41%	96.23%
non-i.i.d.	MNIST (500)	97.31%	79.23%	—	83.26%
	MNIST (1000)	98.78%	91.90%		95.89%
	MNIST (2000)	98.76%	95.18%		96.88%

data created by the common generator achieves high accuracy on MNIST (LeCun et al., 1998). Then, we estimate expected privacy loss of the federated GAN and evaluate the effectiveness of artificial data against the model inversion attack on CelebA face attributes dataset (Liu et al., 2015). Note that we do not repeat the evaluation of some secondary aspects featured in Section 3.6.

Our implementation to a large extent follows Section 3.6.1. To train the federated generator we use `FedAvg` algorithm (McMahan et al., 2016). As a *sim* function, introduced in Section 3.5.2, we once again use the distance between InceptionV3 feature vectors (Szegedy et al., 2016).

4.5.1 Learning Performance

First, we evaluate the generalisation ability of the student model trained on artificial data. Equivalently to the previous chapter, the experiment adheres to the following steps:

1. Train the federated generative model (*teacher*) on the original distributed data.
2. Generate an artificial dataset and use it to train ML models (*students*).
3. Evaluate students on a held-out test set.

We compare learning performance with the baseline centralised discriminative model trained on the original data, as well as the same model trained on artificial samples obtained from the centrally trained GAN, introduced in Chapter 3, denoted here as `CentGP`. Furthermore, we include MD-GAN (Hardy et al., 2019) in our comparison—another distributed GAN approach, differing from our federated GAN by the fact that critics are randomly exchanged between clients in a peer-to-peer fashion.

Since critics stay private in `FedGP` and only access data of a single user, the size of each

Table 4.2 – Average-case privacy parameters: expected privacy loss bounds μ_C and μ_F (for centralised and federated solutions), and probability γ of exceeding it. A typical ε of DP in this setting is > 2.

Setting	**Dataset**	μ_C	μ_F	γ
i.i.d.	MNIST (500)	0.0101	0.0117	10^{-15}
	MNIST (1000)	0.0046	0.0069	
	MNIST (2000)	0.0015	0.0021	
	CelebA	0.0009	0.0009	
non-i.i.d.	MNIST (500)	–	0.0090	10^{-15}
	MNIST (1000)	–	0.0044	
	MNIST (2000)	–	0.0020	

individual dataset has significant effect. Therefore, in our experiment we vary sizes of user datasets and observe its influence on training. In each experiment, we specify an average number of points per user, while the actual number is drawn from the uniform distribution with this mean, with some clients getting as few as 100 data points.

We also study two settings: i.i.d. and non-i.i.d data. In the first setting, distribution of classes for each client is identical to the overall distribution. In the second, every client gets samples of 2 random classes, imitating the situation when a single user observes only a part of overall data distribution.

Details of the experiment can be found in Table 4.1. We observe that training on artificial data from the federated GAN allows to achieve 96.9% accuracy on MNIST with the baseline of 98.8%. We can also see how accuracy grows with the average user dataset size. A less expected observation is that non-i.i.d. setting is actually beneficial for `FedGP`. A possible reason is that training critics with little data becomes easier when this data is less diverse (i.e. the number of different classes is smaller).

We find that the performance of MD-GAN is similar to `FedGP` in the i.i.d. case and is slightly behind in the non-i.i.d. case. Therefore, we believe that the additional privacy leakage and the extra communication complexity of MD-GAN associated with the critics exchange are not justified in the examined setting. Comparing to the centralised generative privacy model `CentGP`, we can see that `FedGP` is more affected by sharding of data on user devices than by overall data size, suggesting that further research in training federated generative models is necessary.

4.5.2 Privacy Analysis

Analogously to Section 3.6.6, we employ the privacy estimation framework to compute the expected privacy loss bound μ by fixing the probability $\gamma = 10^{-15}$. Note the significant

Table 4.3 – Face detection and recognition rates (pairs with distances below 0.99) for images recovered by model inversion from the non-private baseline and the `FedGP`-trained model.

	Baseline	**FedGP**
Detection	25.5%	1.2%
Recognition	2.8%	0.1%

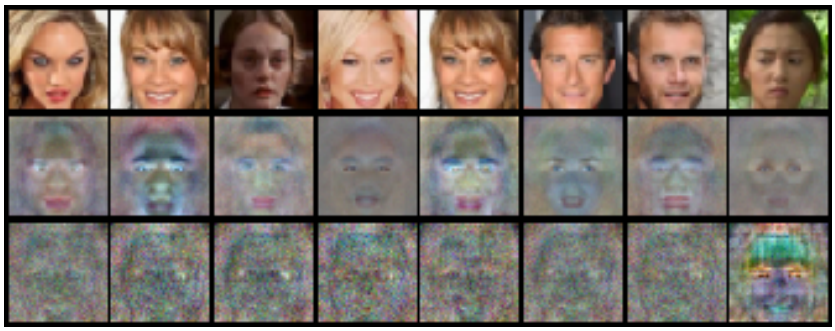


Figure 4.2 – Results of the model inversion attack. Top to bottom: real target images, reconstructions from the non-private model, reconstructions from the model trained by `FedGP`.

reduction in γ due to the use of Proposition 1.

Table 4.2 summarises the bounds we obtain. As anticipated, the privacy guarantee improves with the growing number of data points, because the influence of each individual example diminishes. Moreover, the average privacy loss μ, expectedly, is significantly smaller than the typical worst-case DP loss ε in similar settings (between 2 and 10, or even larger). To put it in perspective, the average change in outcome probabilities estimated by ADP is $\sim$1% even in more difficult settings, while the state-of-the-art DP method would place the worst-case change at $> 100\%$ or even $> 1000\%$ without giving much information about a typical case. Compared to the centralised solution (μ_C), the federated version may have slightly weaker privacy guarantees, probably because of the higher degree of overfitting for critics. But this difference diminishes with growing data size, and for CelebA μ_F actually gets smaller than μ_C.

On top of estimating the expected privacy loss bounds, we repeat our test of resistance to the *model inversion attack* (Fredrikson et al., 2015) for `FedGP`, running the attack on two student models: the one trained on the original data samples and the other one trained on the artificial samples. Once again, we only consider *passive adversaries* and we leave evaluation with active adversaries for future work.

Similarly to the previous chapter, we pick the CelebA attribute prediction task to run this experiment, because facial recognition is a more privacy-sensitive application and

offers a better visualisation of the attack. This time, we train a student model (again, a simple MLP with two hidden layers of 1000 and 300 neurons) in two settings: the real data and the artificial data generated by the federated GAN.

We analyse real and reconstructed image pairs using OpenFace (Amos et al., 2016) (see Table 4.3). It confirms our theory that artificial samples would shield real data in case of the downstream model attack analogously to the centralised setting. In the images reconstructed from a non-private model, faces were detected 25.5% of the time and recognised in 2.8% of cases. For our method, detection succeeded only in 1.2% of faces and the recognition rate was 0.1%, well within the state-of-the-art error margin for face recognition.

Figure 4.2 shows results of the model inversion attack. The top row presents the real target images. The following rows depict reconstructed images from the non-private model and the model trained on the federated GAN samples. One can observe a clear information loss in reconstructed images going from the non-private to the `FedGP`-trained model. Despite failing to conceal general shapes in training images (i.e. faces), our method seems to achieve a trade-off, hiding most of the specific features, while the non-private model reveals important facial features, such as skin and hair colour, expression, etc. The obtained reconstructions are either very noisy or converge to some average feature-less faces.

4.6 Conclusions

In this chapter, we studied the intersection of federated learning and private data release using GANs. Combined, these methods enable important advantages and applications for both fields, such as higher flexibility, reduced requirements on trust and expertise for users, hierarchical data pooling, and data trading.

The choice of GANs as a generative model ensures scalability and makes the technique suitable for real-world data with complex structure. In our experiments, we show that student models trained on artificial data can achieve high accuracy on classification tasks, even without access to similar publicly available data.

We estimate and bound the expected privacy loss of an average client by using differential average-case privacy thus enhancing privacy of traditional federated learning. We find that, in most scenarios, the presence or absence of a single data point would not change the outcome probabilities by more than 1% on average. Additionally, we evaluate the provided protection by running the model inversion attack and showing that training with datasets generated by federated GANs reduces information leakage (e.g. face detection in recovered images decreases from 25.5% to 1.2%).

Bayesian Differential Privacy Part II

5 Bayesian Differential Privacy

5.1 Introduction

Let us take a step back from privacy-preserving data release, and instead, consider privacy in machine learning in a more general sense. We have already established that *differential privacy (DP)* (Dwork, 2006; Dwork et al., 2006a,b) is a desirable guarantee, but is very hard to achieve in many machine learning tasks without a marked decline in the model quality. In this chapter, we attempt to answer why it is the case and how to provide a significantly better privacy-utility trade-off.

Historically, DP algorithms were introduced for database privacy and focused on sanitising simple statistics, such as a mean, a median, and so on, using a technique known as output perturbation. In recent years, the field made a lot of progress towards the goal of privacy-preserving machine learning, through works on objective perturbation (Chaudhuri et al., 2011), stochastic gradient descent with DP updates (Song et al., 2013), to more complex and practical techniques (Abadi et al., 2016; Papernot et al., 2016, 2018; McMahan et al., 2018).

However, borrowing the theoretical underpinnings from database privacy ignored specifics of ML and AI applications, such as a focus on a given task or a particular data distribution. As a result, despite significant advances, differentially private machine learning still suffers from two major problems: (a) utility loss due to excessive amounts of noise added during training and (b) difficulty in interpreting the privacy parameters ε and δ. In many cases where the first problem appears to be solved, it is actually being hidden by the second problem. To illustrate it, we design a motivational example in Section 5.3.3 that shows how a seemingly strong privacy guarantee turns out to allow for the attacker accuracy to be as high as 99%. Even considering that this guarantee is very pessimistic and holds against a very powerful adversary with any auxiliary information, it can hardly be viewed

This chapter is based on the paper accepted for publication at the 37th International Conference on Machine Learning (ICML 2020) (Triastcyn and Faltings, 2020a).

as a reassurance to a user. Moreover, it provides only the worst-case bound, leaving users to wonder how far is the worst-case from a typical case.

In this thesis, we focus on practicality of privacy guarantees in the context of machine learning, and therefore, we propose a variation of differential privacy that provides more meaningful guarantees for *typical* scenarios on top of the global differential privacy guarantee. We name it *Bayesian differential privacy (BDP)*.

The key to our relaxation is our definition of *typical* scenarios. At the core of it lies the observation that machine learning models are designed and tuned for a particular data distribution (for example, an MRI dataset is very unlikely to contain a picture of a car). Such prior distribution of data is also often available to the attacker. We consider a scenario *typical* when all sensitive data is drawn from the same distribution. While the traditional differential privacy treats all data as equally likely and hides differences by large amounts of noise, Bayesian differential privacy calibrates noise to the data distribution. Thus, for any two datasets drawn from the same distribution, and given the same privacy mechanism with the same amount of noise, BDP guarantees are tighter than DP guarantees. It is important to note that this data distribution can be *unknown*, and the necessary statistics can be estimated from data as shown in the following sections.

To accompany the notion of Bayesian differential privacy (Section 5.4.1), we provide its theoretical analysis and the privacy accounting framework (Section 5.4.2). The latter considers the privacy loss random variable and employs principled tools from probability theory to find concentration bounds on it. It provides a clean derivation of privacy accounting in general (Sections 5.4.2 and 5.4.5), as well as in the special case of subsampled Gaussian noise mechanism (Section 5.4.3). Further, we show that it is a generalisation of the well-known moments accountant (MA) (Abadi et al., 2016) (Section 5.4.6).

Since our privacy accounting relies on data distribution samples, a natural concern would be that the data not present in the dataset are not taken into account, and thus, are not protected. However, this is not the case, because our finite sample estimator is specifically designed to address this issue (see Section 5.4.5).

This chapter contains the following key contributions:

- we propose a relaxation of DP, called Bayesian differential privacy, that allows to provide more practical privacy guarantees in a wide range of scenarios;
- we derive a clean, principled method for privacy accounting in learning that generalises the moments accountant;
- we experimentally demonstrate advantages of our method (Section 5.5), including the state-of-the-art privacy bounds in deep learning applications (Section 5.5.2) and

variational inference (Section 5.5.3), a popular class of algorithms rarely considered in privacy research.

5.2 Related Work

Differential privacy (Dwork, 2006; Dwork et al., 2006b) is one of the strongest privacy standards that can be employed to protect ML models from these and other attacks. Since pure ε-DP is hard to achieve in many realistic learning settings, a notion of approximate (ε, δ)-DP is used across-the-board in machine learning. It is often achieved as a result of applying the Gaussian noise mechanism (Dwork et al., 2014). Several other alternative notions and relaxations of DP have also been proposed, such as probabilistic DP (Machanavajjhala et al., 2008), computational DP (Mironov et al., 2009), mutual-information privacy (Mir, 2012; Wang et al., 2016a), different versions of concentrated DP (CDP) (Dwork and Rothblum, 2016), zCDP (Bun and Steinke, 2016), tCDP (Bun et al., 2018)), and Rényi DP (RDP) (Mironov, 2017). Some other relaxations (Abowd et al., 2013; Schneider and Abowd, 2015; Charest and Hou, 2017; Wang et al., 2016b; Triastcyn and Faltings, 2019c) tip the balance even further in favour of applicability at the cost of weaker guarantees, considering the average-case instead of the worst-case or limiting the guarantee to a given dataset. Unlike these relaxations, our notion is not limited to a particular dataset, but rather a particular distribution of data (e.g. emails, MRI images, etc.), which is a much weaker assumption.

For a long time, approximate DP remained unachievable in more popular deep learning scenarios. Some earlier attempts (Shokri and Shmatikov, 2015) led to prohibitively high bounds on ε (Abadi et al., 2016; Papernot et al., 2016) that were later shown to be ineffective against attacks (Hitaj et al., 2017). A major step in the direction of bringing privacy loss values down to more practical magnitudes was done by Abadi et al. (2016) with the introduction of the *moments accountant*, currently a state-of-the-art method for keeping track of the privacy loss during training. Followed by improvements in differentially private training techniques (Papernot et al., 2016, 2018), it allowed to achieve single-digit DP guarantees ($\varepsilon < 10$) for classic supervised learning benchmarks, such as MNIST, SVHN, and CIFAR.

In general, an important aspect of a privacy notion is composability, accountability, and interpretability. Apart from sharp bounds, the moments accountant is attractive because it operates within the classic notion of (ε, δ)-DP. Some of the alternative notions of DP, such as (Mironov, 2017; Bun et al., 2018), also provide tight composition theorems, along with some other advantages, but to the best of our knowledge, they are not broadly used in practice compared to traditional DP (although there are some examples (Geumlek et al., 2017)). One of the possible reasons for that is interpretability: parameters of (α, ε)-RDP or (μ, τ)-CDP are hard to interpret. While it may be difficult to quantify the actual guarantee provided by specific values of ε, δ of the traditional DP, it is still

advantageous that they have a clearer probabilistic interpretation.

Our privacy notion can be related to some of the past work on DP relaxations. In Section 5.4.6, we discuss its connection to RDP (Mironov, 2017) and the moments accountant (Abadi et al., 2016). Similarly, there is a link to concentrated DP definitions.

A number of previous relaxations considered a similar idea of limiting the scope of protected data or using the data generating distribution, either through imposing a set of data evolution scenarios (Kifer and Machanavajjhala, 2014), policies (He et al., 2014), distributions (Blum et al., 2013; Bhaskar et al., 2011), or families of distributions (Bassily et al., 2013; Bassily and Freund, 2016). Some of these definitions (e.g. (Blum et al., 2013)) may require more noise, because they are stronger than DP in the sense that datasets can differ in more than one data point. This is not the case with our definition: like DP, it considers adjacent datasets *differing in a single data point*. The major problem of such definitions, however, is that in real-world scenarios it is not feasible to define distributions or families of distributions that generate data. And even if this problem is solved by restricting the query functions to enable the usage of the central limit theorem (e.g. (Bhaskar et al., 2011; Duan, 2009)), these guarantees would only hold asymptotically and may require prohibitively large batch sizes. While Bayesian differential privacy can be seen as a special case of some of the above definitions, the crucial difference and the primary reason it is defined this way, comes with Bayesian accounting (Sections 5.4.2, 5.4.5), which *only requires a finite number of samples from these data distributions*, and hence, allows a broad range of real-world applications.

There are also approaches that use the data distribution information, in one way or another, and coincidentally share the same (Yang et al., 2015) or similar (Leung and Lui, 2012) names. Yet, similarly to the methods discussed above, their assumptions (e.g. the bound on the minimum probability of a data point) and implementation requirements (e.g. potentially constructing correlation matrices for millions of data samples) make practical applications difficult.

Perhaps the most similar to our approach is random differential privacy (Hall et al., 2011), which is also based on incorporating data randomness in the probability space. They take probability with respect to the $(n+1)$-fold product measure on the space of all datasets of $n+1$ samples from some distribution P. However, this is impractical in many machine learning scenarios, where datasets and models are large and neither data nor model parameter distributions are analytically defined. Alternatively, we consider the probability space of a single additional example, which allows us to perform a wider range of statistical estimation methods. Furthermore, Hall et al. (2011) only propose a basic composition theorem, which is not tight enough for accounting in iterative methods, and do not prove other crucial properties, such as post-processing and group privacy. We provide a more formal explanation of the relation between Bayesian DP and random DP in Section 5.4.6. In a similar fashion, Dandekar et al. (2020) proposed the notion of

	Basic Properties			Applicability			Guarantee	
	Post-proc.	*Comp.*	*Group*	*Adv. comp.*	*Runtime*	*Assumptions*	*Outcomes*	*Data*
ε-DP	✔	✔	✔	✘	✔	✔	W	W
(ε, δ)-DP	✔	✔	✔	✔	✔	✔	P	W
(ε, δ)-NP	?	✔	?	?	?	✘	–	P_D
$(\varepsilon, \delta, \Delta, \Gamma)$-CWP	✔	✘	?	?	?	?	P	W_Δ
(α, η, γ)-Random DP	?	✔	?	?	?	?	P	P_D
$(\varepsilon_\mu, \delta_\mu)$-Bayesian DP	✔	✔	✔	✔	✔	✔	P	$\mathrm{P}_{x'}$

Table 5.1 – Comparison of privacy notions. NP refers to noiseless privacy (Bhaskar et al., 2011), CWP to coupled-worlds privacy (Bassily et al., 2013). Evaluation is based on the claims found in corresponding papers, and the **?** sign suggests that we were not able to conclude a particular outcome for the property and further investigation is needed.

privacy at risk. According to the authors, it unifies probabilistic and random DP notions. They prove post-processing and composition properties, but do not solve the problem of unknown data distributions.

We evaluate our method on two popular classes of learning algorithms: deep neural networks and variational inference (VI). Privacy-preserving deep learning is being extensively studied, and is frequently used in combination with moments accountant (Abadi et al., 2016; Papernot et al., 2016, 2018), which makes it a perfect setting for comparison. Bayesian inference methods, on the other hand, have received less attention from the private learning community. There are, nonetheless, very interesting results suggesting one could obtain DP guarantees "for free" (without adding noise) in some methods like posterior sampling (Dimitrakakis et al., 2014, 2017) and stochastic gradient Monte Carlo (Wang et al., 2015). A differentially private version of variational inference, obtained by applying noise to the gradients and using moments accountant, has also been proposed (Jälkö et al., 2016). We show that Bayesian DP allows to build VI that is both accurate and differentially private by sampling from a variational distribution.

Table 5.1 highlights some aspects of different privacy definitions and illustrates how our definition compares to prior work. Apart from the basic properties and the advanced composition, we consider applicability and extent of guarantees. Under applicability, we look at computational efficiency in real-world applications ("Runtime"), and whether or not assumptions are realistic ("Assumptions"). For instance, we think that obtaining sufficiently many i.i.d. samples to apply CLT in noiseless privacy, or correlation assumptions of the earlier Bayesian DP by Yang et al. (2015) are too idealistic. Note that this category does not include assumptions about adversaries, but rather about data.

Finally, under "Guarantee," we state whether the provided guarantee is worst-case (W) or a tail bound (P). Subscripts indicate either a domain (Δ), or a distribution over which the probability is computed. One aspect of the guarantee not outlined in the table is resilience w.r.t. attackers' background knowledge. Generally, DP notions are designed to provide guarantees under any auxiliary information (except for coupled-worlds privacy, which explicitly computes probability over auxiliary inputs). Nevertheless, we caution the reader that this aspect may need additional research for data-aware notions. Hall et al. (2011) do not discuss it in their paper. Bhaskar et al. (2011) examine some scenarios where limited auxiliary information is available to attackers, but not about the protected entries. In Bayesian DP, information that alters the belief about the data distribution should be handled by the failure probability of the sample estimator (see Section 5.4.5). However, we have not explored the case when adversaries have more prior information about the privacy loss distribution (see Theorem 5) and leave it for future work.

5.3 Preliminaries

A large portion of this chapter relies heavily on a few definitions. Hence, we find it important to refresh the reader's memory in this section, even though some definitions have already appeared earlier in this thesis. We also describe the general setting of the problem and consider a more in-depth motivational example.

5.3.1 Definitions and Notation

We use D, D' to represent neighbouring (adjacent) datasets. Unless otherwise specified, these datasets differ in a single example x' (i.e. either $D' = D \cup \{x'\}$ or $D = D' \cup \{x'\}$). Private learning outcomes (model parameters, neural network weights, etc., after applying privacy mechanism) are denoted by w. Whenever ambiguous, we denote expectation over data as $\mathbb{E}_{x'}$, and over the privacy mechanism randomness as $\mathbb{E}_w$.

Recall the definitions of approximate differential privacy and privacy loss, both crucial for understanding this chapter.

Definition 5 ((ε, δ)-Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$ and for any set of outcomes $\mathcal{S} \subset \mathcal{R}$ the following holds:*

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta.$$

Definition 7 (Privacy Loss)**.** *Privacy loss $L_{\mathcal{A}}$ of a randomised algorithm $\mathcal{A} : \mathcal{D} \times \Xi \to \mathcal{R}$ for an outcome $s \in \mathcal{R}$, datasets $D, D' \in \mathcal{D}$, and auxiliary information $\xi \in \Xi$ is given by:*

$$L_{\mathcal{A}}(w; D, D', \xi) = \log \frac{\Pr\left[\mathcal{A}(D, \xi) = w\right]}{\Pr\left[\mathcal{A}(D', \xi) = w\right]}.$$

As stated earlier, in the case of continuous distributions this ratio is actually the ratio of probability density functions $p_{\mathcal{A}}(w|\xi, D)$ and $p_{\mathcal{A}}(w|\xi, D')$, such that $\Pr[\mathcal{A}(D, \xi) \in \mathcal{S}] = \int_{\mathcal{S}} p_{\mathcal{A}}(w|\xi, D)dw$. For notational simplicity, we often omit the designation $\mathcal{A}$ (i.e. we use $p(w|\xi, D)$) and other arguments, such as ξ, when it can be inferred from context. We use L to denote the privacy loss random variable, which is generated by drawing $w \sim p(w|\xi, D)$ and computing $L_{\mathcal{A}}(w; D, D', \xi)$ (see Dwork and Rothblum (2016, Section 2.1 and Definition 3.1)). Considering privacy loss random variable helps linking to well-known divergences and concentration inequalities.

Let us also define the subsampled Gaussian mechanism. This is a version of the Gaussian mechanism (Definition 8) applied to sampled subsets of inputs. It is popular in machine learning, because it maps well to mini-batch learning.

Definition 12 (Subsampled Gaussian Mechanism)**.** *Subsampled Gaussian noise mechanism for a function $f : \mathcal{D} \to \mathbb{R}^m$, is defined as*

$$\mathcal{A}_{q,\sigma}(D) \triangleq f(\{x \mid x \in D, \textit{sampled with probability } q\}) + \mathcal{N}(0, \sigma^2 \mathbb{I}^m),$$

where each element of D is sampled with probability q independently, without replacement.

We will also need the definition of Rényi divergence:

Definition 13. *Rényi divergence of order λ between distributions P and Q, denoted as $\mathcal{D}_\lambda(P\|Q)$ is defined as*

$$\begin{aligned}\mathcal{D}_\lambda(P\|Q) &= \frac{1}{\lambda - 1} \log \int p(x) \left[\frac{p(x)}{q(x)}\right]^{\lambda-1} dx \\ &= \frac{1}{\lambda - 1} \log \int q(x) \left[\frac{p(x)}{q(x)}\right]^{\lambda} dx,\end{aligned}$$

where $p(x)$ and $q(x)$ are corresponding density functions of P and Q.

Analytic expressions for Rényi divergence exist for many common distributions and can be found in (Gil et al., 2013). Van Erven and Harremos (2014) provide a good survey of Rényi divergence properties in general.

5.3.2 Setting

We assume a general iterative learning algorithm, such that each iteration t produces a non-private learning outcome $g^{(t)}$ (e.g. a gradient over a batch of data). This outcome gets transformed into a private learning outcome $w^{(t)}$ that is used as a starting point for the next iteration. The learning outcome can be made private by applying some privacy noise mechanism (e.g. a Gaussian noise mechanism) or by drawing it from a

distribution. In both cases, we say it comes from $p(w^{(t)}|w^{(t-1)}, D)$ (here we assume the Markov property of the learning process for brevity of notation, but it is not necessary in general). The process can run on subsamples of data, in which case $w^{(t)}$ comes from the distribution $p(w^{(t)}|w^{(t-1)}, B^{(t)})$, where $B^{(t)}$ is a batch of data used for parameters update at iteration t, and privacy is amplified through sampling (Balle et al., 2018). For each iteration, we would like to compute a quantity c_t (we call it a *privacy cost*) that accumulates over the learning process and allows to compute privacy loss bounds ε, δ using concentration inequalities.

The overall privacy accounting workflow does not significantly differ from prior work, but is in fact a generalisation of the well-known moments accountant (Abadi et al., 2016). Importantly, it is not tied to a specific algorithm or a class of algorithms, as long as one can map it to the above setting.

5.3.3 Motivation

Before we proceed, we find it important to have a more in-depth look at our motivation of the research on alternative definitions of privacy, as opposed to fully concentrating on new mechanisms for DP. On the one hand, there is always a combination of data and a desired statistic that would yield large privacy loss in DP paradigm, regardless of the mechanism. In other words, there can always be data outliers that are difficult to hide without a large drop in accuracy. On the other hand, we cannot realistically expect companies to sacrifice model quality in favour of privacy. As a result, we get models with impractical worst-case guarantees (as we demonstrate below) without any indication of what is the privacy guarantee for the majority of users.

