Differentially private and distributed Bayesian learning

Mikko Heikkilä

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki in Physicum E204 on 25.8.2023 at 12 o'clock.

Department of Mathematics and Statistics
Faculty of Science
University of Helsinki
Finland

Supervisor
Antti Honkela, University of Helsinki, Finland

Pre-examiners
Mi Jung Park, University of British Columbia, Canada
Yu-Xiang Wang, University of California, Santa Barbara, United States

Opponent
Borja de Balle Pigem, Google DeepMind, United Kingdom

Custos
Antti Honkela, University of Helsinki, Finland

Contact information

Department of Mathematics and Statistics
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

ISBN 978-951-51-9363-6 (paperback)
ISBN 978-951-51-9364-3 (PDF)
Helsinki 2023
Unigrafia

Differentially private and distributed Bayesian learning

Mikko Heikkilä

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
mixheikk@gmail.com
http:///mixheikk.github.io/

PhD Thesis
Helsinki, July 2023
ISBN 978-951-51-9363-6 (paperback)
ISBN 978-951-51-9364-3 (PDF)

Abstract

Machine learning aims to learn patterns from data. When the data are about people, a machine learning model will learn information about people. Such models can be used by malicious actors to attack the individuals, whose data have been used in training the model. The goal of privacy-preserving machine learning is to guarantee that such attacks are not successful.

Differential privacy is a mathematical definition of privacy, based on randomising the learning process: if the result would have been nearly the same whether any given individual was present in the training data set or not, then the result will not violate the privacy of any given individual.

The articles in this dissertation focus on combining differential privacy with Bayesian machine learning, which is a learning paradigm based on representing the quantities of interest as probability distributions, and using the standard rules of probability calculus to update these distributions in light of the available data to arrive at a posterior. The posterior combines our prior beliefs with evidence from data.

As for the data, while it is often realistic to consider a single centralised data set that is directly available for learning, there are also many settings where the data set is distributed among several parties. In such cases, especially with sensitive personal data, we want to have learning methods that do not require centralising all the data, but work directly in the distributed setting.

In the first article included in this dissertation, we consider differentially private Bayesian learning from data that are distributed between several parties. We propose a secure secret sharing method that can be combined with differential privacy for increased utility. We show theoretically that, under some assumptions, learning conjugate-exponential family models can be done efficiently with the proposed method, and empirically test the method with Bayesian linear regression models.

In the second publication, we introduce a general method for approximating intractable distributions via sampling from a suitable Markov chain while guaranteeing differential privacy. The method is based on analysing the inherently stochastic accept-reject decisions needed to produce samples from the chain, without the need to add extra randomness for privacy.

In the third article, we develop methods for numerically establishing the total privacy level in the shuffle model of differential privacy, a specific distributed learning setting. We show how to improve this privacy accounting for some algorithms commonly used for guaranteeing differential privacy.

Finally, in the fourth article, we introduce a general framework for approximating intractable distributions based on variational inference, when the data are distributed among several parties and we additionally want to minimise the number of communication rounds between the parties. Within the general framework, we propose, analyse, and compare three specific alternatives with varying characteristics.

Acknowledgements

I would first like to thank my supervisor Professor Antti Honkela, who has taken the time and made all the effort to instruct, encourage and support me during the years, even when things did not seem to move at a stellar pace. In retrospect, I am very happy to have walked into that particular office in Exactum to discuss the possibility of pursuing a doctoral degree.

I am also grateful for having had a number of very good co-authors and colleagues during the years, especially in the machine learning privacy groups at University of Helsinki, lead by Prof. Antti Honkela, and at Aalto University, headed by Prof. Sami Kaski. I have enjoyed and benefited much from being able to discuss privacy, machine learning, music etc. including with Onur, Joonas, Antti, Razan, Lukas, Ossi, and Marlon.

Doing actual research is a lot easier when one does not need to spend the majority of time writing grant applications. Luckily, for most of the time I was supported by a longer-term funding from the doctoral school DOMAST, as well as by smaller grants including from Nokia Foundation.

Finally, it goes without saying that I am very grateful for all the support to my friends and to my family Elina, Konsta and Janne, and most of all to my parents Terttu and Olli; I promise this is the last degree I will pursue. More than anyone else, the ones who have had to endure through countless hours of endless musings, complaining, and random chatter about machine learning, with and without privacy, are Satu and Kaarlo. Apologies, maybe we can now talk about something else for a change.

Barcelona, June 2023
Mikko Heikkilä

List of included publications

Publication I: Heikkilä, M. A., Lagerspetz, E., Kaski, S., Shimizu, K., Tarkoma, S., and Honkela, A. (2017). Differentially private Bayesian learning on distributed data. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Publication II: Heikkilä, M. A., Jälkö, J., Dikmen, O., and Honkela, A. (2019). Differentially private Markov chain Monte Carlo. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Publication III: Koskela, A., Heikkilä, M. A., and Honkela, A. (2023). Numerical accounting in the shuffle model of differential privacy. In *Transactions on Machine Learning Research*, 2023.

Publication IV: Heikkilä, M. A., Ashman, M., Swaroop, S., Turner, R. E., and Honkela, A. (2023). Differentially private partitioned variational inference. In *Transactions on Machine Learning Research*, 2023.

Author's contributions

In Publication I, MH participated in formulating the main ideas of the article, formulated the proposed encryption scheme jointly with the other authors, wrote the privacy proofs for the proposed algorithms, did most of the implementations and simulations, and participated heavily in writing the final article.

In Publication II, MH participated in planning the main ideas of the article, wrote the proofs jointly with JJ, implemented the algorithms jointly with JJ, and co-wrote the final article with the other authors. II is also included in the dissertation of JJ.

In Publication III, MH helped in planning the main ideas of the article, proposed and analysed the weaker adversary model, and participated in writing the final article.

In Publication IV, MH participated in coming up with the main ideas of the article, wrote the proofs, wrote all privacy-specific code for implementing the proposed algorithms, and lead the writing for the final article.

Note on the notations

The notation used in this dissertation is fairly standard and aims to be easily readable. For example, for probability distributions, we overload the notation in a standard way by identifying the distributions by their arguments. Therefore, for a distribution $p_\theta(\theta)$ we generally simply write $p(\theta)$. This should not create any confusions.

Considering measure-theoretic backdrops needed for formulating some definitions and results in a mathematically rigorous way, we generally assume, e.g., that random variables $P, Q : \Omega \to \mathbb{R}$ are absolutely continuous w.r.t. some reference measure on Ω, so the Radon-Nikodym derivative $\frac{dP}{dQ}$ is simply a density ratio. We will write such ratios with lower case $\frac{p}{q}$ with the understanding that p, q are densities or probability mass functions as necessary (e.g. in Definitions 5 & 6, and in Section 3.1.3).

Contents

1 Introduction 1

2 Differential privacy 5

2.1 Basic properties of differential privacy 7

2.1.1 Robustness against powerful adversaries 7

2.1.2 Immunity to post-processing 8

2.1.3 Group-privacy 8

2.2 Alternative differential privacy definitions 9

2.3 Standard DP mechanisms 12

2.4 Privacy amplification by data subsampling 16

2.5 Composing differentially private algorithms 18

2.5.1 Composing general DP algorithms 18

2.5.2 Composing specific DP algorithms: the Gaussian mechanism 21

2.6 From centralised to distributed DP 22

2.6.1 DP with secure aggregation 26

2.6.2 Shuffle model of DP 28

3 Privacy-preserving Bayesian learning 31

3.1 Bayesian learning 31

3.1.1 Conjugate-exponential family of distributions 32

3.1.2 Markov chain Monte Carlo 34

3.1.3 Variational inference 37

3.2 DP Bayesian learning 39

3.2.1 DP via posterior sampling 40

3.2.2 Sufficient statistic perturbation 41

3.2.3 Approximate posterior sampling with DP MCMC . . 43

3.2.4 DP variational inference 44

4 Conclusion 47

5 Thesis contribution **49**
5.1 Publication I 49
5.2 Publication II 49
5.3 Publication III 50
5.4 Publication IV 50

References **53**

Chapter 1

Introduction

Personal privacy is a fundamental right in a free society. While there probably has never been a period in history when individual privacy would not have been under various threats, recent years have seen the emergence of entirely new possibilities for attacking individual privacy. These changes are driven by the increase of online interactions in all spheres of life, along with the advance of machine learning.[1]

With the advent of mobile computing, sensitive data are often produced on individual mobile devices. In such cases, centralising all user data for learning might not be ideal or even possible. For example, centralising mobile device user data would typically require some form of consent from the device users and could incur additional costs as well as create a huge liability to the data curator. Distributed learning methods, such as algorithms for federated learning (McMahan et al., 2017; Kairouz et al., 2021b), avoid the problems caused by data centralisation by enabling learning directly on the distributed data.

However, when the data are used for training machine learning models, simply keeping the data secure is not enough: when models are trained with data that are in some sense about individuals, such as web searches, online shopping histories, or mobility traces, the models can contain information about specific individuals. A trained model can therefore be used to attack individuals present in the training data, if the attacker can extract the

[1] See, e.g., Roser et al. 2015 for estimates on the global number of internet users, Ferrantino and Koten 2019 for some context on the increase in e-commerce on a global level, Roussi 2020 for an example case where the space of individual privacy is encroached upon by machine learning outside any intentional online activity, Villalobos et al. 2022 for the increase in large-scale machine learning models sizes, Google 2023 for some examples of data set sizes used for training machine learning models, and Kearns and Roth 2019 for a more general discussion.

information from the model (see, for example, Fredrikson et al. 2014, 2015; Carlini et al. 2019).