Consider the following example. The datasets D, D' consist of income values for residents of a small town. There is one individual x' whose income is orders of magnitude higher than the rest, and whose residency in the town is what the attacker wishes to infer. The attacker observes the mean income w sanitised by a differentially private mechanism with $\varepsilon = \varepsilon_0$ (we consider the stronger, pure DP for simplicity). What we are interested in is the change in the posterior distribution of the attacker after they see the private model compared to prior (Mironov, 2017; Bun, 2017). If the individual is not present in the dataset, the probability of w being above a certain threshold is extremely small. On the contrary, if x' is present, this probability is higher (say it is equal to r). The attacker computes the likelihood of the observed value under each of the two assumptions, the corresponding posteriors given a flat prior, and applies a Bayes optimal classifier. The attacker then concludes that the individual is present in the dataset and is a resident.

By the above expression, r can only be e^{ε_0} times larger than the respective probability without x'. However, if the $re^{-\varepsilon_0}$ is small enough, then the probability $P(A)$ of the

attacker's guess being correct is as high as $r/(r + re^{-\varepsilon_0})$, or

$$P(A) = \frac{1}{1 + e^{-\varepsilon}}. \tag{5.1}$$

To put it in perspective, for a DP algorithm with $\varepsilon = 2$, the upper bound on the accuracy of this attack is as high as 88%. For $\varepsilon = 5$, it is 99.33%. For $\varepsilon = 10$, 99.995%. Notably, these values of ε are common in DP ML literature (Shokri and Shmatikov, 2015; Abadi et al., 2016; Papernot et al., 2018), and can be even higher in real-world deployments[1].

This guarantee does not tell us anything other than that this outlier cannot be protected while preserving utility. But what is the guarantee for other residents of the town? Intuitively, it should be much stronger. In the next section, we present a novel DP-based privacy notion. It uses the same privacy mechanism and augments the general DP guarantee with a much tighter guarantee for the expected case, and, by extension, for any percentile of the user/data population.

5.4 Bayesian Differential Privacy

In this section, we define *Bayesian differential privacy (BDP)*. We then state its properties analogous to the classical DP, derive a practical privacy loss accounting method, and discuss other aspects, such as relation to the moments accountant.

5.4.1 Definition

We start with the definition of *strong* Bayesian differential privacy (Definition 14) and (*weak*) Bayesian differential privacy (Definition 15). The first provides a better intuition, connection to concentration inequalities, and is being used for privacy accounting. Unfortunately, it may not be closed under post-processing, and therefore, the actual guarantee provided by BDP is stated in Definition 15 and mimics the (ε, δ)-differential privacy (Dwork et al., 2014). The reason Definition 14 may pose a problem with post-processing is that it does not consider sets of outcomes, and a routine that integrates groups of values into one value could therefore invalidate the guarantee by increasing the probability of exceeding the ratio beyond epsilon. On the other hand, it can still be used for accounting with adaptive composition, because in this context, every next step is conditioned on a single outcome of the previous step. This separation mirrors the moments accountant approach of bounding tails of the privacy loss random variable and converting it to the (ε, δ)-DP guarantee (Abadi et al., 2016), but does so in a more explicit manner.

[1] https://www.macobserver.com/analysis/google-apple-differential-privacy/

Definition 14 (Strong Bayesian Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$, with domain $\mathcal{D}$ and range $\mathcal{R}$, satisfies $(\varepsilon_\mu, \delta_\mu)$-strong Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$,* differing in a single data point $x' \sim \mu(x)$*, and auxiliary inputs ξ, the following holds:*

$$\Pr[L_{\mathcal{A}}(w; D, D', \xi) \geq \varepsilon_\mu] \leq \delta_\mu, \tag{5.2}$$

where probability is taken over the randomness of the outcome $w \sim \mathcal{A}(D, D', \xi)$ and the additional example $x' \sim \mu(x)$.

In the above definition, the probability is more formally defined as

$$\Pr(L \geq \varepsilon_\mu) = \int \mu(x') \int p_{\mathcal{A}}(w|\xi, D) \mathbb{1}\{L(w; D, D', \xi) \geq \varepsilon_\mu\} \, dw \, dx'. \tag{5.3}$$

We use the subscript μ to underline the main difference between the classic DP and Bayesian DP: in the classical definition the probability is taken only over the randomness of the outcome (w), while the BDP definition contains two random variables (w and x'). Therefore, the privacy parameters ε and δ depend on the data distribution $\mu(x)$.

Addition of another random variable yields the change in the meaning of δ_μ compared to the δ of DP. In Bayesian differential privacy, it also accounts for the privacy mechanism failures in the tails of data distributions in addition to the tails of outcome distributions.

Remark 1. Strong BDP definition is analogous to probabilistic DP (see Definition 6). However, we omit the absolute value of the privacy loss due to the symmetry with respect to D and D'. Since the condition has to hold for any pair of datasets D, D', and $L(w; D, D', \xi) = -L(w; D', D, \xi)$, we can use L instead of $|L|$ and still proof all the necessary properties of the definition. The same omission is used in the moments accountant (see Abadi et al. (2016, Appendix A)).

Definition 15 (Bayesian Differential Privacy)**.** *A randomised function (algorithm) $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$, differing in a single data point $x' \sim \mu(x)$, and for any set of outcomes $\mathcal{S}$ the following holds:*

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta_\mu. \tag{5.4}$$

Proposition 2. *$(\varepsilon_\mu, \delta_\mu)$-strong Bayesian differential privacy implies $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy.*

Let us also formulate some basic properties of Bayesian DP that mirror the properties of the classic DP. The proofs of these properties, as well as the above proposition, can be found in Appendix A.1.

First, the post-processing property, which only holds for the "weak" sense of BDP.

Proposition 3 (Post-processing)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for any arbitrary randomised data-independent mapping $f : \mathcal{R} \to \mathcal{R}'$, $f(\mathcal{A}(D))$ is $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private.*

At the same time, the following two propositions can be proven for both "strong" and "weak" variants of Bayesian DP.

Proposition 4 (Basic composition)**.** *Let $\mathcal{A}_i : \mathcal{D} \to \mathcal{R}_i$, $\forall i = 1..k$, be a sequence of $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithms. Then their combination, defined as $\mathcal{A}_{1:k} : \mathcal{D} \to \mathcal{R}_1 \times \ldots \times \mathcal{R}_k$, is $(k\varepsilon_\mu, k\delta_\mu)$-Bayesian differentially private.*

Proposition 5 (Group privacy)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for all pairs of datasets $D, D' \in \mathcal{D}$, differing in k data points $x_1, \ldots, x_k$ s.t. $x_i \sim \mu(x)$ for $i = 1..k$, $\mathcal{A}(D)$ is $(k\varepsilon_\mu, ke^{k\varepsilon_\mu}\delta_\mu)$-Bayesian differentially private.*

In the remainder of the chapter, whenever dependency on $\mu(x)$ can be inferred from the context, we omit the subscript μ.

Remark 2. While Definitions 14 and 15 do not specify the distribution of any point in the dataset other than the additional example x', it is natural and convenient to assume that all examples in the dataset are drawn from the same distribution $\mu(x)$. This holds in many real-world applications, including all applications evaluated in this thesis. Furthermore, it allows using samples from the dataset for estimating privacy loss, instead of requiring knowing the true distribution. In this case, it is more appropriate to say that $x' \sim \mu(x|D)$, and use the notation $\varepsilon_{\mu|D}$ and $\delta_{\mu|D}$, although we omit it for brevity.

Remark 3. We also assume that all data points are exchangeable (Aldous, 1985), i.e. any permutation of data points has the same joint probability. It enables tighter accounting for iterative applications of the privacy mechanism (see Sections 5.4.2 and 5.4.5), and is naturally satisfied in the considered scenarios.

5.4.2 Privacy Accounting

In the context of learning, it is important to be able to keep track of the privacy loss over iterative applications of the privacy mechanism. And since the bounds provided by the basic composition theorem are loose, we derive the *advanced composition theorem* and develop a general accounting method for Bayesian differential privacy, the *Bayesian accountant (BA)*, that provides a tight bound on privacy loss and is straightforward to implement. We draw inspiration from the moments accountant. In Section 5.4.6, we show that it is actually a generalisation of the moments accountant.

Observe that Eq. 5.2 is a typical concentration bound inequality, which are well studied in probability theory. One of the most common examples of such bounds is Markov's inequality. In its extended form, it states the following:

$$\Pr\left[|L| \geq \varepsilon \mid D, \xi\right] \leq \frac{\mathbb{E}\left[\varphi(|L|) \mid D, \xi\right]}{\varphi(\varepsilon)}, \tag{5.5}$$

where $\varphi(\cdot)$ is a monotonically increasing non-negative function. It is immediately evident that it provides a relation between ε and δ (i.e. $\delta = \mathbb{E}[\varphi(|L|)]/\varphi(\varepsilon)$), and in order to determine them we need to choose φ and compute the expectation $\mathbb{E}[\varphi(|L(w; D, D', \xi)|)]$.

We use the Chernoff bound that can be obtained by choosing $\varphi(L) = e^{\lambda L}$. It is widely known because of its tightness, and although not explicitly stated, it is also used by Abadi et al. (2016). The inequality in this case transforms to

$$\Pr[L \geq \varepsilon \,|\, D, \xi] \leq \frac{\mathbb{E}[e^{\lambda L} \,|\, D, \xi]}{e^{\lambda \varepsilon}}. \tag{5.6}$$

This inequality requires the knowledge of the moment generating function of L or some bound on it. The choice of the parameter λ can be arbitrary, because the bound holds for any value of it, but it determines how tight the bound is. By simple manipulations we obtain

$$\begin{aligned} \mathbb{E}\left[e^{\lambda L} \,\middle|\, D, \xi\right] &= \mathbb{E}\left[e^{\lambda \log \frac{p(w|\xi,D)}{p(w|\xi,D')}} \,\middle|\, D, \xi\right] \\ &= \mathbb{E}\left[\left(\frac{p(w|\xi,D)}{p(w|\xi,D')}\right)^{\lambda} \,\middle|\, D, \xi\right]. \end{aligned} \tag{5.7}$$

If the expectation is taken only over the outcome randomness, this expression is the function of Rényi divergence (see Definiton 13) between $p(w|\xi, D)$ and $p(w|\xi, D')$, and following this path yields re-derivation of Rényi differential privacy (Mironov, 2017). However, by also taking the expectation over additional examples $x' \sim \mu(x)$, we can further tighten this bound.

By the law of total expectation,

$$\mathbb{E}\left[\left(\frac{p(w|\xi,D)}{p(w|\xi,D')}\right)^{\lambda}\middle| D,\xi\right] = \mathbb{E}_{x'\sim\mu(x)}\left[\mathbb{E}_{w\sim p(w|\xi,D)}\left[\left(\frac{p(w|\xi,D)}{p(w|\xi,D')}\right)^{\lambda}\middle| x'\right]\right], \tag{5.8}$$

where the inner expectation is again the function of Rényi divergence, and the outer expectation is over the distribution of a distinct example $\mu(x)$.

Combining Eq. 5.7 and 5.8 and plugging it in Eq. 5.6, we get

$$\Pr[L \geq \varepsilon] \leq \mathbb{E}_{x'}\left[e^{\lambda\mathcal{D}_{\lambda+1}[p(w|\xi,D)\|p(w|\xi,D')]-\lambda\varepsilon}\right]. \tag{5.9}$$

This expression determines how to compute ε for a fixed δ (or vice versa) for one invocation of the privacy mechanism. However, to accommodate the iterative nature of learning, we need to deal with the composition of multiple applications of the mechanism. We already determined that our privacy notion is naively composable, but in order to achieve better bounds we need a tighter composition theorem.

Let us first define the notion of *privacy cost*.

Definition 16 (Privacy Cost)**.** *Privacy cost $c(\lambda,\xi,T)$ for order λ, auxiliary input ξ, datasets D, D', and exponent r is defined as*

$$c_r(\lambda,\xi,D,D') = \log\mathbb{E}_{x'}\left[e^{r\lambda\mathcal{D}_{\lambda+1}(p_D\|p_{D'})}\right]^{\frac{1}{r}} \tag{5.10}$$

where $p_D = p(w|\xi,D)$, $p_{D'} = p(w|\xi,D')$.

We can now formulate the advanced composition theorem.

Theorem 1 (Advanced Composition)**.** *Let $\mathcal{A}^{(1:T)} = (\mathcal{A}^{(1)},\ldots,\mathcal{A}^{(T)})$ be a sequence of privacy mechanisms, $\xi^{(1:T)} = (\xi^{(1)},\ldots,\xi^{(T)})$ a sequence of auxiliary inputs to mechanisms. Then the total privacy cost $c(\lambda,\xi^{(1:T)},D,D')$ of $\mathcal{A}^{(1:T)}$ satisfies*

$$c(\lambda,\xi^{(1:T)},D,D') \leq \sum_{t=1}^{T} c_T(\lambda,\xi^{(t)},D,D').$$

Proof. We follow the main steps of the moments accountant proof (Abadi et al., 2016).

Let $w^{(1)},\ldots,w^{(T)}$ denote outputs of private mechanisms. The total privacy loss can be

written as

$$L^{(1:T)} = \log \frac{\Pr[\mathcal{A}(D, \xi^{(1)}) = w^{(1)}, \ldots, \mathcal{A}(D, \xi^{(T)}) = w^{(T)} \mid \xi^{(1:T)}, D]}{\Pr[\mathcal{A}(D', \xi^{(1)}) = w^{(1)}, \ldots, \mathcal{A}(D', \xi^{(T)}) = w^{(T)} \mid \xi^{(1:T)}, D']} \tag{5.11}$$

$$= \log \prod_{t=1}^{T} \frac{p(w^{(t)} \mid \xi^{(t)}, D)}{p(w^{(t)} \mid \xi^{(t)}, D')} \tag{5.12}$$

$$= \sum_{t=1}^{T} L^{(t)} \tag{5.13}$$

Unlike the composition proof of the moments accountant in Abadi et al. (2016), we cannot simply swap the product and the expectation in our proof, because the additional example x' remains the same in all applications of the privacy mechanism and probability distributions will not be independent. However, we can use generalised Hölder's inequality:

$$\left\| \prod_{t=1}^{T} f_t \right\|_r \leq \prod_{t=1}^{T} \|f_t\|_{p_t}, \tag{5.14}$$

where p_t are such that $\sum_{t=1}^{T} \frac{1}{p_t} = \frac{1}{r}$, and $\|f\|_r = \left(\int_S |f|^r dx\right)^{1/r}$.

Choosing $r = 1$ and $p_t = T$:

$$\mathbb{E}\left[e^{\lambda L^{(1:T)}} \;\middle|\; D, \xi \right] = \mathbb{E}\left[\prod_{t=1}^{T} e^{\lambda \log \frac{p(w^{(t)} \mid \xi^{(t)}, D)}{p(w^{(t)} \mid \xi^{(t)}, D')}} \;\middle|\; D, \xi \right] \tag{5.15}$$

$$= \mathbb{E}_{x'}\left[\mathbb{E}_{w}\left[\prod_{t=1}^{T} e^{\lambda \log \frac{p(w^{(t)} \mid \xi^{(t)}, D)}{p(w^{(t)} \mid \xi^{(t)}, D')}} \;\middle|\; x' \right] \right] \tag{5.16}$$

$$= \mathbb{E}_{x'}\left[\prod_{t=1}^{T} \mathbb{E}_{w}\left[e^{\lambda \log \frac{p(w^{(t)} \mid \xi^{(t)}, D)}{p(w^{(t)} \mid \xi^{(t)}, D')}} \;\middle|\; x' \right] \right] \tag{5.17}$$

$$= \mathbb{E}_{x'}\left[\prod_{t=1}^{T} e^{\lambda \mathcal{D}_{\lambda+1}(p_D \| p_{D'})} \right] \tag{5.18}$$

$$\leq \prod_{t=1}^{T} \mathbb{E}_{x'}\left[e^{T\lambda \mathcal{D}_{\lambda+1}(p_D \| p_{D'})} \right]^{\frac{1}{T}}, \tag{5.19}$$

where (5.17) is because $w^{(t)}$ is independent of $w^{(1:t-1)}$ given $\xi^{(t)}$.

Taking logarithm on both sides yields the claim. □

The upper bound in the theorem may appear loose, but we found that the inequality tends to be tight in practice. In Section 5.5.1, we show that the privacy curve over iterations obtained via the above theorem is close to the one obtained by naïvely swapping the product and the expectation.

We can now relate ε and δ parameters of BDP through the privacy cost.

Theorem 2. *Let the algorithm $\mathcal{A}$ run for T steps and produce a sequence of private outcomes $w^{(1)} \dots w^{(T)}$ using a known probability distribution $p(w^{(t)}|\xi^{(t)}, D)$. Then, for a fixed ε*

$$\log \delta_\mu \leq \max_{\xi} \sum_{t=1}^{T} c_T(\lambda, \xi^{(t)}, D, D') - \lambda\varepsilon.$$

Corollary 1. *Under the conditions above, for a fixed δ:*

$$\varepsilon_\mu \leq \frac{1}{\lambda} \max_{\xi} \sum_{t=1}^{T} c_T(\lambda, \xi^{(t)}, D, D') - \frac{1}{\lambda} \log \delta.$$

Theorems 1, 2 and Corollary 1 immediately provide us with an efficient privacy accounting algorithm. During training, we compute the privacy cost $c_T(\lambda, \xi^{(t)}, D, D')$ for each iteration t, accumulate it, and then use to compute ε, δ pair. This process is ideologically close to that of the moment accountant, but accumulates a different quantity (note the change from the privacy loss random variable to Rényi divergence). We further explore this connection in Section 5.4.6.

The link to Rényi divergence is a great advantage for applicability of this framework: as long as the outcome distribution $p(w|\xi, D)$ has a known expression for Rényi divergence, it can be used within a privacy mechanism, and our accountant can track its privacy loss. Analytic expressions for Rényi divergence of many common distributions can be found in (Gil et al., 2013; Van Erven and Harremos, 2014).

Remark 4 (Optimal choice of λ)**.** Chernoff inequality holds for any parameter $\lambda > 0$, and thus, to get the optimal estimates of ε, δ one should minimise the right-hand side in Theorem 2 w.r.t. λ. While Abadi et al. (2016) suggest computing moments for $\lambda \leq 32$, we observe that since the moment generating function is log-convex it is possible to find an optimal value of λ that minimises the total bound. For some distributions, e.g. Gaussian without subsampling, it can be found analytically by computing the derivative and setting it to 0. Unfortunately, for a more interesting case of subsampled privacy mechanisms, this is less straightforward. Section 5.5.1 provides some more details on how ε depends on the choice of λ.

5.4.3 Subsampled Gaussian Mechanism

In this section, we consider the subsampled Gaussian noise mechanism (Definition 12), the primary mechanism used in privacy-preserving machine learning. It differs from

the original mechanism by the fact that it is applied on batches of data sampled with some probability rather than the whole dataset. In this case, privacy is amplified by sampling (Abadi et al., 2016).

The outcome distribution $p(w^{(t)} \mid w^{(t-1)}, D')$ in this case is equivalent to the mixture of two Gaussians $p(w^{(t)} \mid w^{(t-1)}, D') = (1-q)\mathcal{N}(g_t, \sigma^2) + q\mathcal{N}(g'_t, \sigma^2)$, where g_t and g'_t are non-private outcomes at the iteration t computed on a batch without x' and with x' correspondingly, σ is the noise parameter of the mechanism, and q is the data sampling probability. Plugging the outcome distribution into the formula for Rényi divergence, we get the following result for the privacy cost.

Theorem 3 (Privacy Cost of Subsampled Gaussian Mechanism)**.** *Given the Gaussian noise mechanism with the noise parameter σ and subsampling probability q, the privacy cost for $\lambda \in \mathbb{N}$ at iteration t can be expressed as*

$$c_T(\lambda, \xi^{(t)}, D, D') = \max\left\{c_T^L(\lambda, \xi^{(t)}, D, D'),\ c_T^R(\lambda, \xi^{(t)}, D, D')\right\},$$

where

$$c_T^L(\lambda, \xi^{(t)}, D, D') = \frac{1}{T}\log \mathbb{E}_{x'}\left[\mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{\frac{k^2-k}{2\sigma^2}\|g_t - g'_t\|^2}\right]^T\right],$$

$$c_T^R(\lambda, \xi^{(t)}, D, D') = \frac{1}{T}\log \mathbb{E}_{x'}\left[\mathbb{E}_{k\sim\mathcal{B}(\lambda,q)}\left[e^{\frac{k^2+k}{2\sigma^2}\|g_t - g'_t\|^2}\right]^T\right],$$

and $\mathcal{B}(\lambda, q)$ is the binomial distribution with λ experiments and the probability of success q.

Proof. Without loss of generality, assume $D' = D \cup \{x'\}$. For brevity, let $d_t = \|g_t - g'_t\|$.

Let us first consider $\mathcal{D}_{\lambda+1}(p(w|D')\|p(w|D))$:

$$\begin{aligned}
&\mathbb{E}_w\left[\left(\frac{p(w|D')}{p(w|D)}\right)^{\lambda+1}\right] \\
&\quad= \mathbb{E}_w\left[\left(\frac{(1-q)\mathcal{N}(0,\sigma^2) + q\mathcal{N}(d_t,\sigma^2)}{\mathcal{N}(0,\sigma^2)}\right)^{\lambda+1}\right] &(5.20)\\
&\quad= \mathbb{E}_w\left[\left((1-q) + q\frac{\mathcal{N}(d_t,\sigma^2)}{\mathcal{N}(0,\sigma^2)}\right)^{\lambda+1}\right] &(5.21)\\
&\quad= \mathbb{E}_w\left[\left((1-q) + qe^{\frac{(w-d_t)^2-w^2}{2\sigma^2}}\right)^{\lambda+1}\right] &(5.22)
\end{aligned}$$

$$= \mathbb{E}_w \left[\left((1-q) + q e^{\frac{2dw - d_t^2}{2\sigma^2}} \right)^{\lambda+1} \right] \tag{5.23}$$

$$= \mathbb{E}_w \left[\sum_{k=0}^{\lambda+1} \binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} e^{\frac{2d_t kw - kd_t^2}{2\sigma^2}} \right] \tag{5.24}$$

$$= \sum_{k=0}^{\lambda+1} \binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} \mathbb{E}_w \left[e^{\frac{2d_t kw - kd_t^2}{2\sigma^2}} \right] \tag{5.25}$$

$$= \sum_{k=0}^{\lambda+1} \binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} e^{\frac{k^2-k}{2\sigma^2} d_t^2} \tag{5.26}$$

$$= \mathbb{E}_{k \sim B(\lambda+1,q)} \left[e^{\frac{k^2-k}{2\sigma^2} \|g_t - g_t'\|^2} \right], \tag{5.27}$$

Here, in (5.34) we used the binomial expansion, in (5.35) the fact that the factors in front of the exponent do not depend on w, and in (5.26) the property $\mathbb{E}_w \left[\exp(2aw/(2\sigma^2)) \right] = \exp(a^2/(2\sigma^2))$ for $w \sim \mathcal{N}(0, \sigma^2)$. Plugging the above in Eq. 5.10, we get the expression for $c_t^L(\lambda)$.

Computing $\mathcal{D}_{\lambda+1}(p(w|D) \| p(w|D'))$ is a little more challenging. Let us first change to $\mathcal{D}_\lambda(p(w|D) \| p(w|D'))$, so that the expectation is taken over $\mathcal{N}(0, \sigma^2)$ (see Definition 13). Then, we can bound it observing that $f(x) = \frac{1}{x}$ is convex for $x > 0$ and using the definition of convexity, and apply the same steps as above:

$$\mathbb{E}_w \left[\left(\frac{p(w|D)}{p(w|D')} \right)^\lambda \right]$$

$$= \mathbb{E}_w \left[\left(\frac{\mathcal{N}(0, \sigma^2)}{(1-q)\mathcal{N}(0, \sigma^2) + q\mathcal{N}(d_t, \sigma^2)} \right)^\lambda \right] \tag{5.28}$$

$$\leq \mathbb{E}_w \left[\left((1-q) + q e^{\frac{d_t^2 - 2dw}{2\sigma^2}} \right)^\lambda \right] \tag{5.29}$$

$$= \mathbb{E}_{k \sim B(\lambda,q)} \left[e^{\frac{k^2+k}{2\sigma^2} \|g_t - g_t'\|^2} \right] \tag{5.30}$$

We have not encountered any instance of $\mathcal{D}_{\lambda+1}(p(w|D') \| p(w|D)) < \mathcal{D}_{\lambda+1}(p(w|D) \| p(w|D'))$ in practice, even computing the latter using numerical integration instead of the above upper bound. Although we have not investigated it in detail, it may be possible to prove that the former is always greater than the latter (Mironov et al., 2019). □

5.4.4 General Subsampled Mechanism

Let us now consider a general subsampled mechanism. Assume there is an abstract outcome distribution $p(w|\cdot)$. In the previous section, we had $p(w \mid D') = (1-q)\mathcal{N}(g, \sigma^2) +$

$q\mathcal{N}(g', \sigma^2)$, where g and g' are non-private outcomes obtained correspondingly from the batches B, without the additional example x', and B', with the additional example x'. In this section, let $p(w \mid D')$ be defined in a more general way:

$$p(w \mid D') = (1-q)p(w \mid B) + qp(w \mid B'). \tag{5.31}$$

We note that there is some prior work in this direction for general subsampled Rényi DP mechanisms (Wang et al., 2019). However, we believe that our formulation and the proof are more intuitive because of the direct parallels with the special case of subsampled Gaussian mechanisms and a more straightforward and shorter proving technique.

Theorem 4 (Privacy Cost of General Subsampled Mechanism)**.** *Given a subsampled privacy mechanism with the outcome distribution $p(w|\cdot)$ and sampling probability q, the privacy cost for $\lambda \in \mathbb{N}$ at iteration t can be expressed as*

$$c_T(\lambda, \xi^{(t)}, D, D') = \max\left\{c_T^L(\lambda, \xi^{(t)}, D, D'),\ c_T^R(\lambda, \xi^{(t)}, D, D')\right\},$$

such that

$$c_T^L(\lambda, \xi^{(t)}, D, D') = \frac{1}{T}\log \mathbb{E}_{x'}\left[\mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{k\mathcal{D}_{k+1}(p(w|B')|p(w|B))}\right]^T\right],$$
$$c_T^R(\lambda, \xi^{(t)}, D, D') = \frac{1}{T}\log \mathbb{E}_{x'}\left[\mathbb{E}_{k\sim\mathcal{B}(\lambda,q)}\left[e^{k\mathcal{D}_{k+1}(p(w|B)|p(w|B'))}\right]^T\right],$$

where $\mathcal{B}(\lambda, q)$ is the binomial distribution with λ experiments and the probability of success q, and B', B denote sampled subsets of data with and without the additional example x' accordingly.

Proof. Similarly to the Gaussian mechanism,

$$\mathbb{E}_w\left[\left(\frac{p(w|D')}{p(w|D)}\right)^{\lambda+1}\right] = \mathbb{E}_w\left[\left(\frac{(1-q)p(w|B) + qp(w|B')}{p(w|B)}\right)^{\lambda+1}\right] \tag{5.32}$$
$$= \mathbb{E}_w\left[\left((1-q) + q\frac{p(w|B')}{p(w|B)}\right)^{\lambda+1}\right] \tag{5.33}$$
$$= \mathbb{E}_w\left[\sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k}q^k(1-q)^{\lambda+1-k}\left(\frac{p(w|B')}{p(w|B)}\right)^k\right] \tag{5.34}$$
$$= \sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k}q^k(1-q)^{\lambda+1-k}\mathbb{E}_w\left[\left(\frac{p(w|B')}{p(w|B)}\right)^k\right] \tag{5.35}$$
$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[\mathbb{E}_w\left[\left(\frac{p(w|B')}{p(w|B)}\right)^k\right]\right] \tag{5.36}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{k\mathcal{D}_{k+1}(p(w|B')|p(w|B))}\right], \tag{5.37}$$

where B' and B denote batches of data in which the additional example x' was and was not sampled correspondingly. In (5.34) we use the binomial expansion, in (5.35) the fact that the factors in front of the exponent do not depend on w.

For the inverse direction, $\mathcal{D}_{\lambda+1}(p(w|D)\|p(w|D'))$, change to $\mathcal{D}_{\lambda}(p(w|D)\|p(w|D'))$ where the expectation is taken over $p(w|D)$. Then, since $f(x) = \frac{1}{x}$ is a convex function for $x > 0$, we can use the definition of convexity (more specifically, that $\frac{1}{(1-q)x+qy} \leq (1-q)\frac{1}{x} + q\frac{1}{y}$), and then apply the same steps as above:

$$\mathbb{E}_w\left[\left(\frac{p(w|D')}{p(w|D)}\right)^{\lambda}\right] = \mathbb{E}_w\left[\left(\frac{p(w|B)}{(1-q)p(w|B)+qp(w|B')}\right)^{\lambda}\right] \tag{5.38}$$

$$\leq \mathbb{E}_w\left[\left((1-q)+q\frac{p(w|B)}{p(w|B')}\right)^{\lambda}\right] \tag{5.39}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda,q)}\left[\mathbb{E}_w\left[\left(\frac{p(w|B)}{p(w|B')}\right)^{k}\right]\right] \tag{5.40}$$

$$= \mathbb{E}_{k\sim\mathcal{B}(\lambda+1,q)}\left[e^{k\mathcal{D}_{k+1}(p(w|B)|p(w|B'))}\right]. \tag{5.41}$$

Combining the above with Definition 16, we can compute $c_T(\lambda, \xi^{(t)}, D, D')$. □

Remark 5. For non-integer λ, one can also perform numerically stable computation using the Gauss error function, described for the moments accountant and RDP by Mironov et al. (2019).

5.4.5 Privacy Cost Estimator

Computing $c_T(\lambda, \xi^{(t)}, D, D')$ precisely requires access to the data distribution $\mu(x)$, which is unrealistic. Therefore, we need an estimator for $\mathbb{E}_{x'}[e^{\lambda\mathcal{D}_{\lambda+1}(p_t\|q_t)}]$.

Typically, having access to the distribution samples, one would use the law of large numbers and approximate the expectation with the sample mean. This estimator is unbiased and converges with the growing number of samples. However, these are not the properties we are looking for. The most important property of the estimator in our context is that it *does not underestimate* $\mathbb{E}_{x'}[e^{\lambda\mathcal{D}_{\lambda+1}(p_t\|q_t)}]$, because the bound (Eq. 5.6) would not hold for this estimate otherwise.

To solve this, we employ the Bayesian view of the parameter estimation problem (Oliphant, 2006) and design an estimator with the single property: given a fixed probability γ, it returns the value that overestimates the true expectation with probability $1-\gamma$. We then incorporate the estimator uncertainty γ in δ.

Binary Case

Let us demonstrate the process of constructing the expectation estimator with the aforementioned property on a simple binary example. This technique is based on (Oliphant, 2006) and it translates directly to other classes of distributions with minor adjustments. We also address a natural concern about taking into account the data not present in the dataset by providing a specific example.

Let the data $\{x_1, x_2, \ldots, x_N\}$, such that $x_i \in \{0, 1\}$, have a common mean and a common variance. As this information is insufficient to solve our problem, let us also assume that the data comes from *the maximum entropy distribution*. This assumption adds the minimum amount of information to the problem and makes our estimate pessimistic.

For the binary data with the common mean ρ, the maximum entropy distribution is the Bernoulli distribution:

$$f(x_i|\rho) = \rho^{x_i}(1-\rho)^{1-x_i}, \tag{5.42}$$

where ρ is also the probability of success ($x_i = 1$). Then, for the entire dataset:

$$f(x_1, \ldots, x_N|\rho) = \rho^{N_1}(1-\rho)^{N_0}, \tag{5.43}$$

where N_1 is the number of ones, and N_0 is the number of zeros in the dataset.

We impose the flat prior on ρ, assuming all values in $[0, 1]$ are equally likely, and use Bayes' theorem to determine the distribution of ρ given the data:

$$f(\rho|x_1, \ldots, x_N) = \frac{\Gamma(N_0 + N_1 + 2)}{\Gamma(N_0 + 1)\Gamma(N_1 + 1)}\rho^{N_1}(1-\rho)^{N_0}, \tag{5.44}$$

where the normalisation constant in front is obtained by setting the integral over ρ equal to 1.

Now, we can use the above distribution of ρ to design an estimator $\hat{\rho}$, such that it overestimates ρ with high probability, i.e. $\Pr[\rho \leq \hat{\rho}] \geq 1 - \gamma$. Namely, $\hat{\rho} = F^{-1}(1-\gamma)$, where F^{-1} is the inverse of the CDF:

$$F^{-1}(1-\gamma) = \inf\{z \in \mathbb{R} : \int_{-\infty}^{z} f(t|x_1, \ldots, x_N)dt \geq 1 - \gamma\}.$$

We refer to γ as the *estimator failure probability*, and to $1-\gamma$ as the *estimator confidence*.

To demonstrate the resilience of this estimator to unseen data, consider the following simple example. Let the true expectation be 0.01, and let the data consist of 100 zeros, and no ones. A typical "frequentist" mean estimator would confidently output 0. However, our estimator would never output 0, unless the confidence is set to 0. When the confidence

is set to 1 ($\gamma = 0$), the output is 1, which is the most pessimistic estimate. Finally, the output $\hat{\rho} = \rho = 0.01$ will be assigned the failure probability $\gamma = 0.99^{101} \approx 0.36$, which is the probability of not drawing a single 1 in 101 draws.

In a real-world system, the confidence would be set to a much higher level (in our experiments, we use $\gamma = 10^{-15}$), and the probability of 1 would be significantly overestimated. Thus, unseen data do not present a problem for this estimator, because it exaggerates the probability of data that increase the estimated expectation.

Continuous Case

For applications in this thesis, we are primarily interested in continuous distributions.

Definition 17. *Let us define the following m-sample estimator of $c_T(\lambda, \xi, D, D')$ for continuous privacy loss distributions with existing mean and variance:*

$$\hat{c}_T(\lambda, \xi, D, D'; \gamma, m) = \frac{1}{T} \log \left[M + \frac{F^{-1}(1-\gamma, m-1)}{\sqrt{m-1}} S \right], \tag{5.45}$$

where

$$\begin{aligned}
M &= \frac{1}{m} \sum_{i=1}^{m} e^{T\lambda \hat{\mathcal{D}}_{\lambda+1}}, \\
S &= \sqrt{\frac{1}{m} \sum_{i=1}^{m} e^{2T\lambda \hat{\mathcal{D}}_{\lambda+1}} - M^2}, \\
\hat{\mathcal{D}}^{(t)}_{\lambda+1} &= \max \left\{ \mathcal{D}_{\lambda+1}(\hat{p}_D \| \hat{p}_{D'}),\ \mathcal{D}_{\lambda+1}(\hat{p}_{D'} \| \hat{p}_D) \right\}, \\
\hat{p}_D &= p(w \mid \xi, D), \\
\hat{p}_{D'} &= p(w \mid \xi, D \setminus \{x_i\}),
\end{aligned}$$

and $F^{-1}(1-\gamma, m-1)$ is the inverse of the Student's t-distribution CDF at $1-\gamma$ with $m-1$ degrees of freedom.

Since in many cases learning is performed on mini-batches, we can compute Rényi divergence in a similar way, on batches $B^{(t)}$ instead of the entire dataset D.

Theorem 5. *Bayesian estimator $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ of the privacy cost, under maximum entropy assumptions (e.g. uninformative priors), satisfies*

$$\Pr \left[\hat{c}_T(\lambda, \xi, D, D'; \gamma, m) < c_T(\lambda, \xi, D, D') \right] \leq \gamma.$$

Proof. The proof is similar to the binary example above, with minor adjustments. First of all, we can drop the logarithm from our consideration because of its monotonicity. Now, assuming that samples $e^{\lambda \hat{D}^{(t)}_{\lambda+1}}$ have a common mean and a common variance, and applying the maximum entropy principle in combination with an uninformative (flat) prior, one can show that the quantity $\frac{1}{S(t)}\left(M(t) - \mathbb{E}\left[e^{\lambda \hat{D}^{(t)}_{\lambda+1}}\right]\right)\sqrt{m-1}$ follows the Student's t-distribution with $m-1$ degrees of freedom. See (Oliphant, 2006) for a more in depth discussion of the Bayesian perspective on mean and variance estimation and derivation of the corresponding posterior distributions. Finally, we use the inverse of the Student's t CDF to find the value that this random variable would only exceed with probability γ. □

Remark 6. By adapting the maximum entropy probability distribution an equivalent estimator can be derived for other classes of distributions (e.g. discrete).

To avoid introducing new parameters in the privacy definition, we can incorporate the probability γ of underestimating the true expectation in δ. We can re-write:

$$\begin{aligned}
&\Pr[L_{\mathcal{A}}(w^{(t)}; D, D', \xi) \geq \varepsilon] \\
&\quad = \Pr\left[L_{\mathcal{A}}(w^{(t)}; D, D', \xi) \geq \varepsilon,\ \hat{c}_T(\lambda, \xi, D, D'; \gamma, m) \geq c_T(\lambda, \xi^{(t)}, D, D')\right] \\
&\qquad + \Pr\left[L_{\mathcal{A}}(w^{(t)}; D, D', \xi) \geq \varepsilon,\ \hat{c}_T(\lambda, \xi, D, D'; \gamma, m) < c_T(\lambda, \xi^{(t)}, D, D')\right].
\end{aligned}$$

When $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m) \geq c_T(\lambda, \xi^{(t)}, D, D')$, using the Chernoff inequality, the first summand is bounded by $\beta = \exp(\sum_{t=1}^{T} \hat{c}_T(\lambda, \xi, D, D'; \gamma, m) - \lambda\varepsilon)$. At the same time,

$$\begin{aligned}
&\Pr[L_{\mathcal{A}}(w^{(t)}; D, D', \xi) \geq \varepsilon,\ \hat{c}_T(\lambda, \xi, D, D'; \gamma, m) < c_T(\lambda, \xi^{(t)}, D, D')] \\
&\quad \leq \Pr[\hat{c}_T(\lambda, \xi, D, D'; \gamma, m) < c_T(\lambda, \xi^{(t)}, D, D')] \\
&\quad \leq \gamma.
\end{aligned}$$

Therefore, the true δ_μ is bounded from above by $\delta = \beta + \gamma$, and despite the incomplete data, we can claim that the mechanism is (ε, δ)-Bayesian differentially private.

Remark 7. This step further changes the interpretation of δ in Bayesian differential privacy compared to the classic DP. Apart from the probability of the privacy loss exceeding ε, e.g. in the tails of its distribution, it also incorporates our uncertainty about the true data distribution (in other words, the probability of underestimating the true expectation because of not observing enough data samples). It can be intuitively understood as accounting for unobserved (but feasible) data in δ, rather than in ε.

Remark 8. We chose Bayesian approach because it provides tighter bounds. However, it introduces a subjective element of prior. An alternative technique, albeit with looser bounds, would be to use an empirical variation of concentration inequalities, e.g. empirical Bernstein bounds (Maurer and Pontil, 2009).

Algorithm 2 DP-SGD with Bayesian differential privacy accounting.

Input:
Dataset $D = \{x_1, \cdots, x_N\}$, loss function $\mathcal{L}(w, D) = \frac{1}{N}\sum_i \mathcal{L}(w, x_i)$.
Parameters: learning rate η_t, noise scale σ, batch size B, gradient norm bound C.

Initialise w_0 randomly, Bayesian accountant with λ, q, σ, C, T
for $t \in [1..T]$ **do**
$\mathcal{B}_t^{p.c.} \leftarrow$ sample points for estimating privacy cost
$\mathbf{g}_t(x_i) \leftarrow \nabla_w \mathcal{L}(w^{(t-1)}, x_i), \forall i \in \mathcal{B}_t^{p.c.}$ ▷ Compute gradients for BA
$\hat{c}_t \leftarrow \text{AccumulatePrivacyCost}(\hat{c}_{t-1}, \{\mathbf{g}_t(x_i)\})$ ▷ Estimate privacy cost

$\mathcal{B}_t \leftarrow$ sample points with probability q
$\mathbf{g}_t \leftarrow \nabla_w \mathcal{L}(w^{(t-1)}, \mathcal{B}_t)$ ▷ Compute gradient for SGD
$\bar{\mathbf{g}}_t \leftarrow \mathbf{g}_t / \max\left(1, \frac{\|\mathbf{g}_t\|_2}{C}\right)$ ▷ Clip gradients
$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{B}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$ ▷ Add noise
$w^{(t)} \leftarrow w^{(t-1)} - \eta_t \tilde{\mathbf{g}}_t$ ▷ Make a gradient step
end for
$(\varepsilon_\mu, \delta_\mu) \leftarrow \text{GetPrivacy}(\hat{c}_T)$
Output: $w^{(T)}$, $(\varepsilon_\mu, \delta_\mu)$.

Algorithm 2 summarises privacy accounting for Bayesian DP in a pseudo-code example with DP-SGD.

5.4.6 Discussion

Relation to DP

To better understand how the BDP bound relates to the traditional DP, consider the following conditional probability:

$$\Delta(\varepsilon, x') = \Pr\left[L_{\mathcal{A}}(w; D, D', \xi) > \varepsilon \mid D, D' = D \cup \{x'\}\right]. \tag{5.46}$$

The moments accountant outputs δ that upper-bounds $\Delta(\varepsilon, x')$ for all x'. It is not true in general for other accounting methods, but let us focus on MA, as it is by far the most popular. Consequently, the MA bound is

$$\max_{x'} \Delta(\varepsilon, x') \leq \delta, \tag{5.47}$$

where ε is a chosen constant. At the same time, BDP bounds the probability that is not conditioned on x', but we can transform one to another through marginalisation and get:

$$\mathbb{E}_{x'}\left[\Delta(\varepsilon, x')\right] \leq \delta_\mu. \tag{5.48}$$

Since $\Delta(\cdot)$ is a non-negative random variable in x, we can apply Markov's inequality and obtain a tail bound on it using δ_μ. *We can therefore find a pair $(\varepsilon, \delta)_p$ that holds for any percentile p of the data distribution, not just in expectation.* In our experiments in Section 5.5, we consider bounds well above 99th percentile, so it is very unlikely to encounter data for which the equivalent DP guarantee doesn't hold. Moreover, it is possible to characterise privacy by building a curve for different percentiles, and hence, gain more insight into how well users and their data are protected.

Relation to Random DP

Continuing the logic of the previous section, we can relate our notion to random DP (Hall et al., 2011). To do so, consider also dependence on D: $\Delta(\varepsilon, x', D)$.

First, recall the definition of (ε, γ)-random DP:

$$\Pr\left[e^{-\varepsilon} \leq \frac{\Pr[\mathcal{A}(D) \in \mathcal{S}]}{\Pr[\mathcal{A}(D') \in \mathcal{S}]} \leq e^{\varepsilon}\right] \geq 1 - \gamma, \tag{5.49}$$

where D and D' are neighbouring datasets, drawn i.i.d. from some common distribution P, and the probability is w.r.t. the $n+1$-fold product measure P^{n+1}.

Extending the derivations above, for Bayesian DP bound, we have

$$\max_D \mathbb{E}_{x'}\left[\Delta(\varepsilon, x', D)\right] \leq \delta_\mu, \tag{5.50}$$

while for (ε, γ)-random DP bound,

$$\mathbb{E}_D\left[\mathbb{E}_{x'}\left[\Delta(\varepsilon, x', D)\right]\right] \leq \gamma. \tag{5.51}$$

In some cases, Bayesian DP and random DP can actually be equivalent. For example, when private outcomes for x' are independent of the rest of the dataset D.

Relation to Moments Accountant and RDP

As mentioned in Section 5.4.2, omitting the expectation over the data distribution and further simplifying Eq. 5.9, we can recover the relation between Rényi differential privacy and (ε, δ)-DP. Given the connections between RDP and various instantiations of concentrated DP (Mironov et al., 2019), we can establish analogous relations between BDP and CDP.

At the same time, our accounting technique closely resembles the moments accountant. In fact, we can show that the moments accountant is a special case of Theorem 3. The proof sketch goes as follows. Imagine we do not possess any information on the data

distribution. Then a sensible choice in our framework would be to assume an improper uniform data prior over the entire space. In order to match the moments accountant bound, we need $\|g_t - g_t'\| = C$ in Theorem 3. Due to the gradient clipping, $\|g_t - g_t'\|$ is always $\leq C$, but there is a set of gradients, for which it is $< C$. However, this is a bounded set with finite probability mass, and because of the improper uniform prior, there are infinitely many sets with the same probability mass that have $\|g_t - g_t'\| = C$. Effectively, it amounts to ignoring the data distribution information and substituting the expectation by $\max_{D,D'}$, which is the exact moments accountant bound.

Sensitivity

One may notice that throughout the chapter, apart from Definition 8, we did not mention an important concept of differential privacy—*sensitivity*. Indeed, bounded sensitivity is not as essential for Bayesian differential privacy, because extreme individual contributions are mitigated by their low probability. However, in practice it is still advantageous to restrict sensitivity in order to have a better control of the accumulated privacy loss and avoid unwanted spikes. We investigate this aspect in Section 5.5.1. Moreover, bounding sensitivity ensures that the privacy mechanism is also differentially private and provides guarantees for data for which the additional assumptions do not hold.

Privacy of $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$

Due to calculating $\hat{c}_t(\lambda)$ from data, our privacy guarantee becomes data-dependent and may potentially leak information. To obtain a theoretical bound on this leakage, we need to get back to the maximum entropy assumption in Section 5.4.5, and assume that $M(t)$ and $S(t)$ are following some specific distributions, such as Gaussian and χ^2 distributions. Hence, in case of simple random sampling, these statistics for two neighbour datasets are differentially private and the privacy parameters can be computed using Rényi divergence. Furthermore, these guarantees are controlled by the number of samples used to compute the statistics: the more samples are used, the more accurate the statistics are and the less privacy leakage occurs. This property can be used to control estimates privacy without sacrificing their tightness, only at the cost of extra computation time. Without distributional assumptions, the bound can be computed in the limit of the sample size used by the estimator, using the CLT.

Another possible solution could be based on computing the estimator from noisy data, ensuring the same level of privacy as the trained model. One can also prove that it does not result in underestimation of the real privacy cost. However, based on our preliminary experiments, this approach requires more investigation of its practicality because the obtained bounds are looser.

Finally, one should consider the fact that the information from many high-dimensional

vectors gets first compressed down to their pairwise distances, which are not as informative in high-dimensional spaces (i.e. the curse of dimensionality), and then down to one number. We believe that at this rate of compression very little knowledge can be gained by an attacker in practice.

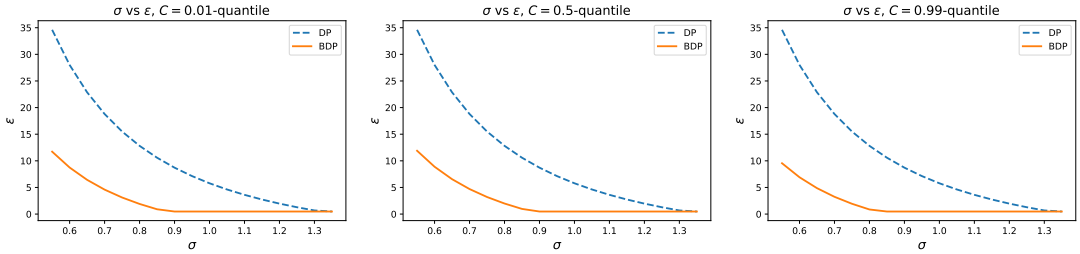
The first approach would provide little information about real-world cases due to potentially unrealistic assumptions, and the second one is too loose in estimation. Hence, we opt for the third approach. We examine pairwise gradient distances of the points within the training set and outside, and demonstrate that the privacy leakage is not statistically significant in practice (see Section 5.5.2).

5.5 Evaluation

This experimental section comprises three parts. First, we study the behaviour of Bayesian differential privacy and the Bayesian accountant on synthetic data. More specifically, we examine the trade-off between the added noise and the privacy guarantee, and how well mechanisms compose over multiple steps. We compare these to the state-of-the-art DP results obtained by the moments accountant (Abadi et al., 2016). Second, we consider the context of deep learning. In particular, we use the differentially private stochastic gradient descent (DP-SGD), a well known privacy-preserving learning technique broadly used in combination with the moments accountant, to train neural networks on classic image classification datasets MNIST and CIFAR10. We then compare the accuracy and privacy guarantees obtained under BDP and under DP. Finally, we study variational inference, a popular probabilistic method that is largely overlooked in private learning research, and adapt our framework to it.

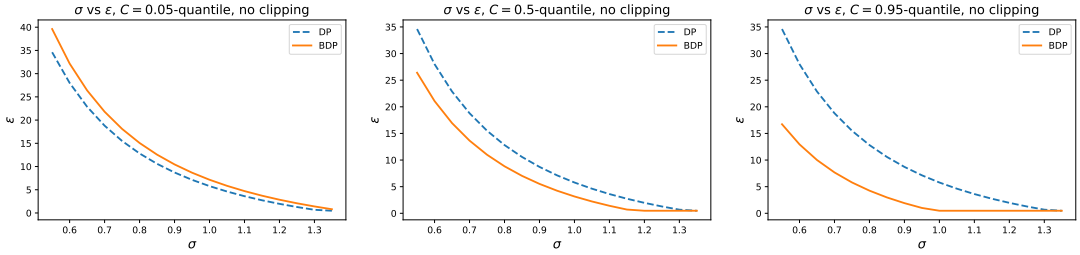
In Sections 5.5.1 and 5.5.2, we use the Gaussian noise mechanism with standard deviation $2C\sigma$. We assume that the input to the privacy mechanism is a batch of gradients. For the moments accountant, these gradients are always clipped to C before adding noise (i.e. the gradients are scaled, such that their $L2$ norm does not exceed C). For the Bayesian accountant, it may be clipped or not, which is always stated over figures or in the text. Whenever the privacy mechanism is invoked repeatedly, Theorem 1 is used to compute the Bayesian ε over multiple learning iterations (*steps*).

As stated above, DP and BDP can use the same privacy mechanism and be accounted in parallel to ensure the DP guarantees hold if BDP assumptions fail. Thus, all comparisons in this section should be viewed in the following way: the reported BDP guarantee would apply to *typical* data (all data drawn from the same distribution as the dataset); the reported DP guarantee would apply to all other data; their difference is the advantage we gain by using BDP for typical data. In some experiments we use smaller noise variance for BDP in order to speed up training, meaning that the reported BDP guarantees will further improve if noise variance is increased to DP levels.



(a) $C = 0.01$-quantile of $\|\nabla f\|$. (b) $C = 0.50$-quantile of $\|\nabla f\|$. (c) $C = 0.99$-quantile of $\|\nabla f\|$.

Figure 5.1 – Dependency between σ and ε for different C when clipping for both DP and BDP.



(a) $C = 0.05$-quantile of $\|\nabla f\|$. (b) $C = 0.50$-quantile of $\|\nabla f\|$. (c) $C = 0.95$-quantile of $\|\nabla f\|$.

Figure 5.2 – Dependency between σ and ε for different C when clipping for DP and not clipping for BDP.

Remark 9. Running Bayesian accountant in a forward manner, as we do in this section and remaining chapters, only computes privacy guarantees for a particular "training context"—a specific path taken by the optimiser in the parameter space—and indicates what guarantees can be achieved in principle. This is done for comparison purposes. A proper implementation would set a bound upfront and ensure that it is not exceeded, regardless of the training path (maximisation over auxiliary inputs in Theorem 2 and Corollary 1). Since we observe very little privacy loss variance in our experiments, we believe the reported guarantees are not far from the actual.

5.5.1 Behaviour of Bayesian Differential Privacy

Let us start by studying the behaviour of Bayesian differential privacy with regards to its parameters, as well as in comparison to classic differential privacy. All experiments in this section are carried out on synthetic data, but as we show in the next section these results hold for real data. The synthetic gradients are drawn from the Weibull distribution with the shape parameter < 1 to imitate a more difficult case of heavy-tailed gradient distributions.

Effect of σ and bounded sensitivity

The primary goal of our research is to obtain more meaningful privacy guarantees sacrificing as little utility as possible. The main factor in the loss of utility is the variance of the noise we add during training. Therefore it is critical to examine how our guarantee behaves compared to the classic DP for the same amount of noise. Or equivalently, how much noise does it require to reach the same ε.