Generally, with almost any machine learning model, it is very hard to know exactly what information the model contains. In addition, there is also no clear boundary on what is the type of information that could be used for attacking a given individual: almost anything could be used for launching an attack when combined with other suitable data.[2]

The privacy problems with machine learning models are unfortunately not purely theoretical: there are several known attacks, ranging from training data reconstructions, where the attacker aims to reconstruct the data points used for training a machine learning model (Fredrikson et al., 2014, 2015; Carlini et al., 2019; Balle et al., 2022; Zhu et al., 2019; Zhao et al., 2020; Geng et al., 2021; Carlini et al., 2021), to membership inference attacks, where the attacker tries to identify which data points were part of the training data set (Homer et al., 2008; Shokri et al., 2017; Yeom et al., 2018; Nasr et al., 2019; Long et al., 2018, 2020; Carlini et al., 2022; Watson et al., 2022; Ye et al., 2022).

More generally, a classic result by Dinur and Nissim (2003), known as the Fundamental result of information recovery, shows that too accurate answers to too many questions inevitably leads to a catastrophic privacy failure, regardless of any possible protection method employed. Therefore, any method that aims to protect privacy in learning needs to somehow quantify exactly how accurate the answers can be for a given number of queries presented to the data.

The current gold-standard in privacy protection for learning from data is *differential privacy* (Dwork et al., 2006b), which formalises mathematically what it means that an algorithm protects privacy. Differential privacy is based on having randomness in the answers to the queries presented to the data, to avoid the catastrophic privacy failure implied by the Fundamental result of information recovery. The resulting strictness of the privacy-guarantee can then be quantified with numbers or privacy parameters.

The articles in this thesis mainly focus on differentially private machine learning, and more specifically, on differentially private Bayesian machine learning:[3] we want to enable updating prior beliefs into more informed

[2] The problems with identifying what information a model learns or what might be a problematic prediction affects machine learning also beyond pure privacy concerns. Identifying and trying to remedy these kinds of issues has lead to a more general emphasis on trustworthy machine learning, where privacy is one subtopic (see e.g. Varshney 2022; Barocas et al. 2019).

[3] See, e.g., Ghahramani (2015) for an accessible high-level introduction to Bayesian machine learning, and Murphy (2012) for a technical and in-depth presentation.

posterior beliefs based on data, and we want to do this while protecting the privacy of data subjects by providing differential privacy.

In order to guarantee individual privacy, however, the first requirement is to keep the sensitive data secure. As noted above, learning from distributed data is an important research direction which can help reduce risks in many settings, e.g., of large-scale data breaches, since the sensitive data are kept distributed on several devices. For this reason, distributed learning is the second main thread running through the articles.

On a general level therefore, the aim of the articles included in this dissertation is to enable privacy-preserving Bayesian learning in any given setting. The specific questions and the contributions of the publications are discussed in detail in introducing the required theory for differentially private Bayesian learning, with centralised as well as with distributed data, in the following sections. This is done in an effort to clarify how the articles relate to the larger field of differentially private machine learning. The main ideas and contributions of the publications are also concisely summarised in isolation in Section 5.

In the next sections, we first formally define differential privacy and discuss it's properties in Section 2, starting with centralised data and ending by changing to the distributed setting. After thereby clarifying what we actually mean by privacy-preserving learning, we move on to discuss privacy-preserving Bayesian learning in Section 3. We start by introducing the main ideas and methods for standard non-DP Bayesian learning in Section 3.1. To finish, we review the most relevant existing approaches to DP Bayesian learning in Section 3.2, and discuss how the included publications fit into this more specialised field.

Chapter 2

Differential privacy

At its core, differential privacy (DP) is a stability guarantee on the output of a randomised algorithm: when a stochastic algorithm guarantees DP, it means that the observed result would have been almost as likely to occur with a slightly different input data. Another way to state this is to say that each sample can only have a limited effect on the output.

The connection to privacy becomes clearer when we think of the input as containing data on individuals: if changing a single individual's data arbitrarily does not alter the output probabilities much, then the output contains relatively little information about any given individual. Formally, we have the following:

Definition 1 (DP, Dwork et al. 2006b,a)**.** *Let $\varepsilon > 0$ and $\delta \in [0,1]$. A randomised algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is (ε, δ)-DP if for every neighbouring $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and every measurable set $E \subset \mathcal{O}$,*

$$\mathbb{P}(\mathcal{M}(\mathbf{x}) \in E) \leq \mathrm{e}^{\varepsilon}\mathbb{P}(\mathcal{M}(\mathbf{x}') \in E) + \delta.$$

$\mathcal{M}$ is tightly (ε, δ)-DP, if there does not exist $\delta' < \delta$ such that $\mathcal{M}$ is (ε, δ')-DP. When $\delta = 0$, we write ε-DP and call it pure DP*. The more general case (ε, δ)-DP is called* approximate DP *(ADP).*

The strictness of DP guarantees can be tuned with the privacy parameters ε and δ, while the granularity of the protection depends the chosen neighbourhood definition. While there is no general agreement on how to choose the privacy parameters, it is often held that ε should be a small constant, while δ should ideally be cryptographically small, or increasing slower than any inverse polynomial in the size of the data set. In practice, commonly used values are, e.g., $\varepsilon \simeq 1$, and $\delta < \frac{1}{|\mathbf{x}|}$.[1]

[1] While ε is typically taken to be small, even huge values might offer protection against

While smaller privacy parameters offer better privacy, they also reduce *model utility*, e.g., prediction accuracy or model likelihood, since tighter privacy essentially requires $\mathcal{M}$ to be more random. For many standard algorithms used for guaranteeing DP, however, the utility can be increased by increasing the amount of data available. Hence, there is always a trade-off between privacy protection and model utility, but the terms of this trade-off can often be mitigated by adding more data.

Instead of choosing a single (ε, δ) point, we ideally want to consider the full range of (tight) privacy parameters achievable by a given randomised algorithm $\mathcal{M}$. This idea is formalised in privacy profiles:

Definition 2 (Privacy profile, Balle et al. 2018)**.** *A function* $\delta_{\mathcal{M}} : \mathbb{R} \to [0, 1]$*,* $\varepsilon \mapsto \delta$ *s.t.* $\mathcal{M}$ *is* (ε, δ)*-DP is called a* privacy profile function*. The corresponding set of all possible privacy parameter values* $\{(\varepsilon, \delta(\varepsilon))\}, \varepsilon > 0$ *is called a* privacy profile*.*

We denote the privacy profile function simply as $\delta(\cdot)$ when there is no risk of confusion. A tight privacy profile, where $\mathcal{M}$ in Definition 2 is tightly $(\varepsilon, \delta(\varepsilon))$-DP $\forall\varepsilon$, is a Pareto frontier, which partitions the space of privacy parameters into the region where $\mathcal{M}$ will satisfy DP and to the region where it does not.

As is clear from Definition 1, what constitutes a "small change" in the input space depends on our neighbourhood definition. The most common DP *protection granularity* is on the sample level, often called sample-level or item-level DP, especially in federated learning: the neighbouring $\mathbf{x}, \mathbf{x}'$ differ by a single sample (see Section 2.6 for a discussion on some other possible granularities in the context of federated learning). When not mentioned explicitly, we assume that the granularity is on a single-sample level, and that the database has a single sample per individual. In this case, sample-level DP corresponds to individual-level DP.

Besides the protection granularity, we also need to choose a *neighbourhood relation*. The usual choices are the *add/remove relation* or *unbounded DP*, meaning that we can turn $\mathbf{x}$ into $\mathbf{x}'$ by adding or removing a single element, and the *replace* or *substitute relation*, sometimes also called *bounded DP*, which means that we can turn $\mathbf{x}$ into $\mathbf{x}'$ by substituting a single element with another one. In all the included publications, we use the replace relation.

some forms of attack (Carlini et al., 2019). The reason for the given δ is that taking $\delta = o(\frac{1}{n})$ allows for releasing some records in the clear without any perturbation while still satisfying Definition 1 via a probabilistic DP argument (see Dwork and Roth 2014; Meiser 2018). This kind of approach to privacy, which provides good protection to most data subjects at the cost of completely failing on a few cannot be considered acceptable.

In general, depending on the setting, some specific relation can feel more natural or turn out to be easier to analyse than others, but for now, we simply treat the relation as an arbitrary choice that fixes the privacy scale. When fixing the exact neighbourhood relation is not important, we simply write $\mathbf{x} \sim \mathbf{x}'$ for any neighbourhood relation.

2.1 Basic properties of differential privacy

DP has several nice properties that make it appealing as a privacy definition. The most important basic ones are i) robustness against powerful adversaries, ii) immunity to post-processing, and iii) group-privacy, which will be covered in short order. Additional properties requiring lengthier treatment, namely, privacy amplification by data subsampling, and composability, are treated separately in Sections 2.4 and 2.5, respectively, after introducing more tools needed for analysing DP in Section 2.2.