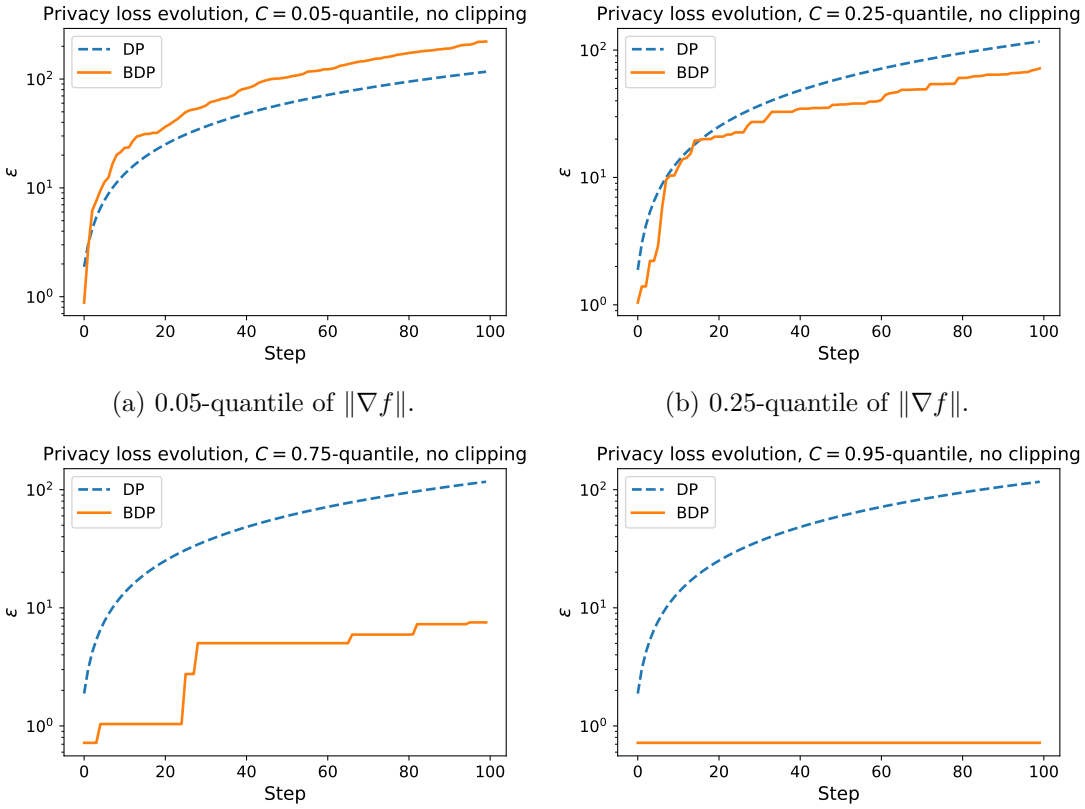
As stated above, there are two possible regimes of operation for the Gaussian noise mechanism under Bayesian differential privacy: with bounded sensitivity and with unbounded sensitivity. The first is just like the classic DP: there is a maximum bound on the contribution of an individual example, and the noise is scaled to it. The second does not have a bound on contribution and mitigates it by taking into account the low probability of extreme contributions.

Figures 5.1 and 5.2 demonstrate the dependency between σ and ε for different clipping thresholds C chosen relative to the quantiles of the gradient norm distribution. If we bound sensitivity by clipping the gradients, it ensures that BDP always requires less noise than DP to reach the same ε, as seen in Figure 5.1. As we decrease the clipping threshold C, more and more gradients get clipped and the BDP curve approaches the DP curve (Figure 5.1a). However, as we observe in Figure 5.2 comparing DP with bounded sensitivity and BDP with unbounded sensitivity, using unclipped gradients results in less consistent behaviour. It may require a more thorough search for the right noise variance to reach the same ε.

Composition

In this experiment, we study the growth rate of the privacy loss over a number of invocations of the mechanism. We do not clip gradients for BDP in order to show the raw effect of the signal-to-noise ratio on the privacy loss evolution behaviour.

In Figure 5.3, we plot ε as a function of steps for different levels of noise. Naturally, as the noise standard deviation gets closer to the expected gradients norm, the growth rate of the privacy loss decreases dramatically. Even when the noise is at the 0.25-quantile, the Bayesian accountant matches the moments accountant. It is worth noting, that DP behaves the same in all these experiments because the gradients get clipped at the noise level C. Introducing clipping for BDP yields the behaviour of Figure 5.3d, as we demonstrate in the next section on real data.



(a) 0.05-quantile of $\|\nabla f\|$.

(b) 0.25-quantile of $\|\nabla f\|$.

(c) 0.75-quantile of $\|\nabla f\|$.

(d) 0.95-quantile of $\|\nabla f\|$.

Figure 5.3 – Evolution of ε over multiple steps of the Gaussian noise mechanism with $\sigma = C$ for DP (with clipping) and BDP (without clipping). Sub-captions indicate the noise variance relative to the gradient norms distribution.

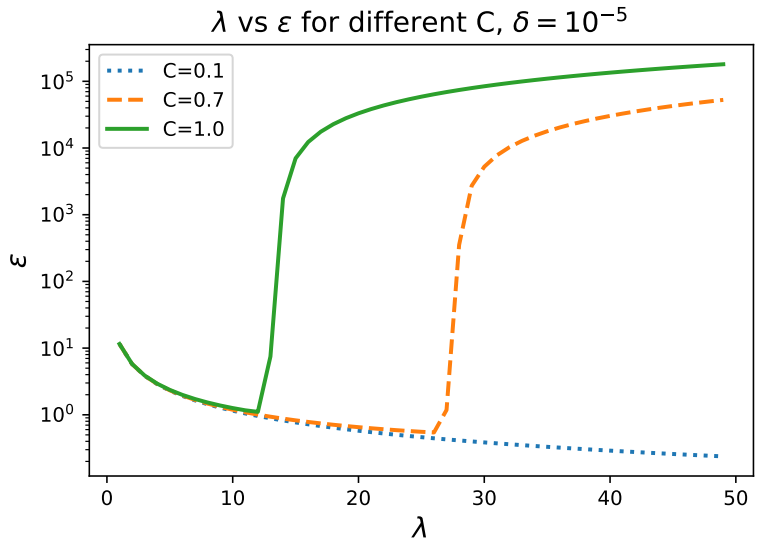


Figure 5.4 – Dependency of λ and ε for different clipping thresholds C, $q = 64/60000$, $\sigma = 1.0$.

Effect of λ

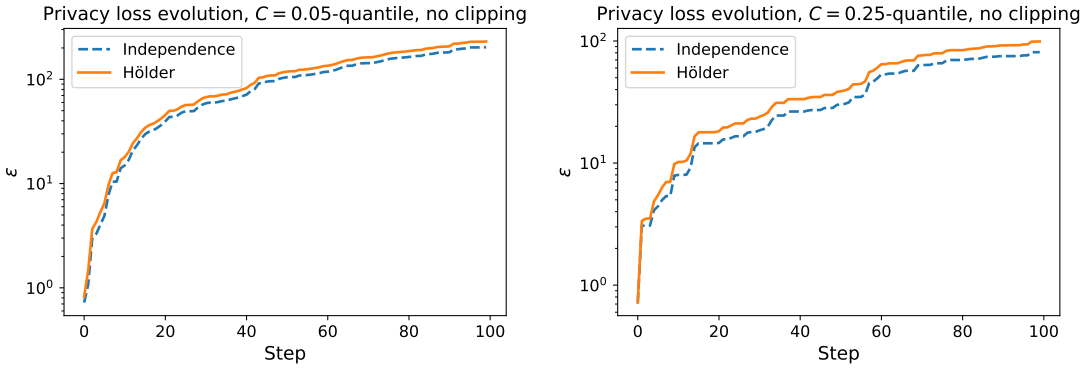
As mentioned in Section 5.4.2, the privacy cost, and therefore the final value of ε, depend on the choice of λ. We run the Bayesian accountant for the Gaussian mechanism with the fixed pairwise gradient distances (s.t. these results apply exactly to the moments accountant) for different signal-to-noise ratios and different λ.

Depicted in Figure 5.4 is ε as a function of λ for 10000 steps. We observe that λ has a clear effect on the final ε value. In some cases this effect is very significant and the change is sharp. It suggests that in practice one should be careful about the choice of λ. We also note that for lower signal-to-noise ratios (e.g. $C = 0.1, \sigma = 1$) the optimal choice of λ is much further on the real line and may well be outside the typically range computed in the literature.

Effect of Hölder's inequality in Theorem 1

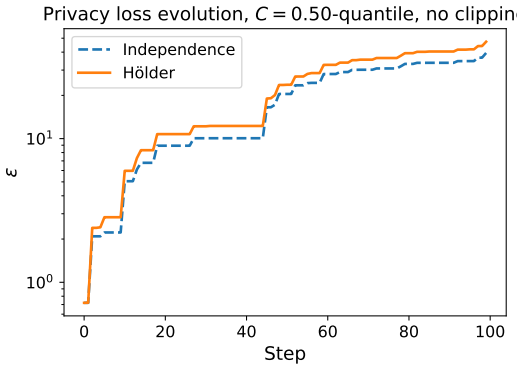
In the advanced composition theorem, we use generalised Hölder's inequality in order to decompose the total privacy cost of the algorithm into privacy costs of each iteration. Its use is necessitated by the fact that the additional training example that we are trying to protect stays the same throughout all invocations of the privacy mechanism. But it raises a question about the sharpness of this bound, and we address it by performing a comparison with the naïve solution of simply swapping the produce and the expectation.

We again run experiments with synthetic gradients, generated by Weibull distribution, and

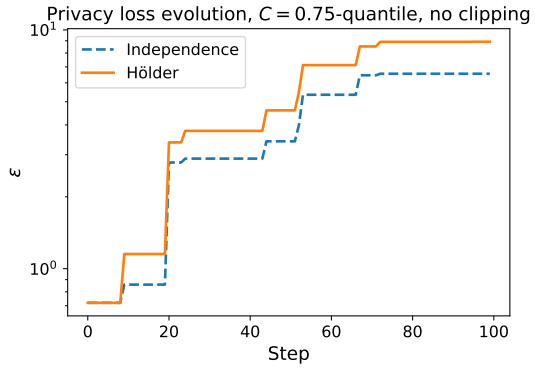


(a) 0.05-quantile of $\|\nabla f\|$.

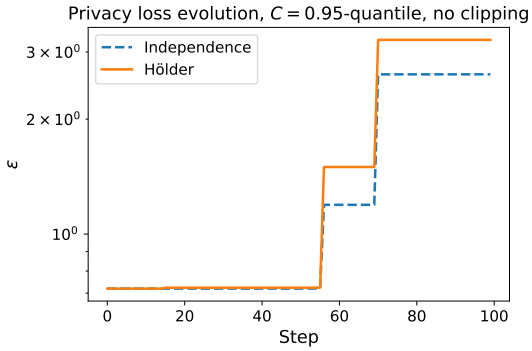
(b) 0.25-quantile of $\|\nabla f\|$.



(c) 0.50-quantile of $\|\nabla f\|$.



(d) 0.75-quantile of $\|\nabla f\|$.



(e) 0.95-quantile of $\|\nabla f\|$.

Figure 5.5 – Illustration of the effect of Hölder's inequality in Theorem 1. Each graph depicts ε over multiple steps of the Gaussian noise mechanism with $\sigma = C$ for BDP with independence assumption, and for BDP with Hölder's inequality. Sub-captions indicate the noise variance relative to the gradient norms distribution.

compare two privacy curves: the one obtained using the theorem with Hölder's inequality, and the other one obtained by naïvely swapping the product and the expectation over the data distribution. As shown in Figure 5.5, the two curves remain close for the duration of the algorithm, even in more difficult cases where the privacy loss spikes due to outliers. In principle, larger spikes of privacy loss could lead to a more noticeable difference between these two cases; however, for practical guarantees, one should choose the privacy mechanism parameters such that sizeable spikes are ruled out or kept to the minimum in any case.

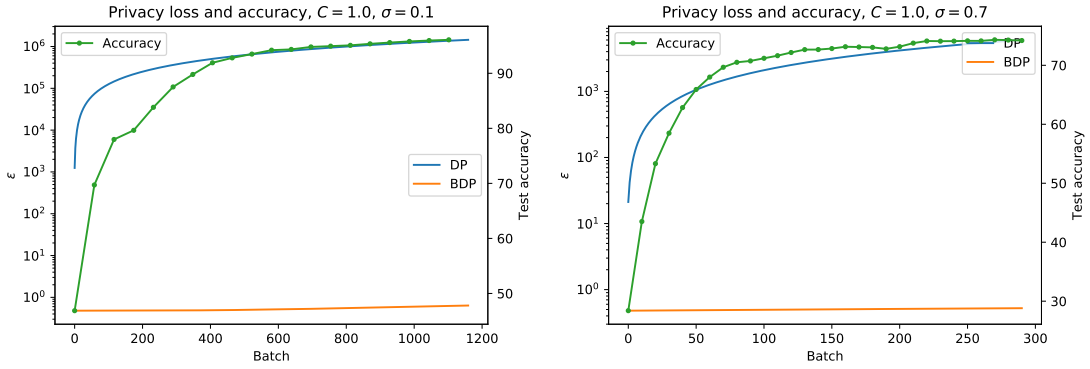
5.5.2 Deep Learning

In this section, we consider the application to privacy-preserving deep learning. Our setting closely mimics that of (Abadi et al., 2016) to enable a direct comparison with the moments accountant and DP. We use a version of DP-SGD (Abadi et al., 2016) that has been extensively applied to build differentially private machine learning models, from deep neural networks to Bayesian learning. The idea of DP-SGD is simple: at every iteration of SGD, clip the gradient norm to some constant C (ensuring bounded sensitivity), and then add Gaussian noise with variance $C^2\sigma^2$.

We train a classifier represented by a neural network (unlike (Abadi et al., 2016), without PCA) on MNIST (LeCun et al., 1998) and on CIFAR10 (Krizhevsky, 2009) using DP-SGD. The first dataset contains 60,000 training examples and 10,000 testing images. We use large batch sizes of 1024, clip gradient norms to $C = 1$, and $\sigma = 0.1$. The second dataset consists of 50,000 training images and 10,000 testing images of objects split in 10 classes. For this dataset, we use the batch size of 512, $C = 1$, and $\sigma = 0.7$. We fix $\delta = 10^{-5}$ in all experiments. In case of CIFAR10, in order for our results to be comparable to (Abadi et al., 2016), we pre-train convolutional layers of the model on a different dataset and retrain a fully-connected layer in a privacy-preserving manner.

Let us briefly outline how DP-SGD works in conjunction with the privacy accountant. The non-private learning outcome at each iteration t is the gradient g_t of the loss function w.r.t. the model parameters, the outcome distribution is the Gaussian $\mathcal{N}(g_t, \sigma^2 C^2)$. Before adding noise, the norm of the gradients is clipped to C. For the moments accountant, privacy loss is calculated using this C and σ. For Bayesian accountant, either pairs of examples x_i, x_j or pairs of batches are sampled from the dataset at each iteration, and used to compute $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$. Although clipping gradients is no longer necessary with BDP, as stated in Section 5.5.1, it is highly beneficial for incurring lower privacy loss at each iteration and obtaining tighter composition. Moreover, as previously discussed, it ensures the classic DP bounds on top of BDP bounds.

Figures 5.6a and 5.6b demonstrate the evolution of ε bound over training batches computed on MNIST and CIFAR10 datasets accordingly. The first observation is that



(a) MNIST. (b) CIFAR10.

Figure 5.6 – Evolution of ε for $\delta = 10^{-5}$ when training CNN.

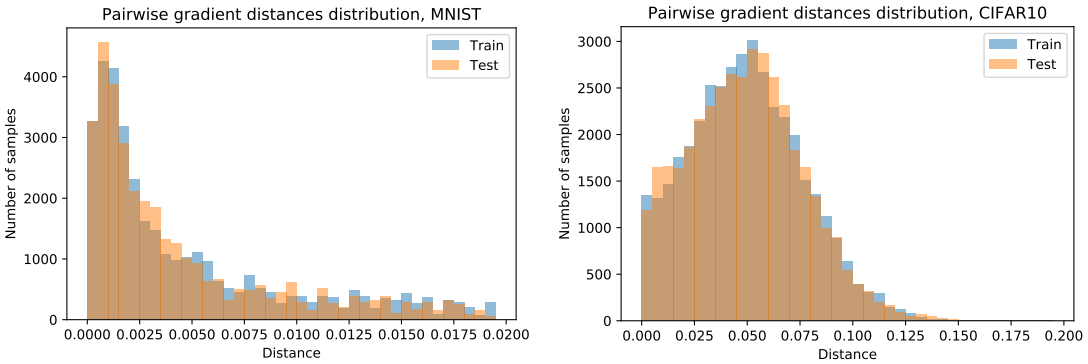
Table 5.2 – Estimated privacy bounds ε with $\delta = 10^{-5}$ for DP, $\delta_\mu = 10^{-5}$ and $\delta_\mu = 10^{-10}$ for BDP (marked as BDP and BDP* accordingly). BDP* bound corresponds to DP bound for 0.99999-quantile of the data distribution.

	Accuracy		ε			$P(A)$		
Dataset	Baseline	Private	DP	BDP	BDP*	DP	BDP	BDP*
MNIST	99%	96%	2.18	**0.62**	**0.95**	89.84%	**65.02%**	**72.1%**
CIFAR10	86%	73%	8.0	**0.51**	**0.76**	99.97%	**62.48%**	**68.1%**
Abalone	77%	76%	7.6	**0.5**	**0.61**	99.95%	**62.25%**	**64.9%**
Adult	81%	81%	0.5	**0.16**	**0.2**	62.25%	**53.99%**	**55.0%**

Bayesian differential privacy allows to add far less noise to achieve comparable ε. Because of this, the models reach the same test accuracy much faster. For example, our model reaches 96% accuracy within 20 epochs for MNIST, while DP model requires hundreds of epochs to avoid ε blowing up, like it is shown in Figures 5.6a and 5.6b. These results also confirm our assumption that the actual disagreement between gradient directions is much smaller than their norms, and therefore, requires less noise to hide.

Overall, using the information about gradient distribution allows the BDP models to reach the same accuracy at a much lower ε. On MNIST, we manage to reduce it from 2.18 to 0.62. For CIFAR10, from 8.0 to 0.51. See details in Table 5.2. To make our results more transparent, we include the potential attacker success probability $P(A)$ from Section 5.3.3 computed using Eq. 5.1. In this interpretation, the benefits of using Bayesian differential privacy become even more apparent.

An important aspect of BDP, discussed in Section 5.4.6, is the potential privacy leakage of the privacy cost estimator. Since at the moment we do not have a rigorous bound on the amount of information it leaks, we conduct the following experiment. After training the model (to ensure it contains as much information about data as possible), we compute



(a) MNIST dataset.

(b) CIFAR10 dataset.

Figure 5.7 – Histograms of pairwise gradient distances for points within and outside of the training set.

the gradient pairwise distances over train and test sets. We then plot the histograms of these distances to inspect any divergence that would distinguish the data that was used in training. Note that this is more information than what is available to an adversary, who only observes ε.

As it turns out, these distributions are nearly identical (see Figures 5.7a and 5.7b), and we do not observe any correlation with the fact of the presence of data in the training set. For example, the sample mean of the test set can be both somewhat higher or lower than that of the train set. We also run the t-test for equality of means and Levene's test for equality of variances, obtaining p-values well over the 0.05 threshold, suggesting that the difference of the means and the variances of these distributions is not statistically significant and the equality hypothesis cannot be rejected.

5.5.3 Variational Inference

While DP-SGD is widely applicable, some machine learning and statistical inference techniques do not require additional noise at all. For example, it has been shown that differential privacy guarantees arise naturally and "for free" in methods like sampling from the true posterior (Dimitrakakis et al., 2014) and Stochastic Gradient MCMC (Wang et al., 2015). Using Bayesian privacy accounting we can show that another popular Bayesian approach–variational inference–also enjoys almost "free" privacy guarantees.

The goal of variational inference is to approximate a posterior distribution $p(w|D)$ by a member of a known family of "simple" distributions $q(w;\theta)$ parametrised by θ. Most commonly, it is done via minimising the reverse KL-divergence $\mathcal{D}_{KL}(Q||P)$, but there are a lot of modern variations, for example using χ-divergence (Dieng et al., 2017), Rényi divergence (Li and Turner, 2016), or other variational bounds (Chen et al., 2018).

As baselines, we use DPVI-MA (Jälkö et al., 2016) and DP-SGLD (Wang et al., 2015). The first one employs DP-SGD combined with moments accountant to train a private VI model, while the second is a stochastic gradient MCMC method achieving DP due to the noisy nature of SGLD algorithm. Following (Jälkö et al., 2016), we run evaluation on two classification tasks taken from UCI database: Abalone (Waugh, 1995) and Adult (Kohavi, 1996). Both are binary classification tasks: predicting the age of abalone from physical measurements, and predicting income based on a person's attributes. They have 4,177 and 48,842 examples with 8 and 14 attributes accordingly. We use the same pre-processing and models as (Jälkö et al., 2016).

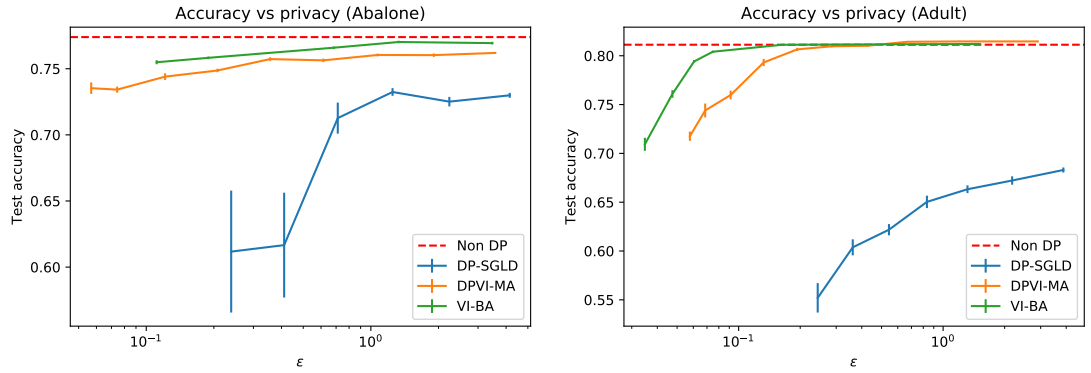
To translate variational inference to the language of our privacy accountant, $q(w;\theta)$ is the outcome distribution, and we are interested in

$$\mathbb{E}_{x'}\left[e^{\lambda \mathcal{D}_{\lambda+1}(q(w;\theta)||q(w;\theta'))}\right],$$

where θ, θ' are variational parameters learnt from D and D'. At each learning iteration, $w^{(t)}$ are sampled from $q(w;\theta^{(t-1)})$, updates are computed using the variational bound and data D (or its subsamples), and parameters are updated to $\theta^{(t)}$. Therefore, for Bayesian accounting, we sample x, x' and w from D and $q(w;\theta^{(t-1)})$ and compute $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$. The exact expression is slightly different from the one for the classical Gaussian noise mechanism because of non-identical variances. To derive it, one could either do it directly by following the steps of Theorem 3, or plug the Rényi divergence formula for two normal distributions in the expressions for the general subsampled mechanism (Theorem 4).

To enable differential privacy for variational inference methods, we have to deal with two important restrictions. First, parameters θ of variational distribution $q(w;\theta)$ are not differentially private and need to be concealed or made private by other means. Second, as a result of the previous point, MAP or MLE estimates based on θ would also reveal private information. However, samples $w \sim q(w;\theta)$ are differentially private and can be used to perform the same tasks. We haven't observed significant loss of accuracy when using a batch of samples w instead of true parameters θ, and thus, we consider it a minor cost. Note also that each sample needs to be accounted for, both in training and after training. In our tests, we run logistic regression using an average of up to 10 samples from variational distribution, significantly less than what is necessary to recover true variational parameters.

We observe in Figures 5.8a and 5.8b that our modified variational inference with Bayesian accountant achieves consistent advantage over DPVI-MA and DP-SGLD both in terms of accuracy and privacy accounting. It is the only method reaching non-DP accuracy on Abalone data and the first to reach it on Adults data, at a fraction of other methods' privacy budget. At any point, the trade-off curve of our technique remains above others. Moreover, the test variance of our approach (computed over 10 trials) is considerably



(a) Abalone dataset.

(b) Adult dataset.

Figure 5.8 – Accuracy-privacy trade-off for variational inference (logistic regression model) with Bayesian DP compared to prior work.

smaller, presumably because there is no noise added in the learning process.

Privacy loss bounds for the same levels of accuracy can be found in Table 5.2. Similarly to the deep learning scenario, the Bayesian accountant with its access to the distribution of gradients has a remarkable advantage. It is also worth mentioning, that for our methods we decreased δ to 10^{-5} on Adult dataset, because the failure probability 10^{-3} originally set in (Jälkö et al., 2016) is too high for almost 50k samples.

5.6 Conclusion

In this chapter, we introduced the notion of $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy, a relaxation of (ε, δ)-differential privacy for sensitive data that are drawn from an arbitrary (and unknown) distribution $\mu(x)$. This relaxation is reasonable in many machine learning scenarios where models and algorithms are designed for and trained on specific data distributions (e.g. emails, face images, ECGs, etc.). For example, it may be unjustified to try hiding the absence of music records in a training set for ECG analysis, because the probability of them appearing there is actually much smaller than δ.

We state and proof the advanced composition theorem for Bayesian differential privacy that allows for efficient and tight privacy accounting. Since the data distribution is unknown, we design an estimator that overestimates the privacy loss with high, controllable probability. Moreover, as the data sample is finite, we employ the Bayesian parameter estimation approach with the flat prior and the maximum entropy principle to reduce the chance of underestimating probabilities of unseen examples. As a result, our interpretation of δ_μ is slightly different: not only is it the probability of the privacy loss exceeding ε_μ in the tails of its distribution, but it also is the probability of underestimating the privacy loss based on a finite sample of data.

Our evaluation confirms that Bayesian differential privacy is highly beneficial in machine learning context where the additional assumptions on data distribution are naturally satisfied. First, it requires less noise to reach the same privacy guarantees. Second, as a result, models train faster and can reach higher accuracy. Third, it may be used along with DP to perform tighter analysis of privacy budget for the population while still maintaining general DP guarantees. In our deep learning experiments with convolutional neural networks and variational inference experiments, ε_μ always remained well below 1, allowing for much more meaningful bounds on the potential attacker success probability.

6 Federated Learning with Bayesian Differential Privacy

6.1 Introduction

After having established the novel definition of privacy, more tailored for machine learning and providing a superior balance between privacy and utility, we return to the setting of *federated learning (FL)* (McMahan et al., 2016). Unlike Chapter 4, here we focus on discriminative models and seek to adapt and evaluate the notion of Bayesian differential privacy in this context.

As shown in recent work (McMahan et al., 2017), federated learning can be combined with differential privacy to provide joint benefits. However, unless the number of users is exceedingly high (e.g. in the scenario of a large population of mobile users considered in (McMahan et al., 2017)), differentially private federated learning provides only weak guarantees. As a reminder, contrary to a wide-spread opinion in machine learning community, values of ε close to 10 can hardly be seen as reassurance to a user: for certain types of attacks, an adversary can theoretically reach accuracy of 99.99%.