2.1.1 Robustness against powerful adversaries

Definition 1 does not explicitly mention any adversary, but is instead a bound on the randomised algorithm $\mathcal{M}$, which holds in the information theoretic sense. As such, it provides guarantees including against a strong computationally unbounded adversary, with access to the entire database $\mathbf{x}$ except for a single element, who tries to make inferences about the final element using arbitrary side information.[2]

In practice, however, implementing DP on finite computers with pseudo-randomness, as well as combining DP with secure protocols (see Section 2.6) requires, e.g., that the adversary is computationally bounded (Mironov et al., 2009; Meiser, 2018).[3]

[2] It is sometimes claimed, that DP requires this so-called strong adversary assumption, i.e., that the adversary has access to the entire database except for a single element (see Tschantz et al. 2020 for a discussion on this point and for references). This is not generally true. Instead, as shown by Tschantz et al. (2020), DP can be understood via causal models without any hidden assumptions.

[3] Implementing DP algorithms that assume real numbers using standard floating-point representation can also cause subtle vulnerabilities that can be exploited even by computationally bounded adversaries: for example, simply taking a floating-point value generated with a pseudo-random number generator from any standard mathematical software, which is not specifically designed for DP, is likely vulnerable to an attack, originally proposed by Mironov (2012), which utilises the imprecision caused by the floating-point representation: in effect, generating values from, say, Gaussian or Laplace distribution is based on first generating pseudo-random numbers from a standard uniform distribution and then transforming these to have the required distribution. However, the transformed values do not cover all the floating points in a uniform manner anymore, but

In this thesis, we mostly use the unbounded adversary that is implicitly covered by Definition 1. This is a common abstraction, but it should be kept in mind that this hides important details that are crucial for actually making DP practical (see e.g. Mironov et al. 2009; Haney et al. 2022; Casacuberta et al. 2022).

Finally, while DP is a strong privacy notion, it does not guarantee that an adversary will not learn anything about a given individual in the protected database: in contrast, learning anything useful even on the aggregate level requires leaking at least some information about the individuals making up the population. DP is explicitly designed to leak information in a controlled fashion so that learning is possible without necessarily causing a catastrophic privacy breach (see e.g. Dinur and Nissim 2003; Dwork et al. 2006b; Dwork and Roth 2014).

2.1.2 Immunity to post-processing

Immunity to post-processing means that the output of a DP algorithm cannot be made less DP without gaining access to the original data or to the internal state of the DP algorithm. This is essential in practice, since it enables releasing the results of DP algorithms without having to worry about some clever attacker coming up with transformations that would breach the privacy protection. Formally we have the following:

Theorem 3 (Post-processing immunity, Dwork and Roth 2014)**.** *Assume a randomised algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is (ε, δ)-DP. Let $f : \mathcal{O} \to \mathcal{O}'$ be an arbitrary (randomised) mapping. Then $f \circ \mathcal{M}$ is (ε, δ)-DP.*

Proof. See Dwork and Roth (2014, Proposition 2.1). □

2.1.3 Group-privacy

The privacy guaranteed by DP also readily extends to more than a single element, i.e., the DP guarantees degrade gracefully when considering groups instead of individuals.

Theorem 4 (Group privacy, Dwork and Roth 2014)**.** *Assume a randomised algorithm $\mathcal{M}$ is (ε, δ)-DP. Then for a group of size $k \geq 1$, $\mathcal{M}$ is $(k\varepsilon, \delta \cdot (\sum_{i=0}^{k-1} \mathrm{e}^{i\varepsilon}))$-DP.*

there will be gaps in the possible values. An attacker might therefore be able to tell for sure which one of the neighbouring data sets was the actual input simply by inspecting the output.

Proof. Write $\mathbf{x}^{(k)}$ for a data set that differs from $\mathbf{x}$ on k elements, so $\mathbf{x}, \mathbf{x}^{(1)}$ are neighbours in the sense of Definition 1. Let $E \subset \mathcal{O}$ be measurable.

$$\begin{aligned}\mathbb{P}(\mathcal{M}(\mathbf{x}) \in E) &\leq \mathrm{e}^{\varepsilon}\mathbb{P}(\mathcal{M}(\mathbf{x}^{(1)}) \in E) + \delta \\ &\leq \mathrm{e}^{2\varepsilon}\mathbb{P}(\mathcal{M}(\mathbf{x}^{(2)}) \in E) + \delta \cdot (\mathrm{e}^{\varepsilon} + 1) \\ &\vdots \\ &\leq \mathrm{e}^{k\varepsilon}\mathbb{P}(\mathcal{M}(\mathbf{x}^{(k)}) \in E) + \delta \cdot (\sum_{i=0}^{k-1} \mathrm{e}^{i\varepsilon}).\end{aligned}$$

□

2.2 Alternative differential privacy definitions

Definition 1 is not the only way to define DP: there are alternative definitions which are equivalent to Definition 1 in the privacy sense, but provide a different perspective that can be easier to analyse (see Zhu et al. 2022 for more discussion on the recently introduced representations and on transforming between them).

Another class of definitions consists of relaxations of the standard DP definition. The aim of these relaxations is typically to make calculating the total privacy parameters easier, especially under composition.

In this section, we present some important privacy definitions from both classes. These are needed in Section 2.4 for privacy amplification by data subsampling, and in Section 2.5 for composing DP algorithms.

The following Definition 5 characterises DP as a bound on the hockey-stick divergence between probability distributions, a member of f-divergence family of divergences (see e.g. Barthe and Olmedo 2013), and is equivalent to Definition 1:

Definition 5 (DP via hockey-stick divergence, Barthe and Olmedo 2013)**.** *A randomised algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is $(\varepsilon, \delta(\varepsilon))$-DP, iff*

$$\delta(\varepsilon) \geq \sup_{\mathbf{x},\mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'} H_{\mathrm{e}^{\varepsilon}}(\mathcal{M}(\mathbf{x}) \| \mathcal{M}(\mathbf{x}')), \tag{2.1}$$

where for $\alpha > 0$ the Hockey-stick divergence *is defined as*

$$H_{\alpha}(P\|Q) = \mathbb{E}_{o \sim Q}\left[\max\{0, \frac{p(o)}{q(o)} - \alpha\}\right]. \tag{2.2}$$

$\mathcal{M}$ is tightly $(\varepsilon, \delta(\varepsilon))$-DP, if Equation 2.1 holds with equality.

Another useful characterisation of DP is based on looking at the privacy loss:

Definition 6 (Privacy loss, Dwork and Rothblum 2016; Bun and Steinke 2016)**.** *Let P, Q be two random variables defined on $\mathcal{O}$, and define the* privacy loss function *as*

$$f_{p/q} : \mathcal{O} \to \mathbb{R} \cup \{-\infty, \infty\}, f_{p/q}(o) = \log \frac{p(o)}{q(o)}.$$

The privacy loss random variable *for P over Q, denoted as $\mathcal{L}_{p/q}$, is distributed as $f_{p/q}(P)$.*

To connect the privacy loss in Definition 6 with (ε, δ)-DP via Definition 2, we have the following result:

Theorem 7 (Privacy loss and ADP, Balle and Wang 2018; Balle et al. 2018)**.** *Let $\mathcal{M}$ be (ε, δ)-DP algorithm. Then for every $\mathbf{x}, \mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'$, and writing $P = \mathcal{M}(\mathbf{x}), Q = \mathcal{M}(\mathbf{x}')$, the privacy profile of $\mathcal{M}$ satisfies*

$$\delta_{\mathcal{M}}(\varepsilon) \geq \mathbb{P}(\mathcal{L}_{p/q} \geq \varepsilon) - \mathrm{e}^{\varepsilon}\mathbb{P}(\mathcal{L}_{q/p} \leq -\varepsilon).$$

Proof. See Balle and Wang (2018, Theorem 5). □

The privacy loss formulation is most commonly used for (numerically) composing DP algorithms (see Section 2.5).

Another important privacy definition that is based on looking at the privacy loss, and which sits between pure DP and ADP, and is therefore not equivalent to Definition 1, is called Rényi differential privacy (RDP):

Definition 8 (Rényi DP, Mironov 2017)**.** *A randomised algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is α-Rényi differentially private of order α, written (α, ε)-RDP, if for every $\mathbf{x}, \mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'$,*

$$D_\alpha(\mathcal{M}(\mathbf{x}) \| \mathcal{M}(\mathbf{x}')) \leq \varepsilon, \tag{2.3}$$

where for $\alpha > 1$, the Rényi divergence is defined as

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{t \sim Q} \left(\frac{p(t)}{q(t)} \right)^\alpha, \tag{2.4}$$

where p, q are densities corresponding to the distributions P, Q.

RDP essentially bounds specific moments of the privacy loss random variable via the moment generating function. This becomes clear after a simple manipulation (see e.g. Steinke 2022): for distributions P, Q, the RDP bound in Definition 8 can be equivalently written as

$$\begin{aligned}
\exp((1-\alpha)\varepsilon) \geq & \mathbb{E}_{t\sim Q}\left[\frac{p(t)}{q(t)}\right]^{\alpha} && (2.5)\\
= & \int \left[(p(t))^{\alpha}(q(t))^{1-\alpha}\right] dt && (2.6)\\
= & \int p(t)\left[\frac{p(t)}{q(t)}\right]^{\alpha-1} dt && (2.7)\\
= & \mathbb{E}_{t\sim P}\left[\frac{p(t)}{q(t)}\right]^{\alpha-1} && (2.8)\\
= & \mathbb{E}_{t\sim P}\left\{\exp\left[(\alpha-1)\log\left(\frac{p(t)}{q(t)}\right)\right]\right\} && (2.9)\\
= & \mathbb{E}\left\{\exp\left[(\alpha-1)\mathcal{L}_{p/q}\right]\right\}. && (2.10)
\end{aligned}$$

RDP is closely related to other privacy notions focusing on the moments of the privacy loss, such as the variants of concentrated DP (CDP, Dwork and Rothblum 2016; zero-concentrated DP or zCDP, Bun and Steinke 2016; truncated concentrated DP or tCDP, Bun et al. 2018), which bound all or several of the moments of the privacy loss at once.

Any RDP bound also directly implies ADP:

Theorem 9 (From RDP to ADP, Mironov 2017). *Assume a randomised algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ is (α, ε)-RDP. Then $\mathcal{M}$ is $(\varepsilon + \frac{\log(1/\delta)}{\alpha-1}, \delta)$-DP.*

Proof. See Mironov (2017, Proposition 3). □

While Theorem 9 shows that RDP implies ADP, the converse is not true in general (see Zhu et al. 2022 for examples). What is more, the conversion from RDP to ADP is necessarily lossy even in cases where RDP seems to behave well. That is, while the simple conversion result given in Theorem 9 can be improved upon, RDP does not fully characterise ADP privacy profile, since this would allow for lossless conversions from RDP to ADP and back (Zhu et al. 2022, see also Balle et al. 2020a).

RDP enjoys many of the nice properties of DP (see Section 2.1), like immunity to post-processing and a form of group-privacy (see e.g. Mironov 2017; Steinke 2022).

We generally use (ε, δ)-DP as the basic privacy definition in all the included articles. However, we use RDP in Publication II to make the privacy analysis with subsampling and composition more tractable (see Sections 2.4 & 2.5 for privacy amplification by subsampling and composing

DP algorithms, respectively), and then convert the final privacy parameters to ADP via Theorem 9.

So far, all the DP definitions have been defined directly in terms of some neighbouring data sets $\mathbf{x}, \mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'$ by looking at the output distributions $\mathcal{M}(\mathbf{x}), \mathcal{M}(\mathbf{x}')$. For further analysis, especially when considering subsampling and composition in ADP in Sections 2.4 & 2.5, it will often be more convenient to work instead with dominating pairs of distributions:

Definition 10 (Dominating pairs, Zhu et al. 2022)**.** *A pair of distributions* (P, Q) *is a* dominating pair *of distributions for a randomised algorithm* $\mathcal{M}$*, if for all* $\alpha \geq 0$*,*

$$\sup_{\mathbf{x},\mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'} H_\alpha(\mathcal{M}(\mathbf{x}) || \mathcal{M}(\mathbf{x}')) \leq H_\alpha(P || Q).$$

If the equality holds for all α*, then* (P, Q) *is* tightly dominating.

The reason for introducing dominating pairs is that for some DP algorithms, there does not exist a single pair of worst-case data sets $\mathbf{x} \sim \mathbf{x}'$ that result in tight privacy bounds (see Zhu et al. 2022 for an example, where the worst-case pair depends on the privacy parameters).

Unlike worst-case pairs, tightly dominating pairs always exists for any DP algorithm (Zhu et al., 2022, Proposition 8), although finding such a pair for a given randomised algorithm might not be trivial. When a worst-case pair exists, the distributions induced by the worst-case pair also form a tightly dominating pair. Therefore, to establish tight privacy bounds for a given algorithm, it is enough to find and analyse a single tightly dominating pair. In Publication III, we show how to find dominating pairs of distributions for some common DP algorithms in the shuffle DP setting (see Section 2.6).

2.3 Standard DP mechanisms

Randomised algorithms satisfying DP are often called *mechanisms*. There are several standard mechanisms for guaranteeing DP. In this section, we will define the most common ones and state their respective privacy bounds.

We start with *k-randomised response* (kRR), which is often used as a mechanism for guaranteeing pure DP when the data are categorical, such as responses to some survey question:

Definition 11 (k-randomised response, Warner 1965)**.** *Let* $\mathcal{M} : [k] \rightarrow [k], k \in \{2, 3, \ldots\}$ *be a randomised algorithm, such that for each* $\mathbf{x}_j, j =$

$1, \dots, n$,

$$\mathbb{P}(\mathcal{M}(\mathbf{x}_j) = i) = \begin{cases} 1 - (\frac{k-1}{k})\zeta & \textit{if } x_j = i \\ \frac{\zeta}{k} & \textit{else}, \end{cases} \tag{2.11}$$

where $\zeta \in (0, 1)$.

Essentially, for each sample $\mathbf{x}_j \in \{1, \dots, k\}$ in the input data set $\mathbf{x}$, the kRR mechanism in Definition 11 simply outputs the input value with probability $1 - \zeta$, and a uniformly random value from the range $\{1, \dots, k\}$ otherwise. Theorem 12 states the resulting DP bounds when using kRR as the privacy mechanism.

Theorem 12. *Let $\mathcal{M}$ be a kRR algorithms as in Definition 11. Then $\mathcal{M}$ is bounded ε-DP with*

$$\varepsilon = \log\left(\frac{k}{\zeta} - k + 1\right).$$

Proof. Directly from the definition, w.l.o.g. assume that the neighbouring data sets differ only on the first sample: $\mathbf{x}_1 \neq \mathbf{x}'_1$. For any outcome vector s we have

$$\frac{\mathbb{P}(\mathcal{M}(\mathbf{x}) = s)}{\mathbb{P}(\mathcal{M}(\mathbf{x}') = s)} = \frac{\prod_{i=1}^n \mathbb{P}(\mathcal{M}(\mathbf{x}_i) = s_i)}{\prod_{i=1}^n \mathbb{P}(\mathcal{M}(\mathbf{x}'_i) = s_i)} \tag{2.12}$$

$$= \frac{\mathbb{P}(\mathcal{M}(\mathbf{x}_1) = s_1)}{\mathbb{P}(\mathcal{M}(\mathbf{x}'_1) = s_1)} \tag{2.13}$$

$$\leq \frac{1 - (\frac{k-1}{k}) \cdot \zeta}{\frac{\zeta}{k}} \tag{2.14}$$

$$= \mathrm{e}^{\log(\frac{k}{\zeta} - k + 1)}, \tag{2.15}$$

and the result follows directly from Definition 1. □

In Publication III, we analyse the kRR mechanism in the federated learning setting under shuffle DP (see Section 2.6), numerically establishing privacy bounds that are tighter than the previously known analytic bounds.

For releasing continuous-valued data under DP, the most common mechanisms are Laplace mechanism for pure DP and Gaussian mechanism for ADP. As a preliminary, we first need to define function sensitivity, which is needed for calibrating the noise level properly for a given function:

Definition 13 (Sensitivity, Dwork et al. 2006b)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function. The* ℓ_1-sensitivity *of f is given by*

$$\Delta_1(f) = \sup_{\mathbf{x}, \mathbf{x}' \in \mathcal{X} : \mathbf{x} \sim \mathbf{x}'} \|f(\mathbf{x}) - f(\mathbf{x}')\|_1, \tag{2.16}$$

*and the ℓ_2-*sensitivity *of f is given by*

$$\Delta_2(f) = \sup_{\mathbf{x},\mathbf{x}' \in \mathcal{X}: \mathbf{x} \sim \mathbf{x}'} \|f(\mathbf{x}) - f(\mathbf{x}')\|_2. \tag{2.17}$$

As noted above, the Laplace mechanism is one standard approach for guaranteeing pure DP with continuous data:

Definition 14 (Laplace mechanism, Dwork et al. 2006b)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_1-sensitivity Δ_1.* Laplace mechanism *is a randomised algorithm $\mathcal{M}$, such that*

$$\mathcal{M}(f, \mathbf{x}, b) = f(\mathbf{x}) + \xi, \tag{2.18}$$

where $\xi_k \sim$ Laplace$(0, b), k = 1, \dots, d$, i.e., ξ is a d-dimensional random variable s.t. each dimension independently follows a Laplace distribution with mean zero and variance $2b^2$.

Considering the inputs to the Laplace mechanism in Definition 14, we want the privacy guarantees to hold with fixed f and b. Theorem 15 quantifies the privacy guarantees due to using the Laplace mechanism for releasing continuous function output values:

Theorem 15 (Pure DP with Laplace mechanism, Dwork et al. 2006b)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_1-sensitivity Δ_1. Releasing the function value through the corresponding Laplace mechanism $\mathcal{M}$ with noise parameter $b = \frac{\Delta_1}{\varepsilon}$ is ε-DP.*

Proof. See e.g. Dwork and Roth (2014, Theorem 3.6). □

The Gaussian mechanism, which adds spherical Gaussian noise to function outputs scaled with the function sensitivity and the privacy parameters, is the most common mechanism for providing ADP with continuous data:

Definition 16 (Gaussian mechanism, Dwork et al. 2006a)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δ_2.* Gaussian mechanism *is a randomised algorithm $\mathcal{M}$, such that*

$$\mathcal{M}(f, \mathbf{x}, \sigma) = f(\mathbf{x}) + \xi, \tag{2.19}$$

where $\xi \sim \mathcal{N}(0, \sigma^2 I_d)$, and $\sigma > 0$.