We propose to augment federated learning with Bayesian differential privacy instead. It would allow to provide tighter, and thus, more meaningful guarantees. The key idea remains the same and is based on the observation that federated learning tasks, just like the centralised one, are often specialised on a particular type of data (for example, finding a film review in the MRI dataset is very unlikely). The differentiating feature of FL is that these data can be generated by a set of non-identical distributions associated with individual users or groups of users. We demonstrate that this characteristic does not prevent the use of BDP, and that one can account privacy using the mixture of user data distributions. Furthermore, we consider different levels of privacy in federated learning (client level and instance level) and how to jointly provide and quantify privacy at both levels.

This chapter is based on the paper published in 2019 IEEE International Conference on Big Data (Triastcyn and Faltings, 2019b).

We extend the notion of Bayesian differential privacy to the federated learning setting in Section 6.4. Our experiments, in Section 6.5, show significant advantage, both in privacy guarantees and the model quality.

The main contributions of this chapter are the following:

- we adapt the notion of Bayesian differential privacy to federated learning, including more natural non-i.i.d. settings (Section 6.4.1), to provide strong theoretical privacy guarantees under minor and practical assumptions;
- we propose a novel joint accounting method for estimating client-level and instance-level privacy simultaneously and securely (Section 6.4.3);
- we experimentally demonstrate advantages of our method, such as shrinking the privacy budget to a fraction of the previous state-of-the-art, and improving the accuracy of the trained models by up to 10% (Section 6.5).

6.2 Related Work

We provided a short overview of the federated learning research in Chapter 4, and more details can also be found in (Yang et al., 2019). Here, we will focus more narrowly on the literature related to privacy protection in federated learning, and particularly, providing formal privacy guarantees.

The first example of augmenting federated learning with theoretical privacy guarantees is given by McMahan et al. (2017). They train large RNNs with client-level differential privacy, using `DP-FedAvg` and `DP-FedSGD`, inspired by DP-SGD, as well as the moments accountant (Abadi et al., 2016). Experiments show that strong privacy guarantees for large populations of users can be achieved with only negligible loss in prediction accuracy. This approach is well suited for many large-scale scenarios at Google, Apple, and other technology companies with massive client bases. For example, Snap Inc. deployed a distributed learning framework under an even stronger local privacy model (Pihur et al., 2018). However, in smaller-scale scenarios, privacy guarantees will be too loose to be practical.

An earlier attempt to design a solution for settings with fewer participants (Geyer et al., 2017) does not ideologically differ from the aforementioned work. Geyer et al. similarly implement a differentially private version of `FedAvg` and use the moments accountant, but experiment with a smaller set of clients. However, the chosen ε is too large for meaningful guarantees and the paper appears to contain some errors that may invalidate the privacy guarantee (e.g. tuning the clipping threshold using the gradient norm median).

Since differential privacy leads to a poor privacy-utility trade-off for more complicated

models and small-scale scenarios, pairing federated learning with an alternative privacy notion might prove beneficial.

Truex et al. (2019) propose to combine differential privacy with secure multi-party computation in a hybrid approach to reduce the amount of noise necessary for maintaining strong privacy guarantees. Their experiments show advantages over the previous local DP approaches and some are performed with just 10 participating clients. However, they report declining accuracy for other methods with the increase in participants given the unchanged privacy level, which goes against the conventional wisdom that privacy is easier to achieve with larger populations. Intuitively, even using the local privacy model should not lead to increases in privacy budget in this case. Furthermore, the deep learning related evaluation on MNIST only considers a binary classification task between digits 0 and 9, which is a considerably simpler problem compared to the full 10-digit classification, prohibiting a direct comparison to other centralised and federated methods. Consequently, it appears that this work needs further investigation.

Another "hybrid" approach is distributed differential privacy (Shi et al., 2011; Rastogi and Nath, 2010). In essence, this technique provides guarantees in the central model of differential privacy, but the noise is added by clients, like in the local model. It is accomplished by using a specifically designed cryptographic scheme, achieving aggregator obliviousness, and the noise distribution satisfying algebraic constraints for multi-user composition of privacy mechanisms (e.g. geometric distribution). The latter condition is necessary due to discretisation required by homomorphic encryption. Providing a middle ground between local and global DP models, this approach brings a lot of promise in federated learning scenarios. Applying a similar approach to our privacy definition when shifting the noise addition to the client side is a promising future research direction.

Finally, our observations and contributions about *instance privacy* versus *client privacy* can be associated with the considerations in differentially private meta learning (Li et al., 2020). More specifically, the notions of *Task-Global DP* and *Task-Local DP* have definitive parallels with instance privacy, because they also focus on protecting individual training examples rather than entire model updates submitted by clients.

6.3 Setting

In Chapter 5, while describing the concept of Bayesian differential privacy, we considered a general iterative learning algorithm, such that each iteration t produces a non-private learning outcome $g^{(t)}$ (e.g. a gradient over a batch of data). In this chapter, we consider the equivalent federated learning setting, where each communication round t produces a set of non-private learning outcomes $u_i^{(t)}$, one for each client i.

The non-private outcome, similarly to the previous chapter, gets transformed into a

private learning outcome $w^{(t)}$ that is used as a starting point for the next iteration or communication round. The learning outcome can be made private by different means, but in this work we consider the most common approach of applying an additive noise mechanism (e.g. a Gaussian noise mechanism). We denote the distribution of private outcomes by $p(w^{(t)}|w^{(t-1)}, D)$ (assuming the Markov property of the learning process for brevity of notation, although it is not necessary in general) or $p(w^{(t)}|w^{(t-1)}, \mathbb{U})$, depending on the scenario.

During the training process, we can sample subsets of clients in each communication round. In this case, $w^{(t)}$ comes from the distribution $p(w^{(t)}|w^{(t-1)}, \mathbb{U}^{(t)})$, where $\mathbb{U}^{(t)}$ is a set of updates from users participating in the communication round t. Privacy is then amplified through sampling (see Sections 5.4.3, 5.4.4 for Bayesian DP; and for DP, Abadi et al. (2016); Balle et al. (2018)).

For each iteration, we would like to compute a quantity c_t (we call it a *privacy cost*) that accumulates over the learning process and allows to compute privacy loss bounds ε, δ using concentration inequalities. The overall privacy accounting workflow does not significantly differ from Chapter 5 conceptually, but is adapted to federated learning setting and may potentially be enhanced with secure aggregation schemes.

6.4 Federated Learning with Bayesian Differential Privacy

In this section, we adapt the Bayesian differential privacy framework and its accounting method to guarantee the client-level privacy, the level most frequently addressed in the literature. We then justify and explore the instance-level privacy and two different techniques for accounting it. Finally, we propose a method to jointly account instance-level and client-level privacy for the `FedSGD` algorithm in order to provide the strongest trade-off between utility and privacy guarantees.

6.4.1 Client Privacy

When it comes to reinforcing federated learning with differential privacy, the foremost attention is given to the client-level privacy (McMahan et al., 2017; Geyer et al., 2017). The goal is to hide the presence of a single user, or to be more specific, to bound the influence of any single user on the learning outcome distribution (i.e. the distribution of the model parameters).

Under the classical DP (McMahan et al., 2017; Geyer et al., 2017), the privacy is enforced by clipping all user updates u_i to a fixed $L2$-norm threshold C and then adding Gaussian noise with the variance $C^2\sigma^2$. The noise parameter σ is calibrated to bound the privacy loss in each communication round, and then the privacy loss is accumulated across the rounds using the moments accountant (Abadi et al., 2016).

We use the same privacy mechanism, but employ the Bayesian accounting method instead of the moments accountant. Intuitively, our accounting method should have a significant advantage over the moments accountant in the settings where data is distributed similarly across the users because in this case their updates would be in a strong agreement. In order to map the Bayesian differential privacy framework to this setting, let us introduce some notation.

Let N denote the number of clients in the federated learning system. Every client i computes and sends to the server a model update $u_i \sim p_i(u)$ drawn from the client's update distribution $p_i(u)$. Considering individual client distributions ensures that our approach is applicable to non-i.i.d. settings that are natural in the federated learning context. Generally, not all users participate in a given communication round. We denote the probability of a user i participating in the round by α_i. Thus, the overall update distribution is given by a mixture:

$$p(u) = \sum_{i=1}^{N} \alpha_i p_i(u). \tag{6.1}$$

In our experiments, we fix $\alpha_1 = \alpha_2 = \ldots = \alpha_N = \alpha$.

To match the notation above, let w_t indicate the privacy-preserving model update:

$$w_t \leftarrow \mathcal{A}(\{u_i | u_i \in \mathbb{U}^{(t)}\}), \tag{6.2}$$

where $\mathcal{A}(\{u_i | u_i \in \mathbb{U}^{(t)}\}) \triangleq \frac{1}{|\mathbb{U}^{(t)}|}\left(\sum_i u_i + \mathcal{N}(0, C^2\sigma^2)\right)$ in the case of Gaussian mechanism, and $\mathbb{U}^{(t)}$ is the set of updates from users participating in the round t.

To bound ε and δ of Bayesian differential privacy, one needs to compute $c_T(\lambda, \xi, D, D') = \max\left\{c_T^L(\lambda, \xi, D, D'),\ c_T^R(\lambda, \xi, D, D')\right\}$, where

$$c_T^L(\lambda, \xi, D, D') = \frac{1}{T}\log \mathbb{E}_u\left[e^{T\lambda D_{\lambda+1}(p_t \| q_t)}\right],$$
$$c_T^R(\lambda, \xi, D, D') = \frac{1}{T}\log \mathbb{E}_u\left[e^{T\lambda D_{\lambda+1}(q_t \| p_t)}\right],$$

and

$$p_t = p(w^{(t)} | w^{(t-1)}, \mathbb{U}^{(t)})$$
$$q_t = p(w^{(t)} | w^{(t-1)}, \mathbb{U}^{(t)} \setminus \{u\})$$

Since the randomness of w comes from the subsampled Gaussian noise mechanism, we use Theorem 3, in combination with user sampling and the privacy cost estimator (Definition 17) for both expressions, to obtain $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ that upper-bounds $c_T(\lambda, \xi, D, D')$ with high probability. Finally, we use Theorems 1 and 2 to compute ε and δ. The required assumption of exchangeability is naturally satisfied because users

Algorithm 3 Server-side code for FL with client-level privacy.

Input:
Clients $\{1, \cdots, N\}$, loss function $\mathcal{L}(\cdot)$.
Parameters: client sampling probability q, update clipping threshold C, noise parameter σ.

Initialise w_0 randomly, Bayesian accountant with λ, q, σ, C, T
for $t \in [1..T]$ **do**
$\mathcal{C}^{(t)} \leftarrow$ sample users with probability q, $K = qN$
for $k \in \mathcal{C}^{(t)}$ **do**
$\Delta_k^{(t)} \leftarrow \text{ClippedUserUpdate}(k, w^{(t-1)}, C)$
end for
$\Delta^{(t)} \leftarrow \text{CombinedUpdate}(\Delta_{1:K}^{(t)})$
$w^{(t)} \leftarrow w^{(t-1)} + \Delta^{(t)} + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ ▷ Update shared model
$\hat{c}_t \leftarrow \text{AccumulatePrivacyCost}(\hat{c}_{t-1}, \Delta_{1:K}^{(t)})$ ▷ Estimate privacy cost
end for
$(\varepsilon_\mu, \delta_\mu) \leftarrow \text{GetPrivacy}(\hat{c}_T)$
Output: w_T, $(\varepsilon_\mu, \delta_\mu)$.

are sampled independently and uniformly. Algorithms 3 and 4 present a pseudo-code of the overall training and privacy accounting process.

6.4.2 Instance Privacy

As noted above, we have not changed the privacy mechanism itself, and the same mechanism can be used in conjunction with the moments accountant to get the classical DP guarantees (McMahan et al., 2017; Geyer et al., 2017). In that case, (ε, δ)-DP at the client level implies the same guarantee at the instance level (i.e. bounding the influence of a single data point). However, it does not hold for Bayesian DP. Moreover, the same privacy guarantee may not be meaningful at the instance level. For example, $\delta = 10^{-3}$ might be reasonable for 100 clients, but if a client has tens of thousands of data points, it is not a reasonable failure probability at the data point level.

At the same time, instance privacy is extremely important in some scenarios. Imagine federated training on medical data from different hospitals: while a hospital participation may be public knowledge, individual patients data must be protected to the highest degree. Another reason for considering instance-level privacy is that it provides an additional layer of protection for users in case of a malicious or untrusted server.

In order to get tighter instance privacy guarantees, we apply the subsampled Gaussian noise mechanism to gradient computation on user devices. The accounting follows the same procedure as described above, except that the noise parameter σ and the sampling probability q may be different, depending on which of the settings below is used.

Algorithm 4 Client-side code for FL with client-level privacy.

Input:

Dataset $D_k = \{x_1, \cdots, x_{N_k}\}$ on client k, loss function $\mathcal{L}(\cdot)$.
Parameters: learning rate η_i, batch size B, number of local iterations I_k, gradient norm bound C.

function ClippedUserUpdate($w^{(0)}$, C)
  for $i \in [1..I_k]$ **do**
    Sample a random batch of examples $\mathcal{B}_i$
    $\mathbf{g}_i \leftarrow \nabla_w \mathcal{L}(w^{(i-1)}, \mathcal{B}_i)$ ▷ Compute gradient
    $w^{(i+1)} \leftarrow w^{(i)} - \eta_i \tilde{\mathbf{g}}_i$ ▷ Make a gradient step
  end for
  $\Delta_k \leftarrow w^{(I_k)} - w^{(0)}$ ▷ Compute update
  $\Delta_k \leftarrow \Delta_k / \max\left(1, \frac{\|\Delta_k\|_2}{C}\right)$ ▷ Clip update
  return Δ_k
end function

There are two possible accounting schemes,*sequential* and *parallel*, described below. We found that sequential accounting produces better results in our experiments; however, it may not necessarily be the case in other settings.

Sequential Accounting

Part of the accounting is performed locally on user devices and part on the server. Overall privacy cost is equivalent to the centralised training with the data sampling probability $q = \frac{B_i}{N}$, where N is the total number of data points across all users, and B_i is the local batch size.

The process proceeds as follows. At each communication round, the server sends N to participating clients, every client performs private gradient updates, computes $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$, and sends it to the server. The server then aggregates the sum of $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ from all users. Since the privacy costs are data-dependent, it is possible to use secure multi-party computation to allow the server know the sum without learning individual costs.

The disadvantage of this method is that every participating client learns the total number of data points, and especially in the settings with a small number of users it may not be desirable. Furthermore, the obtained bounds apply to the commonly learnt model but not to the individual updates of each user, requiring them to maintain a separate local bound. These issues are addressed by parallel accounting.

Parallel Accounting

In this scheme, every client computes $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ using $q = \frac{B_i}{N_i}$, where N_i is the local dataset size of the client. Consequently, since $N_i \leq N$, the privacy costs will be higher. But this is compensated by using parallel composition instead of sequential: the server aggregates the maximum of $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ over all users. Again, using secure multi-party computation is possible to prevent the server from learning individual privacy costs. However, the server would still learn one of the privacy costs—the true maximum.

Parallel composition is applicable in this scenario because user updates within a single round are independent. However, the server needs to sum up maximum privacy costs over the rounds because updates are dependent on previous rounds. As we show in Section 6.5, parallel accounting may also require more communication rounds to converge to the same quality solution with the same privacy guarantee. The gap is more notable on non-identically distributed data.

6.4.3 Joint Privacy

Instance privacy provides tighter and more meaningful guarantees for every data point contribution to the trained model. Nevertheless, there is a downside: adding noise both during the on-device gradient descent and during the averaging phase on the server results in slow convergence or complete divergence of the federated learning algorithm.

To tackle this problem, we propose *joint accounting*, where the noise added on the client side is re-counted towards the client-level privacy guarantee. The main idea of joint accounting is the following. When instance privacy is enforced, the client updates sent to the server are already noisy. Instead of introducing more noise, the server could re-count the added noise to compute the client-level bound. However, without changes in the accounting protocol, the server would not be able to estimate $c_T(\lambda, \xi, D, D')$, as it would no longer have access to non-private client updates distribution.

Fortunately, the inner expectation in $c_T^L(\lambda, \xi, D, D')$ and $c_T^R(\lambda, \xi, D, D')$ can be computed locally, suggesting the following procedure. Every client computes $\hat{D}_{\lambda+1}^{(t)}$ (see Definition 17) with $\hat{p}_t$ and $\hat{q}_t$ being the private outcome distributions with and without their entire update. Then, the server computes $M(t)$, $S(t)$ (also from Definition 17), and $\hat{c}_T(\lambda, \xi, D, D'; \gamma, m)$ by simple averaging. Additionally, one can implement this averaging step with secure multi-party computation to further privacy protection. For the moment, however, it can only be used with `FedSGD`, and not `FedAvg`, because every noisy step in `FedAvg` would change the point at which the gradient is computed, potentially leading to a different gradient distribution or underestimated total noise variance.

Joint accounting allows to achieve tight instance and client privacy guarantees and preserve

the speed of convergence almost at the level of client-only privacy (see Section 6.5.4).

6.5 Evaluation

In this section, we provide results of the experimental evaluation of our approach. We begin by describing the datasets we used, as well as the setting details shared by all experiments. The subsequent structure follows that of the previous section. We first evaluate the client-level privacy by comparing accuracy and privacy guarantees of the traditional DP method (Geyer et al., 2017) to ours (Section 6.5.2). Then, in Section 6.5.3, we perform experiments on the two proposed methods of instance privacy accounting. Finally, Section 6.5.4 describes the results of the joint accounting approach.

6.5.1 Experimental Setting

We perform experiments on two datasets. The first dataset is MNIST (LeCun et al., 1998), which we have been using throughout this thesis and described in detail in Chapter 3. The second dataset is the APTOS 2019 Blindness Detection challenge dataset[1] (in figures, tables and text, we refer to this dataset as *Retina* or *APTOS*). It consists of 3662 retina images taken using fundus photography. The images are labelled by clinicians to reflect the severity of diabetic retinopathy on the scale from 0 to 4. Unlike other datasets commonly evaluated in the privacy literature (Abadi et al., 2016; McMahan et al., 2017; Geyer et al., 2017), this one actually has more serious implications of a privacy leak.

All experiments have the following general setup. There is a server, that coordinates federated training of the shared model, and a number of clients (100, 1000, or 10000), each holding a subset of data. Some setups with a higher number of users will entail repetition of data, like in (Geyer et al., 2017), which can be natural in some scenarios, e.g. shared or very similar images on different smartphones. In MNIST experiments, each user holds 600 examples. For the APTOS dataset, we use data augmentation techniques (e.g. random cropping, resizing, etc.) to obtain a larger training set, and then split it such that every client gets $\sim$350 images. Testing is performed on the official test split for MNIST, and on the first 500 samples in case of APTOS.

We use the following models, and they remain the same in all experiments. For MNIST, we build a simple CNN with two convolutional layers and two fully connected layers (similar to the one described in the TensorFlow tutorial[2]). In case of APTOS, due to the small dataset size and a harder learning task, we employ ResNet-50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) and re-train only the last fully-connected layer of the network. We do not do extensive hyper-parameter tuning in general, since

[1] https://www.kaggle.com/c/aptos2019-blindness-detection/overview/description
[2] https://www.tensorflow.org/tutorials/images/deep_cnn

Table 6.1 – Accuracy and privacy guarantees (reported as a pair (ε, δ)) on MNIST, non-i.i.d. setting.

	Accuracy			Privacy	
Clients	**Baseline**	**DP**	**BDP**	**DP**	**BDP**
100	97%	78%	**88**%	$(8, 10^{-3})$	$\mathbf{(4.0, 10^{-3})}$
1K	98%	95%	**96**%	$(3, 10^{-5})$	$\mathbf{(1.5, 10^{-5})}$
10K	99%	96%	**97**%	$(1, 10^{-6})$	$\mathbf{(0.6, 10^{-6})}$

we are interested in relative performance of private models compared to non-private ones rather than the best classification accuracy, and thus, our non-private baseline results may not match the ones reported in (McMahan et al., 2016) or on Kaggle. For the same reason, we restrict the number of communication rounds (≤ 300) and use `FedSGD` instead of `FedAvg`, although all the methods, except for joint accounting, are compatible with `FedAvg`.

One of the important aspects of federated learning is that data might not be distributed identically among users. In agreement with previous work (McMahan et al., 2016; Geyer et al., 2017), we include experiments in both *i.i.d.* and *non-i.i.d.* settings for MNIST, because it allows for a natural non-identical split. More specifically, in the i.i.d. setting, every user is assigned a subset of uniformly sampled examples. In the non-i.i.d. setting, we follow the same scheme as (McMahan et al., 2016) and (Geyer et al., 2017): splitting the dataset on shards of 300 points within the same class and then assigning 2 random shards to each client. The scenario of 100 clients with non-identically distributed data is particularly hard for privacy applications: there are $\binom{10}{2} = 45$ possible digit combinations that clients can hold and only 100 clients, meaning that some clients might be easily distinguishable by their data distribution. Therefore, it is important to note that it may not be possible to obtain a reasonable privacy bound in this scenario without seriously compromising accuracy.

The privacy accounting is performed by two methods. To obtain the bounds on ε and δ of differential privacy, we use the moments accountant (Abadi et al., 2016). In the case of Bayesian differential privacy, we follow the technique described in Sections 5.4 and 6.4, i.e. sample a number of user updates (or gradients for instance privacy), estimate the upper bound on the privacy cost, and use it to compute the corresponding pair of ε, δ.

6.5.2 Client Privacy

In this experiment, we test adding client privacy the same way it is done by McMahan et al. (2017) and Geyer et al. (2017). We fix the noise level σ and account DP and Bayesian DP in parallel using the moments accountant and Bayesian accountant, as described in Section 6.4.1, accordingly.

Table 6.2 – Accuracy and privacy guarantees (reported as a pair (ε, δ)) on MNIST, i.i.d. setting.

	Accuracy			Privacy	
Clients	**Baseline**	**DP**	**BDP**	**DP**	**BDP**
100	97%	86%	**92**%	$(8, 10^{-3})$	$\mathbf{(2.0, 10^{-3})}$
1K	98%	97%	97%	$(3, 10^{-5})$	$\mathbf{(1.0, 10^{-5})}$
10K	99%	97%	**98**%	$(1, 10^{-6})$	$\mathbf{(0.5, 10^{-6})}$

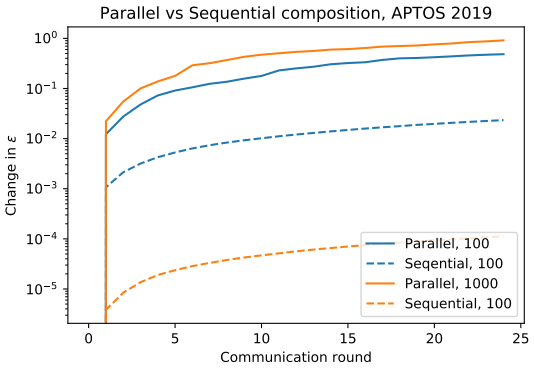
Table 6.3 – Accuracy and privacy guarantees (reported as a pair (ε, δ)) on APTOS 2019, i.i.d. setting.

	Accuracy			Privacy	
Clients	**Baseline**	**DP**	**BDP**	**DP**	**BDP**
100	70%	60%	**65**%	$(8, 10^{-3})$	$\mathbf{(2.1, 10^{-3})}$
1K	71%	67%	**68**%	$(2, 10^{-5})$	$\mathbf{(0.5, 10^{-5})}$
10K	72%	68%	**69**%	$(1, 10^{-6})$	$\mathbf{(0.2, 10^{-6})}$

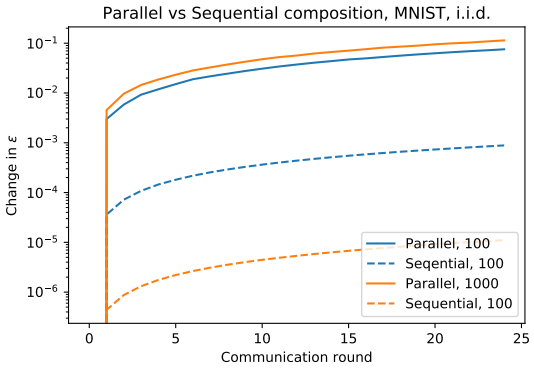
Tables 6.1, 6.2, and 6.3 summarise accuracy and privacy guarantees obtained in this setting for MNIST (non-i.i.d. and i.i.d.) and APTOS respectively. The first column indicates the number of clients; the second, the baseline accuracy of a non-private federated classifier (models described in the previous section). The following columns contain accuracy and privacy parameters obtained for private models using the classical DP and BDP. Despite being trained in parallel, the two techniques may differ in accuracy, because in some cases, we do early stopping for DP to prevent exceeding the privacy budget.

In all cases and for all datasets, we observe substantial benefits of using Bayesian accounting. The accuracy gains are most notable in the non-i.i.d. setting of MNIST, where our method can achieve 10% higher accuracy in the 100 clients setting, because it presents a more difficult learning scenario as explained in the previous section. The privacy gains are consistently significant across all datasets and settings, and taking into account the fact that ε is exponentiated to get the bound on outcome probability ratios, BDP can reach $e^8/e^2 \approx 400$ times stronger guarantee under its assumptions. Nevertheless, in the settings with few clients, even Bayesian differential privacy does not reach a more comfortable guarantee of $\varepsilon = 1$, suggesting that a better privacy-accuracy trade-off may not be feasible due to higher clients identifiability, or that more work is needed in improving training with noise and developing novel privacy mechanisms for federated learning.

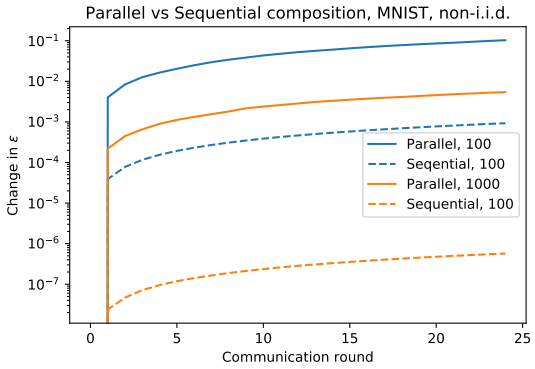
Importantly, there is no computation or communication overhead from the users' point of view in these experiments since the privacy accounting code is executed on the server.



(a) APTOS 2019



(b) MNIST, i.i.d.



(c) MNIST, non-i.i.d.

Figure 6.1 – Change in ε relative to its initial value for parallel and sequential composition modes of instance privacy in the settings of 100 and 1000 clients.

6.5.3 Instance Privacy

As noted in Section 6.4.2, instance privacy is very important in scenarios like training on medical data from a number of hospitals where patient privacy is at least as crucial as hospital privacy. In this section, we compare two accounting methods proposed earlier: *sequential* and *parallel* accounting.

Depicted in Figure 6.1 are the curves showing the growth of ε estimate with communication rounds. We subtracted the initial value and applied logarithmic scale in order to better show the difference in the rate of growth. Across all settings, it can be seen that parallel accounting leads to faster growth rates, despite the fact that the parallel composition is more efficient (taking maximum over clients instead of a sum). This behaviour can be explained by the fact that each client is unaware of the total dataset size and, having a small number of data points, is convinced that every data example has significant influence on the outcome. The unawareness about other clients in the case of parallel accounting can also explain the fact that we don't observe any improvement in the ε growth rate with increasing number of clients. The only exception is the non-i.i.d.

Table 6.4 – Accuracy and privacy guarantees (a pair (ε, δ)), at instance and client levels, using joint privacy accounting in the setting of 100 clients.

	Accuracy			Privacy	
Dataset	**Baseline**	**DP**	**BDP**	**Client**	**Instance**
APTOS 2019	70%	42%	**64**%	$(1, 10^{-3})$	
MNIST (iid)	97%	15%	**74**%	$(2, 10^{-3})$	$(0.1, 10^{-5})$
MNIST (non-iid)	97%	12%	**62**%	$(4, 10^{-3})$	

MNIST experiment, where the difference likely comes from increasing stability of training and decreasing gradient variability with more clients.

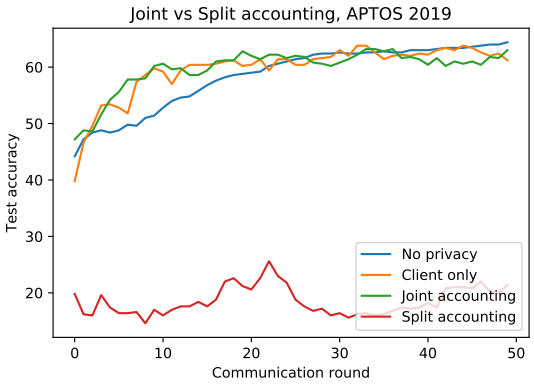
The main takeaway from this experiment is that it appears to be beneficial to use sequential accounting for privacy of the federated model whenever communicating the total size of the dataset to users is acceptable. In other cases, and for personal privacy accounting in case of the untrusted curator, parallel accounting can be used, but more noise is needed for reasonable privacy guarantees and further investigation is necessary to protect the client with the maximum privacy cost.

6.5.4 Joint Privacy

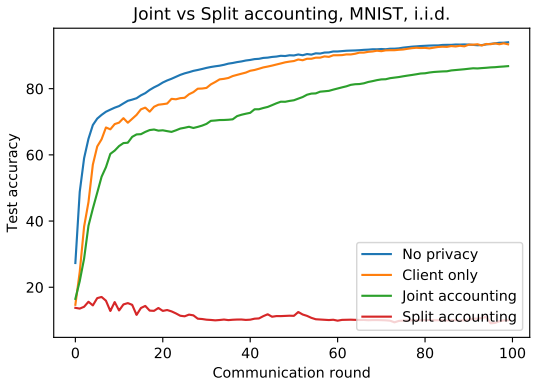
Lastly, we would like to test the proposed method of joint accounting for instance-level and client-level privacy, and contrast it with accounting at these two levels separately. We perform experiments in the same settings as above, fixing the client privacy at a certain level ($\varepsilon = 1$) and evaluating the speed and quality of training. We also compare to what can be achieved by introducing privacy only at the client level.

Figure 6.2 displays the test accuracy evolution over communication rounds in the setup of 100 clients for APTOS and 1000 clients for MNIST. The graphs contain curves for training without privacy, client-level-only privacy, and the two accounting paradigms: *joint* and *split*. As expected, the non-private training achieves the best accuracy. Nevertheless, in the i.i.d. setting, client-only private training quickly approaches non-private training in quality. Notably, training with both instance and client privacy using the joint accounting performs nearly as well, while training with the separate accounting completely fails due to excessive amounts of noise at both instance and client levels. For the non-i.i.d. setting, private training is slower, but there is little difference between introducing privacy only at the client level and using the joint accounting at both levels: after a slightly larger number of rounds, training with the joint accounting reaches similar performance. Based on these experiments, we conclude that by using joint accounting we can introduce instance privacy on clients and get client-level privacy at almost no cost.

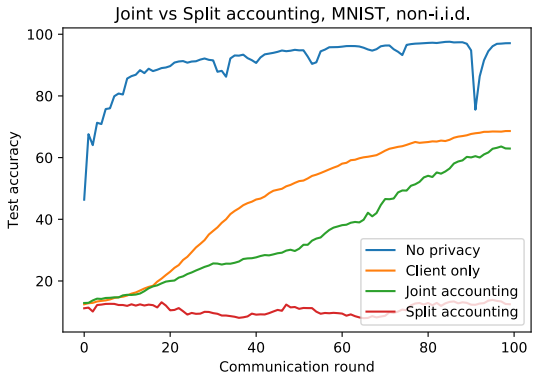
Finally, we evaluate our method in the strong privacy setting. We set the instance-level



(a) APTOS 2019, 100 clients.



(b) MNIST, i.i.d., 1000 clients.



(c) MNIST, non-i.i.d., 1000 clients.

Figure 6.2 – Test accuracy as a function of a communication round for non-private, client-level-only private, and jointly private (using either joint or separate accounting) scenarios.

privacy to $\varepsilon = 0.1$ and stop training when the client privacy reaches the level similar to previous experiments (except APTOS dataset, where we were able to achieve comparable results with lower privacy cost), and report the accuracy that can be achieved in this strict setting. We have also chosen the most difficult scenario of 100 clients. As seen in Table 6.4, the algorithm with differential privacy performs very poorly on APTOS dataset, and fails to learn on MNIST, in both i.i.d. and non-i.i.d. setting. Its performance is especially affected by the strict instance privacy requirement, since such low levels of ε necessitate large quantities of noise to be added. It is worth noting, that it might be possible to achieve better results with DP by performing per-example gradient clipping, as in (Abadi et al., 2016), but we do not use this technique due to its impracticality.

On the other hand, our approach manages to achieve reasonable accuracy even under such a strict privacy budget. On APTOS dataset, it is just 6% lower than the non-private baseline, while on MNIST, it correctly classifies more than 70% of the test data in the i.i.d. setting and over 60% in the non-i.i.d. setting. One could potentially add more

noise on the server and combine the accounting with the instance level noise to slow down the growth of ε and reach even better performance, but we leave these experiments for future work.

Both instance and joint privacy accounting add some computation overhead on user devices due to multiple gradient calculations. However, performing FL routines when devices are idle and charging, as suggested in (Bonawitz et al., 2019), alleviates this problem. Communication overhead is negligible because only a single floating point number is added to user messages.

6.6 Conclusion

We employed the notion of Bayesian differential privacy, a relaxation of (ε, δ)-differential privacy, to obtain tighter privacy guarantees for clients in the federated learning settings. Similarly to the centralised setting, the idea of this approach is to utilise the fact that users come from a certain population with similarly distributed data, and therefore, their updates will likely be in agreement with each other.

We adapt an efficient and tight privacy accounting method for Bayesian differential privacy to the federated setting in order to estimate client privacy guarantees. Moreover, we emphasise the importance of instance-level privacy and propose two variants of privacy accounting at this level. Finally, we introduce a novel technique of joint accounting suitable for obtaining privacy guarantees at instance and client levels jointly from only instance-level noise.

Our evaluation provides evidence that Bayesian differential privacy is more appropriate for federated learning. It allows models to train in fewer communication rounds and achieve higher accuracy by using significantly less noise, compared to DP, to reach strong privacy protection. When the number of clients reaches an order of thousands, which is realistic in many federated learning scenarios, ε_μ of BDP can be kept below 1. Finally, we demonstrate that by using joint accounting we can get client privacy "for free" when adding instance privacy. This way, the privacy budget can be kept close to $\varepsilon_\mu = 1$ for client privacy and $\varepsilon_\mu = 0.1$ for instance privacy while maintaining reasonably high accuracy.

7 Generating Data with Bayesian Differential Privacy

7.1 Introduction

With machine learning (ML) becoming ubiquitous in many aspects of our society, questions of its privacy and security take centre stage. A growing field of research in privacy attacks on ML (Fredrikson et al., 2015; Shokri et al., 2017; Hitaj et al., 2017; Truex et al., 2018) tells us that it is possible to infer information about training data even in a black-box setting, without access to model parameters. However, this primarily remains a matter of interest of the research community. A wider population, on the other hand, is concerned with privacy practices used in the ML development cycle, such as company employees or contractors manually inspecting and annotating user data[1,2].

Conversely, using federated learning (FL), differential privacy (DP) and other privacy practices creates an additional hurdle for developers, as they cannot inspect data, especially in decentralised settings, making it difficult to understand the model behaviour and find bugs in data and implementations. To the best of our knowledge, Augenstein et al. (2019) were the first to formulate these questions, provide a more complete characterisation and propose a solution similar in spirit to our `FedGP` framework.

This chapter circles back to that idea of adopting generative adversarial networks (GAN) trained in a privacy-preserving manner, a concept first introduced in Chapter 3, for addressing these issues. Augenstein et al. (2019) use the conventional DP notion and strongly rely on the user population sizes of millions to provide acceptable guarantees and data quality. In contrast, we use the notion of Bayesian differential privacy, enabling significantly more practical privacy guarantees for in-distribution samples. More details on the overall approach and privacy are provided in Section 7.4.

This chapter is based on the paper available as a preprint (Triastcyn and Faltings, 2020c).

[1] https://www.theguardian.com/technology/2020/jan/10/skype-audio-graded-by-workers-in-china-with-no-security-measures

[2] https://www.bloomberg.com/news/articles/2019-04-10/is-anyone-listening-to-you-on-alexa-a-global-team-reviews-audio

An important advantage of using our privacy definition is that it enables generating data of higher fidelity (i.e. visual quality) compared to previous work on GANs with DP, allowing for finer-grained inspection of data. While some problems with data or data pipelines can be discovered using very coarse samples (e.g. pixel intensity inversion in (Augenstein et al., 2019)), more subtle bugs, like partial data corruption, would require samples of much better quality, rendering the DP guarantee too loose to be meaningful. Moreover, if fidelity is high enough, synthetic data can be used for annotation and training itself, removing the related privacy concerns and extending applicability of FL. We evaluate our solution in these two aspects in Section 7.5.

On the other hand, Bayesian DP is also superior to average-case DP, considered in our private data release solutions in Part I. Bayesian DP provides a stricter theoretical guarantee, closer in strength and generality to the traditional DP. Therefore, protecting generative models with BDP is preferable, if the generated sample quality is not greatly affected.

Our main contributions in this chapter are as follows:

- we use Bayesian DP to enable higher quality GAN samples, while still providing a strong privacy guarantee;
- we demonstrate that this technique can be used to discover finer data errors than previously reported;
- we also show that for some tasks synthetic data are of high enough quality to be used for labelling and training.

7.2 Related Work

Up until recently, the aspect of human involvement in the development cycle and manual data processing has been largely overlooked in privacy-preserving machine learning. These issues can be mitigated, at least partially, by federated learning (FL) (McMahan et al., 2016), which brings a great promise for user privacy. Yet, FL paradigm creates additional problems of its own. Augenstein et al. (2019) provide a good starting point, systematising these problems and proposing a solution by the use of synthetic data. Their solution is based on generating synthetic datasets using federated differentilly private GANs. Privacy-preserving data synthesis using GANs, including with differential privacy, has been introduced in earlier works (Beaulieu-Jones et al., 2017; Xie et al., 2018; Zhang et al., 2018; Triastcyn and Faltings, 2019a; Jordon et al., 2018; Long et al., 2019), but these papers mainly focused on achieving high utility of synthetic data without addressing a broader scope of privacy leakage via manual data handling. A major contribution of Augenstein et al. (2019) is the taxonomy of common ML modeller tasks and the extension of the GAN-based paradigm to solving these tasks. Apart from that, this topic has

not been extensively studied in the literature, and in addition to what we have already covered in this thesis, we could only add the survey by Humbatova et al. (2019) on faults in deep learning systems, also highlighted by Augenstein et al. (2019).

A common problem of differentially private GANs, however, is that the generated samples have very low fidelity, unless the privacy guarantee is unreasonably weak. For example, observe the quality of image samples in Augenstein et al. (2019, Figure 3). While the quality is sufficient to detect the change of background colours, individual symbols are not distinguishable, although symbols in itself should not reveal any private information unless the handwriting style is preserved. Further, consider Augenstein et al. (2019, Table 3). The authors show that privacy guarantees are strong in scenarios of millions of users, but in the simulations with hundreds or thousands of users the values of ε and δ are extremely high, rending the guarantees meaningless.

Our approach makes progress in exactly this perspective: we can achieve much higher quality outputs with little compromise in privacy guarantees (and only for outliers that are difficult to hide). As a result, our synthetic data yield better performance of downstream analytics and provide more powerful data inspection capabilities.

7.3 Preliminaries

This chapter largely relies on the notions and methods introduced earlier in this thesis. In particular, we use generative adversarial networks as the means to create privacy-preserving synthetic datasets, the approach introduced and described in detail in Chapter 3. Although we do not run experiments in the federated learning setting, due to computational intensiveness, Chapter 4 could provide a roadmap for mapping our solutions from this chapter to federated scenarios. Instead of average-case DP, initially used with our GAN-based data release, we employ Bayesian differential privacy, defined in Chapter 5 and extended to federated learning in Chapter 6.

Additionally, in parts of the chapter, we refer to Augenstein et al. (2019) classification of ML developer tasks, which can be condensed to:

T1 Sanity checking data.
T2 Debugging mistakes.
T3 Debugging unknown labels / classes.
T4 Debugging poor performance on certain classes / slices / users.
T5 Human labelling of examples.
T6 Detecting bias in the training data.

We refer the reader to Augenstein et al. (2019, Section 2) for further details.

Augenstein et al. (2019) were primarily interested in the tasks T3 and T4, testing the

ability of generative models to produce private samples of sufficient quality to detect out-of-vocabulary words or to investigate a drop in the model performance for a sub-population of users. We compare Bayesian DP GAN with their solution in the data inspection experiment in Section 7.5.2. However, our other focus is the task T5, as we believe it is under-represented in the private ML literature and is important to the broader society.

In short, the task T5 arises when the data used to train the ML model are not labelled, annotated, or cannot be self-annotated. An example of self-annotating the data is the next word prediction when typing on a smartphone keyboard. In situations, where this is not possible, data are typically annotated by the employees of the company or contractors, because this task if often too burdensome for users. It is also possible that users would not be qualified to label their own data, for example, in the case of a medical application.

7.4 Our Approach

We employ the same approach as in Chapters 3 and 4 of this thesis, and as in (Augenstein et al., 2019). The primary distinction is using Bayesian differential privacy instead of average-case DP or the traditional DP. The fact that BDP takes into account the data distribution and assumes that all data points are drawn from that same distribution, although the distributions can be multimodal, highly complex, and generally unknown, maps well to the context of generative models, especially GANs. The task of generative modelling in itself is to learn the underlying data distribution, and thus, a common distribution is an implicit belief. This results in an organic match with BDP, because there are no assumptions to add to the problem.

7.4.1 Formal Overview

We are given a dataset D of labelled $(\{(x_i, y_i) \mid (x_i, y_i) \sim \mu(x, y),\ i = 1..n\})$ or unlabelled $(\{x_i \mid x_i \sim \mu(x),\ i = 1..n\})$ examples. This dataset can be decentralised, in which case we would use FL (see the next subsection). Our task is to train a GAN, which consists of the generator $\mathcal{G}$ and the critic $\mathcal{C}$ (discriminator), to generate synthetic samples from μ.

Our privacy mechanism follows the previous work on differentially private GANs (Beaulieu-Jones et al., 2017; Xie et al., 2018). More specifically, it applies Gaussian mechanism to the discriminator updates at each training step. That is, we clip the updates to norm C and add Gaussian noise with variance $C^2\sigma^2$. Privacy of the generator is then guaranteed by the post-processing property of BDP. It is worth mentioning, however, that clipping and/or adding noise to the generator gradients can be beneficial for training in some cases, to keep a better balance in the game between the critic and the generator, and it

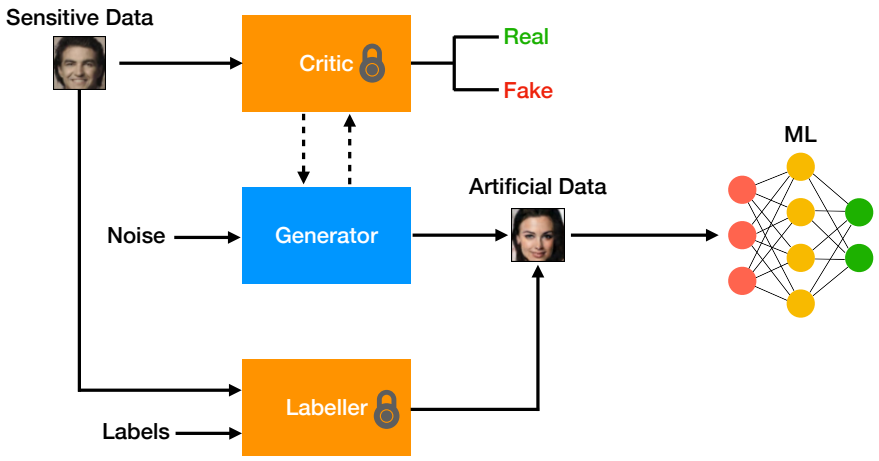


Figure 7.1 – Architecture of our solution using an unconditional GAN and a separate annotator with Bayesian DP. The lock icon indicates the models trained with the noisy gradient descent.

should not be overlooked by developers.

We choose not to implement more complicated schemes, such as PATE-GAN (Jordon et al., 2018) or G-PATE (Long et al., 2019), which use PATE framework (Papernot et al., 2018) to guarantee differential privacy for GANs. Our key rationale is that a more complicated structure of these solutions could create unnecessary errors and additional privacy leakage (e.g. leaking privacy by backpropagating through the teachers' votes to the generator, thereby neglecting the added noise). Nevertheless, we show in our evaluation that due to the distribution-calibrated BDP accounting (and hence, less added noise) our GAN generates better quality samples compared to these more complex solutions.

There is another notable difference between our current technique and prior work, as well as our approach in Chapters 3 and 4. Instead of training conditional GANs and generating labelled exampled directly, we train unconditional GANs along with a separate private classifier. We then label GAN samples with this classifier. This tweak allows for a closer imitation of external labelling of data examples. Besides, we found that this approach helps mitigate the problem of noisy gradient descent for GANs to a certain extent in our experiments. Although, we do not have sufficient observations to claim that it is universally better. Figure 7.1 presents a birds-eye view of the technique.

Remark 10. Despite the substantially lower amount of noise needed for meaningful BDP guarantees, training GANs with it still presents a serious challenge. We believe that the reason lies in the subtle balance between the two networks: disproportionately hindering training of one of them causes the other to converge to a poor local optimum. At best, it could make hyper-parameter search more difficult. At worst, render the

approach impractical for some applications. Possible remedies include rolling back to a weaker ADP-based solution (Chapter 3) or pre-training on public data.

7.4.2 Additional Privacy Control

Our solution has an additional lever to control privacy leakage, which would not be possible neither with ADP nor with the conventional DP. It is based on the intuition that the foremost source of privacy leakage are outliers. On the one hand, outliers' privacy loss is discounted in Bayesian accounting due to their low probability. On the other hand, we can reduce the number of samples generated by the GAN to decrease the chances of these outliers appearing in the synthetic dataset.

Remember the interpretation of BDP guarantees we proposed in Section 5.4.6. More specifically, we considered the following random variable:

$$\Delta(\varepsilon, x') = \Pr\left[L(w, D, D') > \varepsilon \mid D, D' = D \cup \{x'\}\right].$$

For DP, the moments accountant outputs δ, an upper bound on $\Delta(\varepsilon, x)$:

$$\max_x \Delta(\varepsilon, x) \leq \delta.$$

Bayesian DP, however, can be shown to produce the expectation upper bound:

$$\mathbb{E}_x\left[\Delta(\varepsilon, x)\right] \leq \delta_\mu.$$

Having the expectation bound, one could compute privacy guarantees for different percentiles of the data distribution and determine the probability of a "privacy outlier" being drawn. This can be done using Markov's inequality:

$$\Pr[\Delta(\varepsilon, x) \geq \phi] \leq \frac{\mathbb{E}_x[\Delta(\varepsilon, x)]}{\phi} \leq \frac{\delta_\mu}{\phi}.$$

Say we achieved $(1, 10^{-8})$-BDP. Then we can calculate that the probability of drawing a data sample x', for which $\Delta(\varepsilon, x') \geq 10^{-5}$, is equal to $\frac{\delta_\mu}{\phi} = \frac{10^{-8}}{10^{-5}} = 0.001$. Or, alternatively, 99.9% of the data distribution upholds $(1, 10^{-5})$-DP guarantee.

If the generative model has properly converged to the underlying distribution, one could limit the size of the generated set to control the probability of a "privacy outlier" appearing in this set. For the example above, if we create a synthetic set of size 1000, the chance that at least a single outlier would appear in it is $1 - 0.999^{1000} \approx 0.632$. However, if the size is only 100, this chance is $1 - 0.999^{100} \approx 0.095$.

In Section 7.5.3, we show that learning from artificial data samples saturates quicker than

for real ones, meaning that fewer synthetic examples need to be generated. And given our results in Chapter 5, going from $\delta_\mu = 10^{-5}$ to $\delta_\mu = 10^{-10}$ does not significantly degrade ε_μ, suggesting that the additional privacy control described above could help create entire synthetic datasets with strong guarantees. Nonetheless, it is worth noting that this would be a heuristic guarantee, as it relies on the assumption that the generative model is closely approximating the data distribution. An interesting future direction would be to analyse this guarantee based on some measure of the generative model convergence to the true distribution.

7.4.3 Federated Learning Case

It is worth mentioning that we did not make any assumptions on where the data are located. A logical scenario to consider would be federated learning, like in (Augenstein et al., 2019), such that the data remain on user devices at all times.

To accommodate FL scenarios, minimal modifications to the approach are required. Training of the generative model would be performed in the same way as any other federated model, and privacy accounting would be done at the user-level (Augenstein et al., 2019). Alternatively, we could modify the federated learning protocol and keep critics private to each client, following `FedGP` approach (see Chapter 4). This is especially beneficial in cases when each client has sufficient data to train a good discriminator. For example, it could be applied when there is a hierarchical structure and each client is not an individual user but an entity, such as a hospital or a bank.

Baysian DP analysis is also directly transferable to FL (see Chapter 6), and privacy bounds are generally even tighter in this case. Moreover, given the difficulty of training GANs with gradient noise, discussed throughout this thesis, as well as in the next section, federated learning with BDP and on-device discriminators trained without noise could significantly boost the overall performance. Hence, it is an especially promising direction for future work.

7.5 Evaluation

We evaluate two major applications of the technique. First, we show that the generated samples can be used for debugging ML model through data inspection, resembling tasks T1–T4 from (Augenstein et al., 2019). Second, we examine the quality of the downstream ML model trained directly on synthetic samples, thus demonstrating a possibility of solving T5 (data labelling/annotation) as well.

In the debugging experiment, we attempt to detect a more subtle bug compared to (Augenstein et al., 2019): an incorrect image rotation that yields lower model performance. While the pixel intensity inversion can be easily spotted using low-fidelity synthetic

samples, image rotation requires higher fidelity to be detected.

7.5.1 Experimental Setting

We use two image datasets, MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017). Both have 60000 training and 10000 test examples, where each example is a 28×28 size greyscale image. The task of Fashion-MNIST is the clothes type recognition. Although these datasets may not be of particular interest from the privacy viewpoint, this choice is directed by the ability to compare to prior work.

For the generative model, we experimented with variations of Wasserstein GAN (Arjovsky et al., 2017) and WGAN-GP (Gulrajani et al., 2017), but found the former to produce better results, probably because gradient clipping is already a part of the privacy mechanism. Our critic consists of three convolutional layers with SELU activations (Klambauer et al., 2017) followed by a fully connected linear layer with another SELU and then a linear classifier. The generator starts with a fully connected linear layer that transforms noise (and possibly labels) into a 4096-dimensional feature vector which is then passed through a SELU activation and three deconvolution layers with SELU activations. The output of the third deconvolution layer is down-sampled by max pooling and normalised with a `tanh` activation function.

All models are trained using Adam with the learning rate 0.0001. The clipping threshold for gradients of discriminators is set to 0.5, no clipping is done for generators, and the noise standard deviation is 0.01 for MNIST and 0.02 for Fashion-MNIST. The reported accuracy and privacy were achieved with 400 epochs for both datasets.

Although we use the centralised setting throughout this section, the results are readily transferable to federated scenarios. Our previous work suggests that neither the GAN sample quality (Chapter 4) nor BDP guarantees (Chapter 6) should be significantly affected.

7.5.2 Data Inspection

The data inspection experiment is setup in the following way. We introduce the rotation bug through randomly rotating some images by 90°. We then train the two generative models, on correct images and on altered images, and visually compare their samples. We also train a model with DP to show that its image quality would not be sufficient to detect the error. In the process of experimentation, we explored other possible bugs and corruption schemes, such as downsampling images (reducing its resolution) or randomly removing parts of images. Although Bayesian DP ensures higher image quality in all these cases, we found that the rotation example demonstrates its advantages more clearly.

(a) Real. (b) Synthetic.

Figure 7.2 – Real and synthetic samples on Fashion-MNIST.

(a) Trained on correct images with BDP. (b) Trained on altered images with BDP. (c) Trained on altered images with DP.

Figure 7.3 – GAN output for detecting unwanted rotations on MNIST.

Figure 7.3 shows the output of generative models trained on MNIST with and without image rotation. By examining the samples, developers can clearly determine that a portion of images was rotated. This way, the error can be promptly identified and fixed. On the other hand, with generative models that uphold the traditional DP guarantee (Figure 7.3c), it would be difficult to detect such a pre-processing error, as the produced samples have very low fidelity. Also, ε in this case is unjustifiably high at the order of 10^7, which is consistent with the results of Augenstein et al. (2019). On the other hand, Bayesian DP results are achieved with a strong privacy guarantee: under $(1, 10^{-10})$-BDP.

We also observe that the synthetic data quality under BDP (see Figures 7.2 and 7.3a) might be sufficient to detect previously unseen classes or dataset biases, such as under-represented classes. Naturally, biases can also be introduced by GANs itself, and it is important to be able to distinguish these two cases. This is another attractive direction for future research on the topic.

7.5.3 Learning Performance

Now, we evaluate the generalisation ability of the student model trained on artificial data. The experiments are set up in the same way as in Chapters 3 and 4. More specifically, we train a student model on generated data and report test classification accuracy on a real held-out set.