As with the Laplace mechanism, we want DP guarantees to hold with fixed f and σ, and therefore typically omit the arguments to the Gaussian mechanism except for the data. Establishing the privacy guarantees using the Gaussian mechanism is more involved than with the Laplace mechanism. The so-called classical Gaussian mechanism result, given in Theorem 17, requires that $\varepsilon \in (0, 1)$, and the resulting bounds can be loose even on that interval:

Theorem 17 (Classical Gaussian mechanism DP bounds, Dwork and Roth 2014)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δ_2. Releasing the function value through the corresponding Gaussian mechanism $\mathcal{M}$ with noise parameter $\sigma^2 \geq 2\log(\frac{5}{4\delta})\frac{\Delta_2^2}{\varepsilon^2}$ is (ε, δ)-DP for any $\varepsilon \in (0, 1), \delta \in (0, 1]$.*

Proof. See Dwork and Roth (2014, Theorem A.1). □

We use the classical Gaussian mechanism DP bounds (Theorem 17) in Publication I.

Theorem 18 gives tight ADP bounds using the Gaussian mechanism.

Theorem 18 (Analytical Gaussian mechanism DP bounds, Balle and Wang 2018)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δ_2. Releasing the function value through the corresponding Gaussian mechanism $\mathcal{M}$ using $\sigma > 0$ is (ε, δ)-DP for any $\varepsilon \geq 0$, $\delta \in [0, 1]$ iff*

$$\delta(\varepsilon) \geq \Phi(\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}) - \mathrm{e}^{\varepsilon}\Phi(-\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}), \tag{2.20}$$

where Φ is the standard normal cdf.

Proof. See Balle and Wang (2018, Theorem 8). □

The main advantage in the classical Gaussian mechanism bounds (Theorem 17) compared to the tight bound (Theorem 18) is the easy analytical form, which also allows for establishing asymptotical optimality results. However, in the current research these kinds of optimality results have increasingly been based on RDP (see Section 2.2) due to the ease of tighter composition and to the known closed-form subsampling amplification results, as well as to avoid the upper bound on ε.

Theorem 19 (Gaussian mechanism RDP bounds, Mironov 2017)**.** *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δ_2. Releasing the function value through the corresponding Gaussian mechanism $\mathcal{M}$ using $\sigma > 0$ is $(\alpha, \frac{\alpha\Delta_2^2}{2\sigma^2})$-RDP.*

Proof. See Mironov (2017, Proposition 7). □

One main workhorse in the current privacy-preserving machine learning is DP stochastic gradient descent (DP-SGD, a basic version is given in Algorithm 1), which uses the Gaussian mechanism (Definition 16) to guarantee ADP in gradient-based optimisation. Besides being important for implementing DP in practice, characterising the exact DP bounds for DP-SGD with subsampling amplification has also been an important direction for the development of privacy accounting more generally (see Section 2.5).

Algorithm 1 DP-SGD (Song et al., 2013)

```
Require: Differentiable loss function $l(\mathbf{x};\theta)$, step size $\alpha$, initial values $\theta_0$, maximum $\ell_2$-norm bound $C$, Gaussian mechanism noise level $\sigma$, total number of optimisation steps $T$.
for $t = 1$ to $T$ do
    Choose a minibatch $B_t$.
    for each sample $\mathbf{x}_j$ in $B_t$ do
        Calculate gradient: $g_j \leftarrow \nabla_\theta l(\mathbf{x}_j;\theta_{t-1})$.
        Clip per-example gradient: $\tilde{g}_j \leftarrow \min\{1, \frac{C}{\|g_j\|_2}\} \cdot g_j$.
    end for
    Aggregate and add Gaussian noise: $g_{DP} \leftarrow \sum_j \tilde{g}_j + \xi$, where $\xi \sim \mathcal{N}(0, C^2\sigma^2 I)$.
    Take an optimisation step: $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot g_{DP}$
end for
return Optimised DP parameters $\theta_T$.
```

We use DP-SGD in Publication IV as one alternative for enforcing DP in partitioned VI (see Section 3.1.3): when the local optimisation runs in PVI are done with DP-SGD, due to the post-processing guarantees (Theorem 3) the trained model will have DP guarantees.

2.4 Privacy amplification by data subsampling

Privacy amplification refers to the general phenomenon where additional randomisation in the learning results in better DP bounds. One of the most important examples of privacy amplification is amplification by subsampling: when a given DP algorithm is run on a randomly sampled minibatch of data, the resulting DP bounds will improve depending on the sampling fraction.

Privacy amplification by subsampling has been considered under various settings and privacy definitions (see e.g. Chaudhuri and Mishra 2006; Kasiviswanathan et al. 2011; Beimel et al. 2014; Bassily et al. 2014; Abadi et al. 2016; Balle et al. 2018; Zhu and Wang 2019)

We will first define the sampling without replacement (WOR) subsampling function that is used in the publications, which returns a constant size minibatch:

Definition 20 (Sampling without replacement)**.** *A randomised function* $\mathbf{WOR}_\gamma$ *is a* sampling without replacement *subsampling function, if it maps a data set* $\mathbf{x}$ *of size* n *or* $n-1$ *to a uniformly random minibatch of size* $b < n$. *The sampling fraction is defined to be* $\gamma = \frac{b}{n}$.

Considering the amplification results, it turns out that a given subsampling function is typically easier to analyse with specific neighbourhood definition (see e.g. Balle et al. 2018; Zhu et al. 2022). Therefore, the privacy amplification results for Poisson subsampling, which is the second common subsampling function that returns varying-sized minibatches, are usually based on the add/remove neighbourhood definition, while the results for WOR subsampling results typically assume the replace neighbourhood.

The next theorems state amplification results for the $\mathbf{WOR}_\gamma$ subsampling.

Theorem 21 (Amplification by subsampling for dominating pairs, Zhu et al. 2022)**.** *Let* $\mathcal{M}$ *be a randomised algorithm, and* $\mathbf{WOR}_\gamma$ *a subsampling function as in Definition 20. If* (P, Q) *dominates* $\mathcal{M}$ *with replace relation for data set of size* γn*, then for all neighbours* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$,

$$H_\alpha\left(\mathcal{M} \circ \mathbf{WOR}_\gamma(\mathbf{x}) \| \mathcal{M} \circ \mathbf{WOR}_\gamma(\mathbf{x}')\right) \leq \begin{cases} H_\alpha\left(\gamma P + (1-\gamma) Q \| Q\right) \textit{ for } \alpha \geq 1, \\ H_\alpha\left(P \| (1-\gamma) P + \gamma Q\right) \textit{ for } 0 < \alpha < 1. \end{cases}$$

Proof. See Zhu et al. (2022, Proposition 30). □

We use Theorem 21 together with Algorithm 1 from Doroshenko et al. (2022) for calculating subsampling amplification in Publication III in the shuffle model of DP (see Section 2.6).

With RDP and WOR subsampling, we have the following privacy amplification result:

Theorem 22 (WOR subsampling amplification for RDP, Wang et al. 2019)**.** *Let* $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ *be a randomised algorithm, and let* $\mathbf{WOR}_\gamma$ *be a subsampling function as in Definition 20. If* $\mathcal{M}$ *is* $(\alpha, \varepsilon(\alpha))$*-RDP with* $\alpha \geq 2, \alpha \in \mathbb{N}$, *then the subsampled mechanism* $\mathcal{M} \circ \mathbf{WOR}_\gamma$ *is* $(\alpha, \varepsilon'(\alpha))$*-RDP with*

$$\begin{aligned} \varepsilon'(\alpha) \leq \frac{1}{\alpha-1} \log \Bigg(1 + \gamma^2 \binom{\alpha}{2} \min\left\{ 4(\mathrm{e}^{\varepsilon(2)} - 1), \mathrm{e}^{\varepsilon(2)} \min\{2, (\mathrm{e}^{\varepsilon(\infty)} - 1)^2\} \right\} \\ + \sum_{j=3}^{\alpha} \gamma^j \binom{\alpha}{j} \mathrm{e}^{(j-1)\varepsilon(j)} \min\{2, (\mathrm{e}^{\varepsilon(\infty)} - 1)^j\} \Bigg). \end{aligned} \tag{2.21}$$

Proof. See Wang et al. (2019, Theorem 10). □

We utilise Theorem 22 in Publication II to enable better RDP bounds when drawing samples with a DP Markov chain Monte Carlo method, when each accept-reject decision is based only on a minibatch of data (see Section 3.1.2).

2.5 Composing differentially private algorithms

The general idea in composition is that when sensitive data about a given individual is used in several algorithms, the privacy guarantees should degrade in a controlled manner. This is essentially what happens with DP, although the details depend on the privacy definition and on the specific type of composition.

Calculating the total privacy cost from repeated queries to DP algorithms is called *privacy accounting.* Establishing more advanced privacy accounting techniques that lead to improved or even tight DP bounds has been an important research topic, which has seen significant progress during recent years.