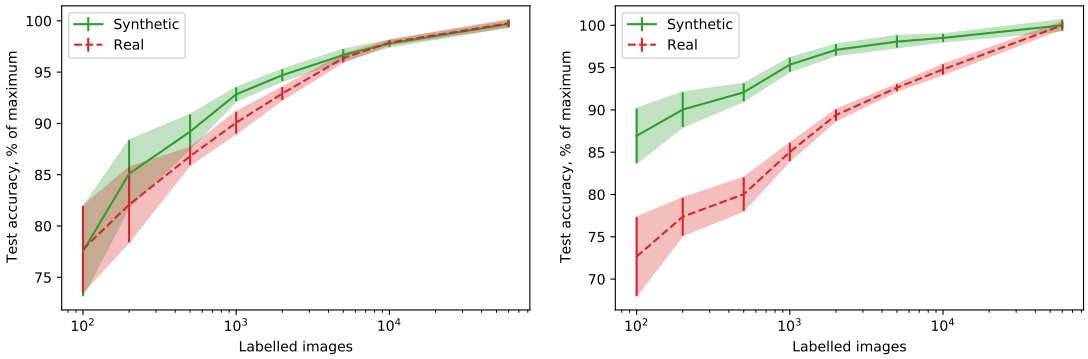
The goal of this experiment is to show that, having a privacy-preserving generative model, we can use synthetic samples to fully replace the real data. Not only it allows to eliminate manual labelling of real (and potentially sensitive) data, but also expand the set of problems that can be solved by FL (task T5 in Augenstein et al. (2019) classification). For example, some medical data cannot be automatically annotated, and users are not qualified to do that, so high-quality synthetic data would allow the annotation to be performed by doctors without privacy risks for users.

We imitate human annotation by training a separate classifier (with the same privacy guarantee as the generative model) and using it to label synthetic images. While this approach is different from prior work on generating data for training ML models, which used conditional GANs, comparisons in this section are still valid because our annotator maintains the same privacy guarantee. Figure 7.1 depicts this aspect of our technique.

We choose to compare with the method called G-PATE (Long et al., 2019), because it is one of the best recent techniques in terms of privacy-utility trade-off. As the name suggests, it uses PATE framework (Papernot et al., 2016) to train GANs with differential privacy. This framework enables more efficient DP training by relying on the idea that the privacy loss should be small when multiple independent models strongly agree on the outcome. Long et al. (2019) showed that their method outperforms another PATE-based approach, PATE-GAN (Jordon et al., 2018), as well as DP-GAN (Xie et al., 2018), based

Table 7.1 – Accuracy of models: (1) non-private baseline (convolutional network); (2) private classifier (convolutional network trained with BDP); and student models: (3) for G-PATE with $(1, 10^{-5})$-DP guarantee; (4) for WGAN with $(1, 10^{-10})$-BDP guarantee (our method).

Dataset	Non-private	Private classifier	G-PATE	Our approach
MNIST	99.20%	95.59%	56.31%	93.64%
Fashion-MNIST	91.51%	82.20%	51.74%	76.83%



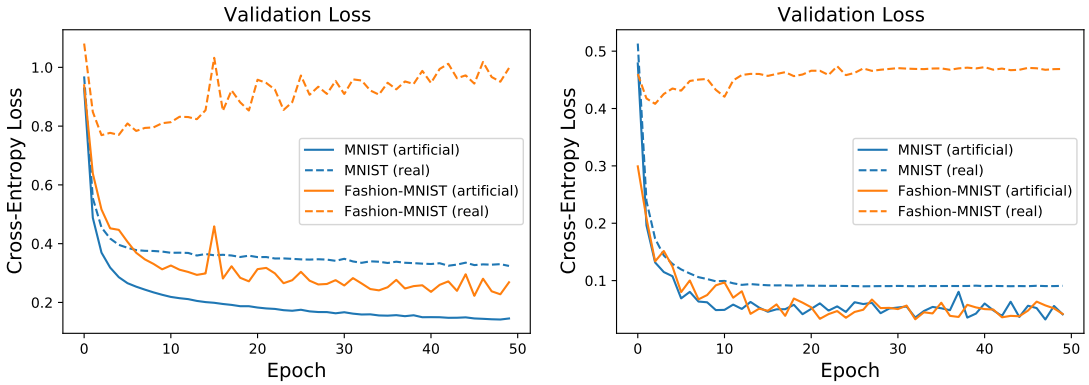
(a) MNIST. (b) Fashion-MNIST.

Figure 7.4 – Relative accuracy (a percentage of maximum achievable accuracy) for different numbers of labelled images.

on DP-SGD. A direct comparison with the latter would be more fair than with G-PATE, but the quality of samples with the comparable privacy guarantee would not be sufficient for learning.

Student model accuracy is shown in Table 7.1. Apart from G-PATE, we compare our method to a non-private classifier trained directly on the real dataset, and a private classifier, trained on the real dataset with Bayesian DP. In the case of generative models, the same (non-private) classifier is trained on the private synthetic output. All results in the table are obtained with the privacy guarantee of $(1, 10^{-5})$-DP, or $(1, 10^{-10})$-BDP, which is equivalent to $(1, 10^{-5})$-DP for this data with high probability. Although Long et al. (2019) report better results for $(10, 10^{-5})$-DP, we do not include those in the study, because $\varepsilon = 10$ is too high for providing meaningful guarantee (see Section 5.3.3).

Generally, we observe that switching from real to synthetic data with privacy protection does not seriously reduce accuracy of the student models on these datasets. On MNIST, the drop in performance between a private discriminative and a private generative approach is less than 2% (from 95.6% to 93.6). It is more noticeable on Fashion-MNIST (from 82.2% to 76.8%), but is still within about 7% and is still lower than the drop between a non-private and a private classifiers. At the same time, BDP GANs significantly



(a) Bayesian DP.

(b) Average-case DP.

Figure 7.5 – Cross-entropy loss for real and artificial validation sets.

outperform G-PATE on both datasets. The DP-based approach only reaches $\sim 56\%$ and $\sim 52\%$ accuracy on MNIST and Fashion-MNIST correspondingly. Even considering the results with a looser privacy guarantee of $\varepsilon = 10$ from Long et al. (2019, Table 2), our solution has the edge.

In another experiment, Figures 7.4a and 7.4b show a percentage of maximum accuracy (i.e. numbers reported in Table 7.1) achievable by the model when only a part of the generated data is labelled. Notably, the models trained on synthetic data achieve the same percentage faster than the ones trained on the real data, which is especially evident for Fashion-MNIST dataset. Moreover, as little as 100 labelled samples is enough to outperform models trained on data generated with comparable DP guarantees. It is worth mentioning that these graphs do not mean that a smaller number of labelled synthetic samples is better for learning than the real examples. Nevertheless, it allows to make a more educated decision about the trade-offs. For instance, 100 labelled synthetic Fashion-MNIST samples produce the same quality model as 100 real samples. Consequently, if one only had resources to label 100 examples, the loss of utility would not be a part of consideration.

Non-private synthetic data (not shown in the table) allow to reach somewhat better results compared to private synthetic data. On MNIST, the student model reaches 96.09%, while on Fashion-MNIST, 84.86%. It suggests that about half of the accuracy loss comes from the limited capacity of the generative model. Figures 7.4a and 7.4b seem to corroborate this finding, as the learning curve for synthetic data saturates quicker. This kind of performance would be reached by our approach taken in Chapter 3, which essentially trains a GAN without additional noise and uses average-case DP to estimate expected privacy loss. However, one should be careful about comparing these two directly, because the Bayesian DP guarantee is substantially stronger than ADP. We leave a more detailed discussion on how ADP and BDP compare for Section 8.2.

Finally, we consider another important aspect of learning—model validation. We perform an experiment on using synthetic data for validation, as we did in Section 3.6.4. Figure 7.5 shows cross-entropy loss curves on real and artificial validation sets. We compare the curves produced by GANs with Bayesian DP and with average-case DP (i.e. trained without gradient noise). On MNIST, we see a similar picture for both privacy notions, with correlation coefficients between real and artificial validation scores being high: 0.987 for BDP and 0.981 for ADP. On Fashion-MNIST, however, both ADP- and BDP-based solutions fail to reflect the true direction of the validation curve, despite capturing some of the finer scale details (at least in case of BDP). Correlation coefficients for Fashion-MNIST are negative, -0.058 and -0.581. It suggest that, despite initially optimistic results, the validation aspect warrants further research. Potentially, one could relate such behaviour to faster saturation of the learning curve for synthetic Fashion-MNIST.

7.6 Conclusion

We explored the use of generative adversarial networks to tackle the problem of privacy-preserving data inspection and annotation in machine learning. While the previous approaches to this problem, including ours in Chapters 3 and 4, involve generative models either without a rigorous privacy guarantee or with differential privacy, we opt for Bayesian differential privacy, introduced in Chapter 5. By capturing the inherent properties of data and allowing for non-uniform privacy loss throughout the dataset, it enables higher-quality synthetic data while still maintaining a strong privacy guarantee, comparable to DP under mild conditions.

We perform data inspection evaluation by introducing image corruption before training the generative model. Our experiments show that privacy-preserving GANs with BDP can be used to detect these bugs, in the data itself or pre-processing pipelines. To the contrary, this particular mistake would be difficult to catch using DP GANs due to low samples fidelity. Similarly, we presume that biases in the data and previously unseen classes can be discovered in a similar way.

In addition, the generated data can be directly annotated and used for training in place of the real data. We demonstrate it by training student models on our synthetic samples labelled by a separate privacy-preserving classifier, imitating a human annotator. These student models are shown to achieve significantly higher accuracy compared to prior state-of-the-art, represented by a DP GAN trained with PATE, and exhibit only a mild drop in performance compared to private classification with real data. Furthermore, this gap is determined by the quality of the generative model to a large extent, and hence, will get smaller with advances in that field.

8 Conclusion

8.1 Summary

Machine learning and data analytics gradually take over the world, becoming commonplace in many aspects of our lives, from smartphones to hospital rooms. This is not surprising, given the potential benefits that these methods bring. At the same time, as society becomes more aware and concerned with privacy protection, security and privacy issues in machine learning take centre stage. Even if the training data itself remain secure, it has been recently shown that certain attacks can infer information about these data from the trained models, in both white-box and black-box settings (Fredrikson et al., 2015; Shokri et al., 2017).

On the other hand, providing meaningful privacy guarantees in machine learning remains an open question. Existing methods typically fall into one of the two categories: the ones that value privacy over accuracy, and the ones that value accuracy over privacy. The first provide strong, formal privacy guarantees, but often have significantly lower learning performance. The second achieve the learning performance comparable to non-private models, but offer very weak and limited privacy protection. This trade-off between privacy and utility is manifested even stronger in privacy-preserving data release, where the task is to publish or provide (almost) unrestricted access to a dataset containing sensitive information.

In this thesis, we explore ways to improve the privacy-utility trade-off in machine learning applications. As a result, we achieve significantly better performance for private models while still maintaining theoretical privacy guarantees, and enable private data release for complex real-world datasets.

First, we propose a method based on generative adversarial networks (GANs) to produce artificial datasets that can be used in place of the original data. This is a promising technique to solve the problem of privacy-preserving data release. The choice of GANs

as a generative model is motivated by its scalability and it improves the technique applicability to real-world data with complex structure. Unlike many previous methods, ours uses a customised data-aware privacy definition and is able to generate higher quality data while providing an empirical privacy guarantee. Relaxing the privacy notion also allows to avoid unrealistic assumptions, such as access to similar publicly available data, necessary for many DP methods to produce usable results. In our experiments, we demonstrate that student models, trained with artificial data, reach high classification accuracy on MNIST and SVHN. On these datasets, models can also be validated on artificial data, and the validation scores correlate with those obtained from real validation sets. Although, this result should be taken with a grain of salt, as we explain in Chapter 7. At the same time, we show that training on artificial data makes classifiers more resistant to model inversion attacks, for example, reducing the face detection rate in reconstructed face images from 63.6% to 1.3%, and the face recognition rate from 2.8% to 0.1%.

Second, we develop an empirical notion of privacy, (μ, γ)-*Average-Case Differential Privacy (ADP)*, oriented towards private synthetic data release. This notion relaxes the requirements of DP by considering an average-case scenario instead of the worst-case. Moreover, due to complexity of data distributions, we substitute the rigorous theoretical bound with an empirical bound based on sampling from the dataset. These relaxations allow to avoid adding excessive amounts of noise, like it is done in DP, and enable *ex post* privacy analysis of generative models. Although ADP does not offer a formal guarantee, its empirical bound is useful for evaluating typical privacy risks of generative models, especially when achieving DP guarantees is not feasible or when the model has already been deployed.

Third, we design a new variant of differential privacy, called $(\varepsilon_\mu, \delta_\mu)$-*Bayesian Differential Privacy (BDP)*, tailored to provide guarantees for sensitive data that are all drawn from the same data distribution $\mu(x)$. The data distribution can be arbitrary and is generally unknown. Such a relaxation is sensible in many machine learning applications, since models and algorithms are typically trained on particular data distributions (e.g. emails, face images, MRIs, and so on). We formulate and prove basic properties of BDP, as well as the advanced composition theorem that enables tight and efficient privacy accounting. We employ the Bayesian parameter estimation approach, along with the flat prior and the maximum entropy principle to address the fact that the data distribution is unknown and the sample size is finite. It allows us to overestimate privacy loss, under mild assumptions on its prior, with high, controllable probability and avoid underestimating probabilities of unseen examples. Our experiments confirm that Bayesian DP is highly advantageous in ML scenarios where the additional data distribution assumptions are organically satisfied. Bayesian DP requires less noise than DP for comparable privacy guarantees, and hence, training converges faster and models achieve higher accuracy. Most importantly, BDP is best used in combination with DP in order to determine significantly tighter ε values for a majority of users or data points, simultaneously maintaining the classical DP guarantees for all. We experimented with deep learning and variational inference applications, and ε_μ

consistently remains well below 1, allowing for much more meaningful privacy guarantees.

Forth, we explore federated learning settings and adapt both Average-Case DP and Bayesian DP to these scenarios. We propose the first private federated data release approach based on GANs. It combines important advantages of both federated learning and private data release, such as higher flexibility and reduced requirements for user expertise and trust, and enables hierarchical data pooling and data trading. We also improve performance of private federated discriminative models by employing $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy. Since it requires less noise than DP to reach comparable privacy guarantees, models can be trained in fewer communication rounds. Simultaneously, privacy guarantees are significantly tighter, and thus, more meaningful. For client bases in the order of thousands, which is realistic in many federated learning scenarios, per-client ε can be as low as 1 or better. Lastly, we show that using the joint accounting technique allows to achieve client-level privacy "for free" when enforcing instance privacy. This way, we can maintain reasonably high accuracy, while keeping client privacy budgets close to $\varepsilon = 1$ and instance privacy budgets close to $\varepsilon = 0.1$.

Finally, we study the use of generative adversarial networks to tackle the problem of privacy-preserving data inspection and annotation in machine learning. Unlike the earlier approaches, which either provide no privacy guarantee or utilise differential privacy, we take advantage of Bayesian differential privacy. By capturing the inherent properties of data and allowing for non-uniform privacy loss throughout the dataset, it enables higher-fidelity synthetic data while still maintaining privacy guarantees comparable to DP. We show that privacy-preserving GANs with BDP can enable detection of subtle bugs in pre-processing pipelines or the data itself. This could not be achieved with DP GANs due to low quality of samples. Similarly, data biases and unseen classes can be discovered. Additionally, for some applications, generated data can be directly annotated and used for training in place of the real data. We demonstrate that student models trained on synthetic samples achieve significantly higher accuracy compared to prior state-of-the-art and exhibit only a mild drop in performance compared to directly learning a private classifier from real data. Moreover, to a large extent, this gap is explained by the quality of the generative model, and hence, will get smaller as the field advances.

8.2 Discussion

This thesis consists of two parts dedicated to two alternative privacy notions, average-case differential privacy, in Part I, and Bayesian differential privacy, in Part II. Apart from a few cross-references, these parts are largely independent, and before concluding this thesis, we find it important to discuss their relation in the bigger picture.

Both definitions share the idea of improving privacy guarantees by taking into account the data distribution. Effectively, it amounts to a shift in the definition of sensitivity.

In traditional privacy notions, sensitivity is defined as a hard, deterministic bound on a change in the output given a change in the input. In both ADP and BDP, sensitivity is a probabilistic quantity dependent on the natural randomness of the data and the data sampling process. However, the crucial difference between the two is that ADP is a function of past outputs, while BDP is a guarantee on future outputs. Consequently, it is more appropriate to think of ADP as a statistical measure of sensitivity. On the other hand, BDP is a privacy notion in the traditional sense, except that it uses a probabilistic sensitivity measure under the hood.

This conceptual difference transfers to differences in practical applications of these two definitions. First of all, due to the weakness of the ADP guarantee (i.e. it only bounds the expected value of the privacy loss), Bayesian DP is more suited for the applications that require stricter privacy protection. Conversely, average-case DP can be used when the goal is to assess potential privacy leakage in generative models, but not necessarily to impose additional protection. Second, related to the previous point, Bayesian DP is designed for privacy mechanisms with a controlled source of randomness independent of the data. In other words, mechanisms that impose additional privacy protection using a pre-defined noise distribution on top of the intrinsic data randomness. It means that BDP cannot be used unless such a mechanism is present, for example for *ex post* analysis of a generative model. To the contrary, we initially devised ADP for this kind of situations, where the developer has a reason to believe that the innate, uncontrolled sources of randomness and architecture of the model are sufficient to hide sensitive data characteristics. A prominent example of this is GANs. Furthermore, ADP can be used for models that are already deployed, while BDP only during training.

Finally, let us address a more direct, practical comparison of these notions in the application where both were applied – generating private artificial datasets using GANs. As follows from the discussion above, we cannot directly compare privacy guarantees because they bound different quantities. But keeping this in mind, we can contrast other aspects, such as the model quality and the real-world applicability. As mentioned in Chapter 7, only a part of the synthetic data utility loss comes from using a BDP mechanism in training. The major challenge is to find a set of hyper-parameters for which mode collapse does not occur. Once the stable training is achieved, the data quality is defined by the GAN capacity to a large extent. It suggests that ADP models, in spite of a considerably weaker guarantee, do not significantly gain in utility. Moreover, Bayesian DP is easier to implement, more computationally efficient (especially when per-example gradients are available (Goodfellow, 2015)), and relies on a much more principled and robust estimation framework. Hence, the only advantages of ADP are the increased training stability and a moderate improvement in utility. And if stability can be maintained in the presence of noise, Bayesian DP is clearly preferable.

In summary, Bayesian DP is the right choice if one needs a theoretical privacy guarantee and if training is robust enough against additional noise. If the model quality is severely

affected by noise, or if one just needs a statistical sensitivity measure for an already trained model, average-case DP can be used.

8.3 Future Directions

With ever increasing real-world use of machine learning, privacy research in this area is important, and at the moment, there is still a lack of good solutions, particularly for private data publishing. Hence, there are a lot of potential directions of future work.

One important question is the automation of privacy parameters search, such as noise variance and gradient clipping threshold. We treated these parameters similar to learning hyper-parameters, but it complicates the training process and increases the privacy budget (due to accounting all cross-validation or grid search runs). An adaptive procedure for choosing these parameters will be a significant contribution; not only for Bayesian DP, but for any differentially private and privacy-preserving ML. There were some attempts to address this problem, e.g. Abadi et al. (2016) proposing to use the median of gradient norms as a clipping threshold, but using non-sanitised information for such decision making is a potential privacy risk. Thus, it warrants further investigation.

For Bayesian DP in particular, a possible future direction is detection and mitigation of users and data points for which BDP privacy guarantees are not applicable, such as extreme outliers or out-of-distribution samples. We described a way of computing the probability of encountering "privacy outliers" in Chapters 5 and 7. It is also possible to compute a general DP bound in parallel with BDP. However, it would be preferable to provide tighter guarantees for points from similar distributions. More generally, given that the mechanism was calibrated to one data distribution, $x' \sim \mu(x)$, what are the privacy guarantees for data points whose distribution, $x' \sim \nu(x)$, imposes a different distribution on privacy loss. Perhaps, it is possible to derive expressions based on distances between protected and unprotected distributions. In high-dimensional spaces, which we primarily consider in this thesis, the privacy loss distribution is not likely to change significantly, but it would still be valuable to quantify this change and the change in the guarantee. A related question is how can auxiliary information about the privacy loss distribution affect guarantees of Bayesian DP, e.g. when conditions of Theorem 5 are not satisfied.

Another interesting direction for BDP is analysing the composition of privacy mechanisms for different data distributions. In this thesis, we considered composition of mechanisms applied to the data generated by the same distribution $\mu(x)$, and it is a common assumption in real-world applications. Nonetheless, in some applications, there may exist two or more different data distributions, i.e. $\mu_1(x)$ and $\mu_2(x)$, and it may be necessary to compose the mechanisms tailored to these individual distributions. Intuitively, both the basic and the advanced composition theorems should hold in this case, but a formal proof and analysis would be a worthwhile contribution.

More discriminative and generative models could be evaluated with both average-case DP and Bayesian DP. In model release settings, we were primarily interested in convolutional neural networks, but there are other commonly used models, such recurrent neural networks (RNNs). Training RNNs with privacy is more complicated, especially for long sequences, due to unrolling over time, and it presents a valuable experimental work. For data release scenarios, we used a rather basic Wasserstein GAN, but there are more advanced GANs for image data, GANs for discrete inputs, RNNs, VAEs, as well as other classes of generative and hybrid methods. A particularly interesting direction is studying behaviour of Bayesian differential privacy with the models that are difficult to train under the conventional DP. Again taking the example of RNNs, successful incorporation of BDP in RNNs could enable generating new classes of data privately, including audio sequences, location traces, texts, and so on.

In the context of using synthetic datasets for data annotation and ML troubleshooting, it would be interesting to perform a large scale crowdsourcing study to test possibilities of discovering irregularities in the datasets, such as under-represented classes, previously unseen classes, or biased distributions. For instance, one could eliminate examples of certain classes (e.g. shoes from Fashion-MNIST), and then set up an A/B style test where crowdsource workers would be shown synthetic images generated with DP and with Bayesian DP and asked if they could detect a bias in the synthetic dataset. Similar experiments could be arranged for data annotation. In Chapter 7, we used a private classifier to label artificial images, but it will not necessarily have perfect correlation with human annotators. Analysing these differences is another interesting research direction.

In the area of federated learning, a very prominent research direction is achieving practical local privacy guarantees, that is using the local model of differential privacy (see Definition 9). This notion is stronger than than the centralised model and requires more noise to be added at the clients. Bayesian DP can straightforwardly be adjusted to work in the local model, but if a client does not have information about other client updates distribution, there could be little or no gain in comparison with the traditional DP. An alternative is to employ a "hybrid" approach to BDP, akin to distributed differential privacy (Shi et al., 2011; Rastogi and Nath, 2010), and use a cryptographic scheme to compute privacy guarantees centrally while adding noise locally at the clients.

In general, research in machine learning methods with privacy guarantees can potentially have a very broad impact, even beyond fending off potential attacks. It could help improve accuracy of AI methods by unlocking access to more sensitive data, accelerate and extend AI development and deployment on mobile, wearable, and IoT devices, and facilitate medical research and data sharing.

A Appendix

A.1 Proofs of Propositions

This section contains proofs of propositions.

Proposition 2. *$(\varepsilon_\mu, \delta_\mu)$-strong Bayesian differential privacy implies $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy.*

Proof. Let us define a set of outcomes for which the privacy loss variable exceeds the ε threshold: $F(x') = \{w : L_{\mathcal{A}}(w, D, D') > \varepsilon\}$, and its compliment $F^c(x')$.

Observe that $L \leq \varepsilon$ implies $\Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x')] \leq e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S} \cap \mathcal{F}^c(x')]$, and therefore, $\Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x') \mid x'] \leq e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S} \cap \mathcal{F}^c(x') \mid x']$, because $\mathcal{A}(D)$ does not depend on x', and $\mathcal{A}(D')$ is already conditioned on x' through D'. Thus,

$$
\begin{aligned}
\Pr[\mathcal{A}(D) \in \mathcal{S}] &= \int \Pr[\mathcal{A}(D) \in \mathcal{S}, x']\, dx' && \text{(A.1)}\\
&= \int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x'), x'] && \text{(A.2)}\\
&\quad + \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x']\, dx' && \text{(A.3)}\\
&= \int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x') | x'] \mu(x') && \text{(A.4)}\\
&\quad + \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x']\, dx' && \text{(A.5)}\\
&\leq \int e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S} \cap \mathcal{F}^c(x') | x'] \mu(x') && \text{(A.6)}\\
&\quad + \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x']\, dx' && \text{(A.7)}\\
&\leq \int e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S}, x'] && \text{(A.8)}\\
&\quad + \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x']\, dx' && \text{(A.9)}\\
&\leq e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta_\mu, && \text{(A.10)}
\end{aligned}
$$

where in the first line we used marginalisation and the last inequality is due to the fact that

$$\int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{A.11}$$

$$\leq \int \Pr[\mathcal{A}(D) \in \mathcal{F}(x'), x'] \, dx' \tag{A.12}$$

$$= \int \mu(x') \Pr[\mathcal{A}(D) \in \mathcal{F}(x') \mid x'] \, dx' \tag{A.13}$$

$$= \int \mu(x') \int_{w \in \mathcal{F}(x')} p_{\mathcal{A}}(w|D, x') \, dw \, dx' \tag{A.14}$$

$$= \mathbb{E}_{x'} \left[\mathbb{E}_w \left[\mathbb{1}\{L > \varepsilon\} \right] \right] \tag{A.15}$$

$$\leq \delta_\mu \tag{A.16}$$

□

Proposition 3 (Post-processing)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for any arbitrary randomised data-independent mapping $f : \mathcal{R} \to \mathcal{R}'$, $f(\mathcal{A}(D))$ is $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private.*

Proof. By Proposition 2, $(\varepsilon_\mu, \delta_\mu)$-strong BDP implies

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta_\mu, \tag{A.17}$$

for any set of outcomes $\mathcal{S} \subset \mathcal{R}$.

For a data-independent function $f(\cdot)$:

$$\Pr\left[f(\mathcal{A}(D)) \in \mathcal{T}\right] = \Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \tag{A.18}$$

$$\leq e^{\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta_\mu, \tag{A.19}$$

$$= e^{\varepsilon_\mu} \Pr\left[f(\mathcal{A}(D')) \in \mathcal{T}\right] + \delta_\mu \tag{A.20}$$

where $\mathcal{S} = f^{-1}[\mathcal{T}]$, i.e. $\mathcal{S}$ is the preimage of $\mathcal{T}$ under f. □

Proposition 4 (Basic composition)**.** *Let $\mathcal{A}_i : \mathcal{D} \to \mathcal{R}_i$, $\forall i = 1..k$, be a sequence of $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithms. Then their combination, defined as $\mathcal{A}_{1:k} : \mathcal{D} \to \mathcal{R}_1 \times \ldots \times \mathcal{R}_k$, is $(k\varepsilon_\mu, k\delta_\mu)$-Bayesian differentially private.*

Proof. This property applies to both strong and weak senses of Bayesian DP. Let us begin with proving it for the strong sense.