One fairly recent innovation in privacy accounting is the focus on numerical accounting techniques (see e.g. Meiser and Mohammadi 2018; Koskela et al. 2020a; Mironov et al. 2019), which often leads to significantly tighter bounds than analytical methods, even when the analytical methods can be shown to have the correct asymptotic scaling. The tighter numerical bounds can be invaluable for real-world deployment, where the constants are important.

We start by stating the main results for composing arbitrary (ε, δ)-DP algorithms, as well as for the RDP composition. To improve on the general bounds, we then move on to consider more specific DP algorithms, namely, the (subsampled) Gaussian mechanism.

2.5.1 Composing general DP algorithms

We are interested in the *adaptive sequential composition*, where the later DP mechanisms can depend on the output from the earlier algorithms. This composition type is suitable, e.g., for running DP-SGD (see Algorithm 1), where we have a fixed data set and fixed DP algorithms. In the following, we refer to the adaptive sequential composition simply as adaptive composition.

There are other possible composition types, like the adaptive composition for a fixed sequence of (ε, δ)-DP mechanisms defined via a composition game (see Dwork et al. 2010b; Rogers et al. 2016), a generalisation of the aforementioned to non-fixed sequences of DP algorithms (Rogers et al., 2016), or the adaptive concurrent composition (Vadhan and Wang, 2021), but we do not consider them in this dissertation.

Definition 23 (Adaptive sequential composition). *Let $\mathcal{M}_i : \mathcal{X} \to \mathcal{O}, i = 1, \dots, k$ be DP algorithms. The* adaptive sequential composition $\mathcal{M}_k \circ \dots \circ \mathcal{M}_1 : \mathbf{x} \mapsto (y_1, \dots, y_k)$ *is the following sequence of algorithms: $y_1 = \mathcal{M}_1(\mathbf{x}), y_2 = \mathcal{M}_2(\mathbf{x}, z_2), \dots, y_k = \mathcal{M}_k(\mathbf{x}, z_k)$, where the DP guarantees always need to hold for the first argument, and z_i is an auxiliary input, such as the output from all the previous mechanisms.*

The most elemental composition bound is the *basic composition* bound, which states that the total epsilons and deltas add up, when running several DP algorithms.

Theorem 24 (Basic composition, Dwork and Lei 2009; Dwork and Roth 2014). *Assume a randomised algorithm $\mathcal{M}_j$ is $(\varepsilon_j, \delta_j)$-DP, with $j = 1, \dots, k$. Then the sequence $\mathcal{M}_k \circ \dots \circ \mathcal{M}_1$ is (ε', δ')-DP with $\varepsilon' = \sum_{i=1}^{k} \varepsilon_i$, and $\delta' = \sum_{i=1}^{k} \delta_i$.*

Proof. See Dwork and Roth (2014, Appendix B.1) for a proof. □

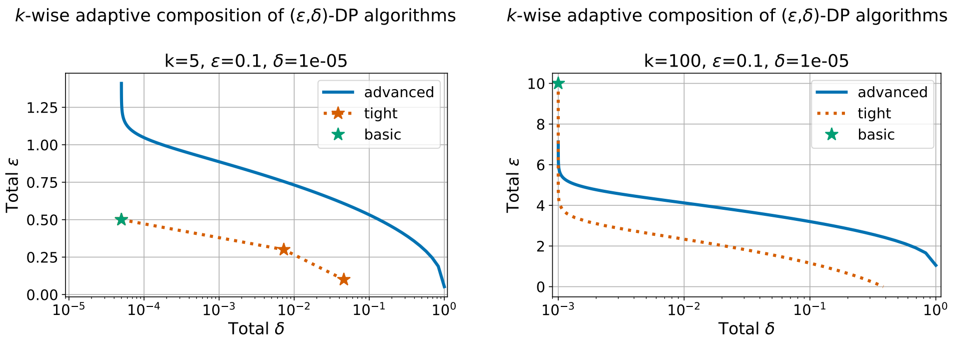
For composing k arbitrary pure ε_i-DP mechanisms, $i = 1, \dots, k$, the basic composition bound in Theorem 24 cannot be improved upon. However, for ADP and other relaxations the case is more interesting. For example, with ADP, by letting the total δ degrade somewhat, it is possible to improve the resulting total ε significantly compared to ε' in Theorem 24. Theorem 25 states the bound commonly known as the advanced composition.

Theorem 25 (Advanced composition, Dwork et al. 2010b). *The adaptive composition $\mathcal{M}_k \circ \dots \circ \mathcal{M}_1$ with privacy parameters $\varepsilon_i = \varepsilon, \delta_i = \delta, i = 1, \dots, k$ is $(\varepsilon', k\delta + \delta')$-DP under adaptive composition for any $\delta' > 0$, and $\varepsilon' = \varepsilon\sqrt{2k \log(1/\delta')} + k\varepsilon(\mathrm{e}^{\varepsilon} - 1)$.*

Proof. See e.g. Dwork and Roth (2014, Theorem 3.20). □

The result in Theorem 25 can also be extended to a varying sequence of $\varepsilon_i, \delta_i, i = 1, \dots, k$ (Kairouz et al., 2015; Rogers et al., 2016). Next, we state the optimal tight bound for the adaptive composition. This bound was first shown by Kairouz et al. (2015) for the case of having the same privacy parameters for all k algorithms, and later expanded to the case of varying parameters by Murtagh and Vadhan (2016). Murtagh and Vadhan also showed that the computational complexity of using their tight bound for a sequence of varying parameters is $\#P$-complete.[4]

[4] Complexity class $\#P$ is the class of counting problems associated with the decision problems in NP.



(a) (b)

Figure 2.1: Comparison of basic (Theorem 24), advanced (Theorem 25), and tight composition (Theorem 26) with $\varepsilon_i = \varepsilon, \delta_i = \delta$ $\forall i$: in a) with only 5 compositions basic composition is superior to advanced composition for much of the total δ range, while in b) with 100 compositions, advanced composition results in clearly better ε even with only a small increase in total δ. Comparing advanced and tight composition makes it clear that the advanced composition always has some slack in the parameters. Note that the bounds from the basic composition are tight if one is not willing to increase total δ.

Theorem 26 (Tight composition, Kairouz et al. 2015; Murtagh and Vadhan 2016)**.** *The adaptive composition $\mathcal{M}_k \circ \cdots \circ \mathcal{M}_1$ with privacy parameters $\varepsilon_j = \varepsilon, \delta_j = \delta, j = 1, \ldots, k$ is*

$$(\varepsilon'_i, 1 - (1-\delta)^k(1-\delta'_i))\text{-}DP$$

for all $i = 0, 1, \ldots, \lfloor k/2 \rfloor$, where $\varepsilon'_i = (k-2i)\varepsilon$, and $\delta'_i = \frac{\sum_{l=0}^{i-1}\binom{k}{l}(\mathrm{e}^{(k-l)\varepsilon} - \mathrm{e}^{(k-2i+l)\varepsilon})}{(1+\mathrm{e}^{\varepsilon})^k}$.

Proof. See Kairouz et al. (2015, Theorem 3.3), and Murtagh and Vadhan (2016, Theorem 3.8). □

Figure 2.1 shows a comparison of the bounds resulting from using Theorems 24, 25, & 26.

Next, we state the main composition results for composing ADP algorithms via the dominating pairs formulation (see Definition 10:

Theorem 27 (Adaptive composition of dominating pairs, Zhu et al. 2022)**.** *Assume the pair (P_i, Q_i) dominates $\mathcal{M}_i$,, $i = 1, \ldots, k$. Then $(P_1 \times \cdots \times P_k, Q_1 \times \cdots \times Q_k)$ dominates the (adaptively) composed mechanism $\mathcal{M}_k \circ \cdots \circ \mathcal{M}_1$.*

Proof. The result for two mechanism was shown by Zhu et al. (2022, Theorem 10), the results for k mechanisms follows immediately by induction. □

In Publication III, we construct dominating pairs of distributions for some common DP algorithms in the shuffle DP setting (see Section 2.6). Due to Theorems 27 & 21, this enables tighter numerical privacy accounting for compositions, including with subsampling, than the previously existing methods.

Finally, as noted earlier, one main reason for considering RDP instead of ADP is the ease of composing RDP mechanisms:

Theorem 28 (RDP composition, Mironov 2017)**.** *Let $\mathcal{M}_i$ be (α, ε_i)-RDP algorithms, $i = 1, \dots, k$. Then the composition $\mathcal{M}_k \circ \dots \circ \mathcal{M}_1$ is $(\alpha, \sum_i \varepsilon_i)$-RDP.*

Proof. The case for two mechanisms was shown by Mironov (2017, Proposition 1), the result for k mechanisms follows immediately by induction. □

We use Theorem 28 in Publication II for privacy accounting in constructing a DP Markov chain (see Section 3.1.2).

While the results in Theorems 28 & 27 hold generally, to improve on the tight bound in Theorem 26, we need some concrete mechanisms that do not match the worst-case covered by the more general results.

2.5.2 Composing specific DP algorithms: the Gaussian mechanism

As noted before, the composition result in Theorem 26 is tight for arbitrary (ε, δ)-DP mechanisms. To improve on this bound, we need to make more specific assumptions about the DP mechanisms.