Denote $L = \log \frac{p(w_1,\dots,w_k|D)}{p(w_1,\dots,w_k|D')}$ and let $L_i = \log \frac{p(w_i|D,w_{i-1},\dots,w_1)}{p(w_i|D',w_{i-1},\dots,w_1)}$. Then,

$$\Pr\left[L \geq k\varepsilon_\mu\right] = \Pr\left[\sum_{i=1}^{k} L_i \geq k\varepsilon_\mu\right] \tag{A.21}$$

$$\leq \sum_{i=1}^{k} \Pr[L_i \geq \varepsilon_\mu] \tag{A.22}$$

$$\leq \sum_{i=1}^{k} \delta_\mu \tag{A.23}$$

$$\leq k\delta_\mu \tag{A.24}$$

For the weak sense of BDP, the proof is similar to the proof of the basic composition theorem for the approximate DP (Dwork et al., 2014). For the case of two mechanisms:

$$\Pr\left[(\mathcal{A}_1(D), \mathcal{A}_2(D)) \in \mathcal{S}\right] = \int_{\mathcal{S}_1} p_{\mathcal{A}_1}(w_1) \Pr\left[(w_1, \mathcal{A}_2(D)) \in \mathcal{S}\right] dw_1 \tag{A.25}$$

$$\leq \int_{\mathcal{S}_1} p_{\mathcal{A}_1}(w_1) \left(\left(e^{\varepsilon_\mu} \Pr\left[(w_1, \mathcal{A}_2(D')) \in \mathcal{S}\right]\right) \wedge 1 + \delta_\mu\right) \tag{A.26}$$

$$\leq \int_{\mathcal{S}_1} p_{\mathcal{A}_1}(w_1) \left(\left(e^{\varepsilon_\mu} \Pr\left[(w_1, \mathcal{A}_2(D')) \in \mathcal{S}\right]\right) \wedge 1\right) + \delta_\mu \tag{A.27}$$

$$\leq \int_{\mathcal{S}_1} (e^{\varepsilon_\mu} p_{\mathcal{A}'_1}(w_1) + \delta_\mu) \tag{A.28}$$

$$\times \left(\left(e^{\varepsilon_\mu} \Pr\left[(w_1, \mathcal{A}_2(D')) \in \mathcal{S}\right]\right) \wedge 1\right) + \delta_\mu \tag{A.29}$$

$$\leq e^{2\varepsilon_\mu} \int_{\mathcal{S}_1} p_{\mathcal{A}'_1}(w_1) \Pr\left[(w_1, \mathcal{A}_2(D')) \in \mathcal{S}\right] + 2\delta_\mu \tag{A.30}$$

$$\leq e^{2\varepsilon_\mu} \Pr\left[(\mathcal{A}_1(D'), \mathcal{A}_2(D')) \in \mathcal{S}\right] + 2\delta_\mu, \tag{A.31}$$

where $\wedge$ operator defines the minimum between the left and the right arguments.

The general case of k mechanisms follows by induction.

□

Proposition 5 (Group privacy)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for all pairs of datasets $D, D' \in \mathcal{D}$, differing in k data points $x_1, \dots, x_k$ s.t. $x_i \sim \mu(x)$ for $i = 1..k$, $\mathcal{A}(D)$ is $(k\varepsilon_\mu, ke^{k\varepsilon_\mu}\delta_\mu)$-Bayesian differentially private.*

Proof. Let us define a sequence of datasets D^i, $i = 1..k$, s.t. $D = D^0$, $D' = D^k$, and D^i and D^{i-1} differ in a single example. Then,

$$\frac{p(w|D)}{p(w|D')} = \frac{p(w|D^0)p(w|D^1)\dots p(w|D^{k-1})}{p(w|D^1)p(w|D^2)\dots p(w|D^k)} \tag{A.32}$$

Denote $L_i = \log \frac{p(w|D^{i-1})}{p(w|D^i)}$ for $i = 1..k$.

Applying the definition of $(\varepsilon_\mu, \delta_\mu)$-strong Bayesian differential privacy,

$$\Pr\left[L \geq k\varepsilon_\mu\right] = \Pr\left[\sum_{i=1}^{k} L_i \geq k\varepsilon_\mu\right] \tag{A.33}$$

$$\leq \sum_{i=1}^{k} \Pr[L_i \geq \varepsilon_\mu] \tag{A.34}$$

$$\leq k\delta_\mu \tag{A.35}$$

For $(\varepsilon_\mu, \delta_\mu)$-BDP,

$$Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon_\mu} Pr\left[\mathcal{A}(D^1) \in \mathcal{S}\right] + \delta_\mu \tag{A.36}$$

$$\leq e^{\varepsilon_\mu}\left(e^{\varepsilon_\mu} Pr\left[\mathcal{A}(D^2) \in \mathcal{S}\right] + \delta_\mu\right) + \delta_\mu \tag{A.37}$$

$$\leq e^{2\varepsilon_\mu} Pr\left[\mathcal{A}(D^2) \in \mathcal{S}\right] + e^{\varepsilon_\mu}\delta_\mu + \delta_\mu \tag{A.38}$$

$$\leq e^{3\varepsilon_\mu} Pr\left[\mathcal{A}(D^3) \in \mathcal{S}\right] + e^{2\varepsilon_\mu}\delta_\mu + e^{\varepsilon_\mu}\delta_\mu + \delta_\mu \tag{A.39}$$

$$\leq \ldots \tag{A.40}$$

$$\leq e^{k\varepsilon_\mu} \Pr\left[\mathcal{A}(D^k) \in \mathcal{S}\right] + \frac{e^{k\varepsilon_\mu} - 1}{e^{\varepsilon_\mu} - 1}\delta_\mu \tag{A.41}$$

$$\leq e^{k\varepsilon_\mu} \Pr\left[\mathcal{A}(D^k) \in \mathcal{S}\right] + \frac{k\varepsilon_\mu e^{k\varepsilon_\mu}}{\varepsilon_\mu}\delta_\mu \tag{A.42}$$

$$\leq e^{k\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + ke^{k\varepsilon_\mu}\delta_\mu, \tag{A.43}$$

where in (A.41) we use the formula for the sum of a geometric progression; in (A.42), the facts that $e^x - 1 \leq xe^x$, for $x > 0$, and $e^x \geq x + 1$.

□

Bibliography

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Abowd, J. M., Schneider, M. J., and Vilhuber, L. (2013). Differential privacy applications to bayesian and linear mixed model estimation. *Journal of Privacy and Confidentiality*, 5(1):4.

Aldous, D. J. (1985). Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198. Springer.

Amos, B., Ludwiczuk, B., Satyanarayanan, M., et al. (2016). Openface: A general-purpose face recognition library with mobile applications.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., et al. (2019). Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*.

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6280–6290.

Bassily, R. and Freund, Y. (2016). Typical stability. *arXiv preprint arXiv:1604.03336*.

Bassily, R., Groce, A., Katz, J., and Smith, A. (2013). Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 439–448. IEEE.

Beaulieu-Jones, B. K., Wu, Z. S., Williams, C., and Greene, C. S. (2017). Privacy-preserving generative deep neural networks support clinical data sharing. *bioRxiv*, page 159756.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.

Berck, P. and Hihn, J. M. (1982). Using the semivariance to estimate safety-first rules. *American Journal of Agricultural Economics*, 64(2):298–300.

Bhaskar, R., Bhowmick, A., Goyal, V., Laxman, S., and Thakurta, A. (2011). Noiseless database privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 215–232. Springer.

Bindschaedler, V. and Shokri, R. (2016). Synthesizing plausible privacy-preserving location traces. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 546–563. IEEE.

Bindschaedler, V., Shokri, R., and Gunter, C. A. (2017). Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment*, 10(5).

Blum, A., Ligett, K., and Roth, A. (2013). A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecny, J., Mazzocchi, S., McMahan, H. B., et al. (2019). Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proc. ACM CCS*, pages 1175–1191. ACM.

Bun, M. (2017). A teaser for differential privacy.

Bun, M., Dwork, C., Rothblum, G. N., and Steinke, T. (2018). Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–86. ACM.

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

Charest, A.-S. and Hou, Y. (2017). On the meaning and limits of empirical differential privacy. *Journal of Privacy and Confidentiality*, 7(3):3.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109.

Chen, L., Tao, C., Zhang, R., Henao, R., and Duke, L. C. (2018). Variational inference and model selection with generalized evidence bounds. In *International Conference on Machine Learning*, pages 892–901.

Chen, M., Mathews, R., Ouyang, T., and Beaufays, F. (2019). Federated learning of out-of-vocabulary words. *arXiv preprint arXiv:1903.10635*.

Dandekar, A., Basu, D., and Bressan, S. (2020). Differential privacy at risk: Bridging randomness and privacy budget. *arXiv preprint arXiv:2003.00973*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. (2017). Variational inference via χ upper bound minimization. In *Advances in Neural Information Processing Systems*, pages 2732–2741.

Dimitrakakis, C., Nelson, B., Mitrokotsa, A., and Rubinstein, B. I. (2014). Robust and private bayesian inference. In *International Conference on Algorithmic Learning Theory*, pages 291–305. Springer.

Dimitrakakis, C., Nelson, B., Zhang, Z., Mitrokotsa, A., and Rubinstein, B. I. (2017). Differential privacy for bayesian inference through posterior sampling. *The Journal of Machine Learning Research*, 18(1):343–381.

Duan, Y. (2009). Privacy without noise. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1517–1520. ACM.

Dwork, C. (2006). Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy. Springer Verlag.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology (EUROCRYPT 2006)*, volume 4004, page 486–503, Saint Petersburg, Russia. Springer Verlag.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

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

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

Fioretto, F. and Van Hentenryck, P. (2019). Privacy-preserving federated data sharing. In *Proc. AAMAS 2019*, pages 638–646.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM.

Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. (2014). Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32.

Geumlek, J., Song, S., and Chaudhuri, K. (2017). Renyi differential privacy mechanisms for posterior sampling. In *Advances in Neural Information Processing Systems*, pages 5289–5298.

Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Gil, M., Alajaji, F., and Linder, T. (2013). Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131.

Goodfellow, I. (2015). Efficient per-example gradient computations. *arXiv preprint arXiv:1510.01799*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1462–1471.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779.

Hall, R., Rinaldo, A., and Wasserman, L. (2011). Random differential privacy. *arXiv preprint arXiv:1112.2680*.

Hardy, C., Le Merrer, E., and Sericola, B. (2019). Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 866–877. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, X., Machanavajjhala, A., and Ding, B. (2014). Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1447–1458. ACM.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. *arXiv preprint arXiv:1706.08500*.

Hitaj, B., Ateniese, G., and Pérez-Cruz, F. (2017). 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. ACM.

Huang, C., Kairouz, P., Chen, X., Sankar, L., and Rajagopal, R. (2017). Context-aware generative adversarial privacy. *Entropy*, 19(12):656.

Humbatova, N., Jahangirova, G., Bavota, G., Riccio, V., Stocco, A., and Tonella, P. (2019). Taxonomy of real faults in deep learning systems. *arXiv*, pages arXiv–1910.

Jälkö, J., Dikmen, O., and Honkela, A. (2016). Differentially private variational inference for non-conjugate models. *arXiv preprint arXiv:1610.08749*.

Jordon, J., Yoon, J., and van der Schaar, M. (2018). Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Kifer, D. and Machanavajjhala, A. (2014). Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):3.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 972–981.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. Citeseer.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

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

Leung, S. and Lui, E. (2012). Bayesian mechanism design with efficiency, privacy, and approximate truthfulness. In *International Workshop on Internet and Network Economics*, pages 58–71. Springer.

Li, J., Khodak, M., Caldas, S., and Talwalkar, A. (2020). Differentially private meta-learning. In *International Conference on Learning Representations*.

Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE.

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Long, Y., Lin, S., Yang, Z., Gunter, C. A., and Li, B. (2019). Scalable differentially private generative student model via pate. *arXiv preprint arXiv:1906.09338*.

Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., and Vilhuber, L. (2008). Privacy: Theory meets practice on the map. In *2008 IEEE 24th international conference on data engineering*, pages 277–286. IEEE.

Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3.

Maurer, A. and Pontil, M. (2009). Empirical bernstein bounds and sample-variance penalization. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2017). Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2018). Learning differentially private recurrent language models.

Mir, D. J. (2012). Information-theoretic foundations of differential privacy. In *International Symposium on Foundations and Practice of Security*, pages 374–381. Springer.

Mironov, I. (2017). Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE.

Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. (2009). Computational differential privacy. In *Annual International Cryptology Conference*, pages 126–142. Springer.

Mironov, I., Talwar, K., and Zhang, L. (2019). R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5.

Oliphant, T. E. (2006). A bayesian perspective on estimating mean, variance, and standard-deviation from data.

Oord, A. V. d., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1747–1756.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. (2016). Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. (2018). Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*.

Pérez-Cruz, F. (2008). Kullback-leibler divergence estimation of continuous distributions. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1666–1670. IEEE.

Pihur, V., Korolova, A., Liu, F., Sankuratripati, S., Yung, M., Huang, D., and Zeng, R. (2018). Differentially-private“ draw and discard” machine learning. *arXiv preprint arXiv:1807.04369*.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rastogi, V. and Nath, S. (2010). Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. pages 1278–1286.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234.

Samarati, P. (2001). Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027.

Samarati, P. and Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

Saw, J. G., Yang, M. C., and Mo, T. C. (1984). Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132.

Schneider, M. J. and Abowd, J. M. (2015). A new method for protecting interrelated time series with bayesian prior distributions and synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(4):963–975.

Shi, E., Chan, T. H., Rieffel, E., Chow, R., and Song, D. (2011). Privacy-preserving aggregation of time-series data. In *Proc. NDSS*, volume 2, pages 1–17. Citeseer.

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

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE.

Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 245–248. IEEE.

Sweeney, L. (2002). K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.

Triastcyn, A. and Faltings, B. (2019a). Federated generative privacy. In *IJCAI Workshop on Federated Machine Learning for User Privacy and Data Confidentiality (FML 2019)*.

Triastcyn, A. and Faltings, B. (2019b). Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2587–2596. IEEE.

Triastcyn, A. and Faltings, B. (2019c). Generating artificial data for private deep learning. In *Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies, AAAI Spring Symposium Series*, pages 33–40.

Triastcyn, A. and Faltings, B. (2020a). Bayesian differential privacy for machine learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR.

Triastcyn, A. and Faltings, B. (2020b). Federated generative privacy. *IEEE Intelligent Systems*, pages 50–57.

Triastcyn, A. and Faltings, B. (2020c). Generating higher-fidelity synthetic datasets with privacy guarantees. *arXiv preprint arXiv:2003.00997*.

Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11.

Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. (2018). Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*.

Van Erven, T. and Harremos, P. (2014). Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

Wang, W., Ying, L., and Zhang, J. (2016a). On the relation between identifiability, differential privacy, and mutual-information privacy. *IEEE Transactions on Information Theory*, 62(9):5018–5029.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. (2019). Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235.

Wang, Y.-X., Fienberg, S., and Smola, A. (2015). Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pages 2493–2502.

Wang, Y.-X., Lei, J., and Fienberg, S. E. (2016b). On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms. In *International Conference on Privacy in Statistical Databases*, pages 121–134. Springer.

Waugh, S. G. (1995). *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. PhD thesis, University of Tasmania.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. (2018). Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, pages 7335–7345.

Yang, B., Sato, I., and Nakagawa, H. (2015). Bayesian differential privacy on correlated data. In *Proceedings of the 2015 ACM SIGMOD international conference on Management of Data*, pages 747–762. ACM.

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., and Yu, H. (2019). Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207.

Yao, A. C.-C. (1982). Protocols for secure computations. In *FOCS*, volume 82, pages 160–164.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41.

Zhang, X., Ji, S., and Wang, T. (2018). Differentially private releasing via deep generative model. *arXiv preprint arXiv:1801.01594*.

Zhao, J., Mathieu, M., and LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

Zhu, T., Li, G., Zhou, W., and Philip, S. Y. (2017). Differentially private data publishing and analysis: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638.

Aleksei Triastcyn

PhD Candidate with 8+ years of hands-on machine learning experience, now working on privacy-preserving machine learning and federated learning.

Chavannes-près-Renens
Switzerland
+41 78 826 88 50
aleksey.tryastsyn@gmail.com

Education

Jul 15, 2015 - Present
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
Artificial Intelligence Lab,
PhD in Computer Science.
Thesis director: Prof. Boi Faltings.
Thesis topic: Practical Privacy-Preserving Machine Learning.

Sep 19, 2011 - Apr 7, 2014
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
MSc in Computer Science, GPA: 5.33 / 6.
Thesis topic: Context-Aware Recommender Systems Using Tensor Factorisation.
Courses: Pattern classification and machine learning, Applied Machine Learning, Information theory and coding, Advanced Algorithms, Image and video processing, Automatic speech processing, Intelligent Agents

Sep 1, 2007 - Jun 22, 2011
Perm State University, Perm, Russia.
BSc in Applied Mathematics and Computer Science, GPA: 4.8 / 5 (honours degree).
Thesis topic: Tracking Moving Objects in Multi-Camera Systems.

Work experience

Jul 15, 2015 - present
Research Assistant, *EPFL*, Route Cantonale, 1015 Lausanne, Switzerland.
- *Bayesian Differential Privacy*
  - Proposed a novel theoretical privacy notion, a relaxation of differential privacy, tailored for ML.
  - Its main idea is to obtain more meaningful privacy guarantees by calibrating them to data distribution, with minimal and realistic assumptions, and by estimating the necessary statistics from data.
  - Evaluated this approach on various classification tasks with deep neural networks and variational inference. Also, adapted and evaluated it on federated learning.
- *Generating Artificial Data for Private Learning using GANs*
  - Was one of the first to propose using GANs for private data synthesis.
  - Showed its applicability on image classification tasks and resistance against model inversion attacks.
  - Designed an empirical framework for estimating the average-case privacy leakage.
  - Extended the approach to federated learning.

I also successfully supervised 9 bachelor and master student projects, and assisted in teaching courses on Artificial Intelligence, Intelligent Agents, Information Theory and Coding.
Tools: Python, PyTorch
Skills: research, writing, teaching, supervision, deep learning, differential privacy, federated learning

Apr 28, 2014 - Jun 30, 2015
Software Engineer, *Microsoft Development Center Norway*, Torggata 2-4-6, 0181 Oslo, Norway.
I contributed to delivering intelligent features for Office 365 users through the Office Graph.
- Quickly on-boarded and became the go-to person for a low-level C++ codebase.
- Increased the test coverage to ∼100% and fixed a critical multi-threading issue.
- Increased the reliability and availability of service by implementing consistency checking and recovery.
- Was a leading engineer of the identification system for graph nodes.

Tools: C/C++, C#
Skills: production quality code, scalable and reliable design, code reviews, testing

Sep 23, 2013 - Mar 22, 2014
Research Intern, *Sony Deutschland GmbH*, Hedelfinger Str. 61, 70327 Stuttgart, Germany.
Research internship in recommender systems:
- Within the first month, covered relevant research literature and implemented baseline algorithms.
- Designed an improved tensor factorisation technique suitable for context-aware recommendations.
- Developed low-level C routines for customised CPU- and GPU-based sparse tensor operations to achieve 9x speed up compared to existing implementations.
- Demonstrated overall benefits of using GPUs by benchmarking important aspects of ML computations.

Tools: C/C++, Matlab, MEX, CUDA
Skills: recommender systems, matrix factorisation, tensor algebra, convex optimisation, sparse data

Apr 29, 2013 - Sep 20, 2013 **Intern in Medical Imaging**, *Siemens Corporate Research, Inc.*, 755 College Rd E, Princeton, NJ 08540, USA.

- Familiarised myself with existing object tracking methods for medical imaging.
- Improved the tracking performance by using information fusion from multiple algorithms alleviating their individual issues.

Tools: C/C++, OpenCV, Qt
Skills: computer vision, object tracking, information fusion

Aug 2012 **Software Developer**, *EPFL*, Route Cantonale, 1015 Lausanne, Switzerland.

- Familiarised myself with transform-domain scrambling and OpenCV.
- Implemented human faces obfuscation in video frames, so as to make them unrecognisable in a raw video but recoverable with a cryptographic key, using OpenCV and transform-domain scrambling software.
- The results were published at the MediaEval 2012 Workshop.

Tools: C/C++, OpenCV
Skills: computer vision, face detection, image processing

Research Activities

Papers (journals, conferences, workshops, symposia)

- Aleksei Triastcyn, Boi Faltings. Bayesian Differential Privacy for Machine Learning. To appear in *37th International Conference on Machine Learning (ICML 2020)*.
- Aleksei Triastcyn, Boi Faltings. Federated Generative Privacy. In *IEEE Intelligent Systems, Special Issue on Federated Machine Learning*, 2020.
- Aleksei Triastcyn, Boi Faltings. Bayesian Differential Privacy for Machine Learning. In *AAAI Workshop on Privacy-Preserving Artificial Intelligence*, New York, USA, Feb 7, 2020.
- Aleksei Triastcyn, Boi Faltings. Federated Learning with Bayesian Differential Privacy. In *IEEE International Conference on Big Data (IEEE Big Data 2019)*, Los Angeles, USA, Dec 9-12, 2019.
- Aleksei Triastcyn, Boi Faltings. Federated Learning with Bayesian Differential Privacy. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, Vancouver, Canada, Dec 13, 2019.
- Aleksei Triastcyn, Boi Faltings. Federated Generative Privacy. In *IJCAI Workshop on Federated Machine Learning for User Privacy and Data Confidentiality*, Macao, China, Aug 12, 2019.
- Aleksei Triastcyn, Boi Faltings. Generating Artificial Data for Private Deep Learning. In *AAAI Spring Symposium Series*, Stanford, Palo Alto, USA, March 25-27, 2019.
- Pavel Korshunov, Aleksei Triastcyn, Touradj Ebrahimi. MediaEval 2012 Visual Privacy Task: Applying Transform-domain Scrambling to Automatically Detected Faces. In *MediaEval 2012 Workshop*, Pisa, Italy, Oct 4-5, 2012.
- Aleksei Triastcyn, Andrei Durakov. Searching people in video streams using clothes pattern: move detection algorithms. In *All-Russian Science Conference "Actual Problems of Mechanics, Mathematics and Informatics"*, Perm, Russia, Oct 12-15, 2010, p. 226.
- Andrei Durakov, Kirill Yurkov, Ilya Blokh, Kirill Luzin, Anna Sivkova, Aleksei Triastcyn. United system of monitoring and notification. In *All-Russian Science Conference of Young Scientists*, Part 5, St. Petersburg, Russia, 2010, p. 95.

Pre-prints, theses, reports

- Aleksei Triastcyn, Boi Faltings. Generating Higher-Fidelity Synthetic Datasets with Privacy Guarantees. *arXiv preprint arXiv:2003.00997* (2020).
- Aleksei Triastcyn. Context-Aware Recommender Systems Using Tensor Factorisation. *Master's Thesis*, 2014.
- Aleksei Triastcyn. Spike and Slab Priors for Expectation Propagation. *Technical Report*, EPFL, Lausanne, Switzerland, 2013.
- Aleksei Triastcyn. Tracking Moving Objects in Multi-Camera Systems. *Bachelor's Thesis*, 2011.

Patents

- Macksood, A., Triastcyn, A.N., Seleskerov, K., Knudsen, V.T. and Sakkos, P., Microsoft Technology Licensing LLC, 2017. *Exposing external content in an enterprise.* U.S. Patent Application 14/845,978. (Pending)
- Triastcyn, A., John, M., Wang, P., Siemens AG, 2014. *Information Fusion for Calcification Tracking.* Invention Disclosure, DOI 10.4421/PAPDEOTT002904.

Posters and talks
- International Conference on Machine Learning (ICML 2020). Online, July 12-18, 2020.
- AAAI Workshop on Privacy-Preserving Artificial Intelligence. New York, USA, February 7, 2020.
- NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality. Vancouver, Canada, December 13, 2019.
- IEEE International Conference on Big Data. Los Angeles, USA, December 9-12, 2019.
- Engineering PhD Summit, Intelligent Systems. Lausanne, Switzerland, October 3, 2019.
- IJCAI Workshop on Federated Machine Learning for User Privacy and Data Confidentiality, Macao, China, August 12, 2019.
- AAAI Spring Symposium Series. Palo Alto, USA, March 25-27, 2019.
- IC Summer Research Institute. Lausanne, Switzerland, June 18-19, 2018.
- IC Research Day. Lausanne, Switzerland, June 7, 2018.
- SDSC-Connect Workshop. Bern, Switzerland, November 29, 2017.

Reviewing
- Workshop on Privacy in Natural Language Processing (in conjunction with EMNLP 2020). *Technical Program Committee member.*
- IEEE Transactions on Dependable and Secure Computing. *Reviewer.*
- ACM International Conference on Information and Knowledge Management (2020). *Reviewer.*
- International Workshop on Federated Machine Learning for User Privacy and Data Confidentiality (in conjunction with IJCAI 2020). *Technical Program Committee member.*
- Computers & Security, Elsevier Journal. *Reviewer.*
- IEEE International Symposium on Information Theory (2020). *Reviewer.*
- IEEE Intelligent Systems Magazine. *Reviewer.*
- Workshop on Privacy in Natural Language Processing (in conjunction with WSDM 2020). *Technical Program Committee member.*
- AAAI/ACM Conference on AI, Ethics, and Society (2020). *Reviewer.*

Awards and competitions

Awards and scholarships
- Best Presentation Award at IJCAI Workshop on Federated Machine Learning (2019)
- People's Choice Award at MDCN Hackathon (2015)
- Microsoft Ship-It Award for Office 365 (2015)
- Swiss Government Scholarship for Foreign Students (2011-2013)
- Diploma with honours for Bachelor's degree (2011)
- Merit-based scholarship at Perm State University (2008-2011, 6/7 semesters)

Competitions
- Hackathon at Microsoft Development Center Norway (2015)
- Hackathon at Siemens Corporate Research (2013)
- HSE National Student Olympiad in System and Software Engineering (2011)
- Microsoft Imagine Cup (2010)
- Local programming contests at high school and university (2006-2008)

Technical expertise

Languages: Python, C/C++, C#, Java, Matlab, SQL

Tools: Jupyter, Visual Studio, Eclipse, Xcode, Git

ML: PyTorch, Scikit-Learn, Scipy, Numpy

Hobbies

Skiing, Hiking, Mountaineering, Photography, Guitar, Cycling