In this section, we state some improved composition results for the Gaussian mechanism (see Definition 16), which is one of the most commonly used approaches to DP learning. Instead of looking only at single (ε, δ)-pairs for each mechanism, we now change the perspective to consider the privacy profiles $\{(\varepsilon, \delta(\varepsilon))\}$, which allows for more fine-grained analysis.

As noted in Section 2.3, a single query answer released via the Gaussian mechanism without privacy amplification is exactly characterised by Theorem 18 for ADP, as well as by Theorem 19 for RDP. For composing Gaussian mechanisms without subsampling, tight bounds can also be easily stated. The following Theorem 29 gives tight bounds in ADP:

Theorem 29 (Gaussian composition for ADP)**.** *Let $\mathcal{M}_i$ be Gaussian mechanisms, with ℓ_2-sensitivity Δ_i and variance $\sigma_i^2, i = 1, \dots, k$. Then the adaptive* *composition $\mathcal{M}_k, \circ \cdots \circ \mathcal{M}_1$ is (ε, δ)-DP for $\varepsilon > 0$, $\delta \in [0, 1]$ iff*

$$\delta(\varepsilon) \geq \Phi \left[\frac{-\varepsilon + \mu}{\sqrt{2\mu}} \right] - \mathrm{e}^{\varepsilon} \Phi \left[\frac{-\varepsilon - \mu}{\sqrt{2\mu}} \right],$$

where Φ is the standard normal cdf and $\mu = \sum_{i=1}^{k} \frac{\Delta_i^2}{2\sigma_i^2}$.

Proof. See e.g. Räisä et al. (2021, Theorem 2.4). □

A bound for composing Gaussian mechanisms without subsampling is also readily available with RDP.

Theorem 30 (Gaussian composition for RDP, Mironov 2017)**.** *Let $\mathcal{M}_i, i = 1, \dots, k$ be Gaussian mechanisms, with the ith mechanism having ℓ_2-sensitivity Δ_i and variance σ_i^2. Then the adaptive composition $\mathcal{M}_k, \circ \cdots \circ \mathcal{M}_1$ is $(\alpha, \alpha \sum_i \frac{\Delta_i^2}{2\sigma_i^2})$-RDP.*

Proof. Follows directly from Theorems 19 & 28. □

For the more complex case of subsampled Gaussian mechanism, privacy bounds are usually established by numerical accounting, based typically on ADP via the privacy loss formulation (essentially, it can be shown that the composition amounts to convolving the privacy loss distribution, which is the density or the probability mass function of the privacy loss random variable in Definition 6, see e.g. Meiser and Mohammadi 2018; Koskela et al. 2020a; Gopi et al. 2021; Doroshenko et al. 2022) or on RDP (Mironov et al., 2019).

Numerical accounting usually allows for finding privacy parameters that are tight up to a given numerical precision for the DP mechanisms in question. For ADP, this results in practically tight guarantees, while for RDP, the common practice of reporting the privacy in ADP means that the results will still be lossy due to the lossy conversion (Zhu et al., 2022).

We use the numerical accounting approach of Koskela et al. (2020a, 2021) for quantifying the privacy bounds for the subsampled Gaussian mechanism in Publication IV, and for *kRR* (Definition 11) in Publication III in the shuffle mode of DP (Section 2.6).

2.6 From centralised to distributed DP

The DP definitions discussed in Sections 2 & 2.2 are based on the assumption that the entire data set is available for a trusted data curator who can enforce DP. There are many settings where this is not true, but instead the samples are distributed among several parties.

Common distributed data examples include user data on mobile devices or data held, for example, by individual hospitals or health service providers. There are several variations of DP definitions that suit such settings, with differing assumptions on the level of trust, on the exact data partitioning, and on the available secure primitives.

A useful baseline with the lowest-level of trust for distributed DP is given by *local DP* (LDP, Kasiviswanathan et al. 2011), which essentially requires that the distributed data sets can only be accessed via independent local DP mechanisms:

Definition 31 (Local DP, Kasiviswanathan et al. 2011)**.** *Assume M parties, where party j has a local data set $\mathbf{x}_j$ and access to a DP algorithm $\mathcal{M}_j$. $\mathcal{M}_j$ is a* local randomiser*, if it only accesses the local data $\mathbf{x}_j$ at party j, and is independent of any other parties. DP algorithms which only access the local data sets via the local randomisers are said to guarantee* LDP*.*

Unless mentioned otherwise, we assume that each individual is present in only one of the local data sets in Definition 31. More complex cases can be handled, e.g., by composition theorems (see Section 2.5).

The following result is immediate from the definition of LDP:

Corollary 32. *Assume a distributed randomised algorithm $\mathcal{M}$ is LDP. Assume that the local randomiser $\mathcal{M}_j$ guarantees $(\varepsilon_j, \delta_j)$-DP for the full protocol run. Then $\mathcal{M}$ is (ε', δ')-LDP, where $\varepsilon' = \max\{\varepsilon_1, \dots, \varepsilon_M\}, \delta' = \max\{\delta_1, \dots, \delta_M\}$.*

Proof. This follows immediately by the so-called parallel composition, since the local data sets are disjoint: for a given local randomiser $\mathcal{M}_j$, the workings of the distributed protocol $\mathcal{M}$ can be seen as presenting queries to $\mathcal{M}_j$ and post-processing them. Since for any j, $\mathcal{M}_j$ is $(\varepsilon_j, \delta_j)$-LDP after accounting for the full distributed protocol run, and $\varepsilon_j \leq \varepsilon', \delta_j \leq \delta' \ \forall j$, it follows that $\mathcal{M}$ is (ε', δ')-LDP due to post-processing immunity (see Theorem 3). □

LDP is a very strong privacy notion that requires the absolute minimum of trust: the only thing necessary is that each party can trust that their own local randomiser satisfies DP. The downside is that the noise level needed to guarantee DP is a lot higher compared to the centralised DP setting:

***Example* 33.** *Using the Laplace mechanism for releasing a sum query with data $\mathbf{x}_j \in [0, 1], j = 1, \dots, M$, the query answer will be $\sum_{j=1}^{M} \mathbf{x}_j + \xi$, where*

$\xi \sim Laplace(0, \frac{\Delta}{\varepsilon})$, so the variance in the centralised setting is $\sigma^2 = \frac{2}{\varepsilon^2}$ since the query sensitivity $\Delta = 1$. In contrast, with LDP and M parties the same sum query can be answered by $\sum_{j=1}^{M}(\mathbf{x}_j + \xi_j)$, where $\xi_j \sim Laplace(0, \frac{\Delta}{\varepsilon}) \forall j$, so the total noise variance will be $\sum_{j=1}^{M} \frac{2}{\varepsilon^2} = M\sigma^2$.

More generally, there are known separation results between the centralised model and the local model used in LDP (see e.g. Kasiviswanathan et al. 2011). To improve on the basic LDP noise level, we need to limit the amount of information the adversary will have, either by assumption or, for example, by leveraging secure primitives. In the following, we generally assume that the data are always *horizontally partitioned*, meaning that all samples have the same features, regardless of who they belong to.[5]

Instead of general distributed learning, we are mostly interested in the *federated learning* (FL) setting introduced by McMahan et al. (2017): we assume there are M parties, often called clients, with client j holding n_j samples, and that all the clients are connected to a central server, which controls the learning protocols. The aim is to learn a single joint model from all clients' data while keeping the actual data distributed.[6]

Depending on the number and capabilities of the clients, it is common to distinguish between cross-device and cross-silo cases (Kairouz et al. 2021b). In the *cross-device* case there can be millions of clients, each with a (very) limited amount of data corresponding to a single individual, and the clients can drop out in the middle of the learning protocol. In contrast, in the *cross-silo* case the number of clients tends to be moderate at most, the clients are more persistent, and each client usually has some amount of samples. Additionally, the data held by a single client usually comes from several individuals.

Considering the granularity of the DP protection, besides the standard individual-level neighbourhood granularity, other proposed neighbourhoods include sample-level neighbourhood (also called example-level, item-level or event-level, see e.g. Heikkilä et al. 2020; Wei et al. 2020; Liu et al. 2022), element-level (which can interpolate between sample-level and user-level depending on the chosen parameters, see Asi et al. 2019), user-level (Dwork et al., 2010a; McMahan et al., 2018; Andrew et al., 2021), and client-level (Geyer et al., 2017; Truex et al., 2019, 2020; Kim et al., 2021).

[5] This is in contrast to vertical partitioning, where different features of data on a given individual are held by different parties (Mohammed et al., 2014; Tajeddine et al., 2020).

[6] We note that there has been plenty of research done on distributed DP before the introduction of federated learning. Distributed DP in a collaborative multiparty setting was first proposed by Dwork et al. (2006a). Goryczka and Xiong (2017) give a nice overview of the most relevant earlier literature on distributed DP.

Changing the granularity does not change the basic DP definition (as in Definition 1). However, explicitly changing the neighbourhood helps to avoid confusion about the DP protection granularity. It also allows us to consider different levels of DP protection at the same time: a DP FL protocol might, for example, satisfy sample-level, element-level, user-level, and client-level DP all at the same time with differing privacy parameters.

While we can often choose the granularity as we wish, it is generally better to try and push the randomisation to the lowest possible level in the protocol, since this guarantees having at least some noise in the results even if an adversary gains access to higher level information.

In DP FL, we often need to be more explicit about the view of the assumed adversary than in the centralised setting, especially with respect to privacy amplification results:

***Example* 34.** *Assume the clients communicate with the server in a FL protocol using secure channels. If we do subsampling on the client level and only the chosen clients communicate with the server, then an adversary who can simply observe who communicates has no additional uncertainty from the subsampling, even if the encryption scheme is information theoretically secure.*

Instead of simply defining a single adversary, a more solid approach would be to consider various adversaries with differing capabilities at the same time. Ideally, we would want to have a layered view of the DP guarantees, where the guarantees erode in a controlled manner when we allow the adversary more and more detailed view to the protocol; starting from a complete outsider adversary, who can only observe the final output of the protocol, and ending with a fully powerful adversary, who can, e.g., choose the data except for a single element on each iteration, and can directly observe the outputs from any DP randomisers in the protocol.

Unfortunately, a full layered security and privacy analysis is typically very hard to do in practice. The current practice therefore is to simply consider a single adversary.

The layered view to DP protocols is tightly related to the ideas of empirical DP guarantees, membership inference attack and adversary instantiation (Erlingsson et al., 2019b; Nasr et al., 2021; Watson et al., 2022). The aim in adversary instantiation is to establish empirical bounds for the privacy protection by instantiating various adversaries and measuring how successful actual membership inference attacks are.

Compared to the worst-case upper bounds derived from DP theory, these attacks provide a corresponding lower bound. When the bounds meet, we know that the theoretical privacy analysis cannot be improved without limiting the adversary further. In case the attack can be shown to be optimal, the empirical bounds could also be directly used as a privacy protection measure.

2.6.1 DP with secure aggregation

As mentioned before, to improve on the LDP guarantees, we need to limit the amount of information the adversary has. A common approach for achieving this is to rely on suitable secure primitives (see e.g. Lindell and Pinkas 2009). The basic idea with secure primitives is to calculate a given function while guaranteeing that the adversary only learns the final result. Compared to DP, this is an orthogonal direction that can be beneficial for DP guarantees: secure primitives guarantee that the adversary only learns the final result, while DP guarantees that the final result does not leak too much sensitive information.

Since general multiparty computation (MPC) can be used to compute basically any distributed functionality (Yao, 1982; Lindell and Pinkas, 2009), in principle we can simply use a general MPC protocol to implement any DP algorithm $\mathcal{M}$ in a given distributed setting, including in FL. The bottleneck is the scalability: for many tasks, the general approach is not efficient enough to be practical.[7] Depending on the problem, however, there might be more efficient protocols available.

Assume that answering sum queries of the form $\sum_{j=1}^{M} f_j(\mathbf{x}_j)$ for some functions f_j, where $\mathbf{x}_j$ is the local data for client j, are sufficient for learning (see e.g. Blum et al. 2005 for a discussion on the power of the sum query framework). Then we can use secure aggregation to guarantee that the adversary can only observe the final sum.

An *additively homomorphic encryption* (AHE) protocol (see e.g. Lindell and Pinkas 2009) is defined with some discrete ring as the message space, typically taken to be the group of integers with modulo m addition, denoted by $\mathbb{Z}_m$, and two efficient algorithms:

1. $+_{pk}$, which takes as input a public key and two ciphertexts $E_{pk}(m_1), E_{pk}(m_2) \in \mathbb{Z}_m$, and outputs $E_{pk}(m_1) +_{pk} E_{pk}(m_2) = E_{pk}(m_1 + m_2)$.

2. $\cdot_{pk}$, which takes as input the public key, a ciphertext $E_{pk}(m)$, and a constant $c \in \mathbb{Z}_m$, and outputs $c \cdot_{pk} E_{pk}(m) = E_{pk}(c \cdot m)$.

[7] However, Jayaraman et al. (2018) show that, for example, learning a fairly large DP linear regression model, where also the randomness for DP is generated in a distributed manner using an MPC protocol, can be done efficiently enough. The general idea of DP via distributed noise generation was originally proposed by Dwork et al. (2006a).

Using a suitable AHE (e.g. Paillier 1999), the clients can guarantee that the server can only decrypt the final result $\sum_{j=1}^{M} E_{pk}(f_j(\mathbf{x}_j)), f_j(\mathbf{x}_j) \in \mathbb{Z}_m$ for all j.

A similar construction can be done using additive *secret sharing* (Shamir, 1979; Boneh and Shoup, 2023). A t-out-of-N secret sharing scheme over $\mathbb{Z}_m$ is a pair of efficient algorithms:

1. A probabilistic sharing algorithm G, which takes as input parameters N, t and a secret $m \in \mathbb{Z}_m$, and outputs shares $(\zeta_1, \dots, \zeta_N)$, a t-out-of-N sharing of m.

2. A deterministic combining algorithm C, which takes as input a subset of the shares $\{\zeta_i\}_{i \in I}, I \subset \{1, \dots, N\}, |I| = t$, and outputs the reconstructed secret m.

The main idea in a t-out-of-N secret sharing scheme is that the secret m can be perfectly reconstructed with any combination of t shares, while any combination of less than t shares reveals no information about m.

In Publication I, we introduce a simple N-out-of-N additive secret sharing variant for a setting where there are several independent servers available: each server receives a share of a secret from each client, and the total sum can be reconstructed only by combining all the shares. Concurrently with Publication I, Corrigan-Gibbs and Boneh (2017) published essentially the same encryption scheme (besides the encryption, Corrigan-Gibbs and Boneh 2017 also introduced novel zero-knowledge proofs).

To combine AHE or additive secret sharing with DP, one often used idea is that each client adds a small amount of noise from some infinitely divisible family of distributions (such as the Gaussian family) and when the contributions from all the clients are summed, the aggregated noise will have the required variance for the chosen DP guarantees.

With AHE, assuming all honest clients for simplicity and writing ξ for the noise that is sufficient to guarantee DP for the full sum query, each client j adds noise ξ_j to their own query and encrypts it:

$$\sum_{j=1}^{M} E_{pk}(f_j(\mathbf{x}_j) + \xi_j) = E_{pk}(\sum_{j=1}^{M} f_j(\mathbf{x}_j) + \xi), \tag{2.22}$$

where $\xi = \sum_{j=1}^{M} \xi_j$. Decrypting the final sum, the server is left with the required DP sum, which in this case has exactly the same amount of noise as required in the centralised setting. Unlike in the LDP setting, the DP guarantees now depend on all the clients, since each noise shard is required for the correct variance.

This basic idea of combining secure summation and DP was first introduced by Rastogi and Nath (2010), and has been subsequently reinvented or borrowed in several papers (e.g. Heikkilä et al. 2020; Wei et al. 2020, see Goryczka and Xiong 2017 for a survey of related techniques in a more general distributed DP setting), including in Publications I and IV.

While this approach works with an ideal trusted aggregator summing true reals, as pointed out, for example, by Kairouz et al. (2021a), implementing it on finite computers runs into problems with continuous noise values: the distribution of the noise ξ after summation is not generally guaranteed to be in the same noise family anymore (see also related discussion in Section 2.1 on vulnerabilities caused by standard floating-point representation).

Several discrete noise distributions which do not suffer from this vulnerability have been proposed, which have been increasingly close to the continuous Gaussian in terms of utility (Agarwal et al., 2018; Canonne et al., 2020; Kairouz et al., 2021a; Agarwal et al., 2021; Chen et al., 2022; Chaudhuri et al., 2022).

Another potential problem with the above scheme of using AHE or secret sharing for DP learning is potential vulnerability to attacks such as model poisoning or backdooring with more malicious clients: since the server can only observe the final noisy sum, malicious clients can try to do model poisoning or create backdoors without being detected.[8]

While there are methods which try to make the models provably more robust against such attacks (for example, certified robustness, introduced by Lécuyer et al. 2019, leverages DP to formally guarantee that small perturbations do not change the model's predictions) or which can guarantee that the input has the expected form (see e.g. Corrigan-Gibbs and Boneh 2017; Sabater et al. 2022 that use scalable zero-knowledge proofs), they also induce additional costs, either in terms of decreased model utility or in increased computational and communications requirements.

2.6.2 Shuffle model of DP

Another secure primitive useful for DP FL is secure shuffling (Chaum, 1981). A secure shuffler takes one or more messages from each client as input, and after receiving all messages, outputs a uniformly random permutation of the inputs. This basic functionality is the basis for shuffle DP, which was first formalised by Cheu et al. (2019) (see also Bittau et al. 2017).

Assuming M clients in total, denote the local randomiser used by client j by $\mathcal{R}_j : \mathcal{X} \to \mathcal{O}^{n_{msg}}$, where n_{msg} is the number of messages the client will

[8] See Lin et al. 2021 for a survey of proposed attacks against machine learning models.

send to the shuffler. We usually assume that $\mathcal{R}_j = \mathcal{R} \; \forall j$. The shuffler is a randomised mapping

$$\mathcal{S} : \mathcal{O}^{M \times n_{msg}} \to \mathcal{O}^{M \times n_{msg}}, \mathcal{S}(\mathbf{x}_1, \ldots, \mathbf{x}_{Mn_{msg}}) = (\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(Mn_{msg})}),$$

where π is a uniformly random permutation.

For privacy, we require that $S(\cup_{j=1}^{M} \mathcal{R}(x_j))$ is DP. As with secure aggregation and in contrast to LDP, in shuffle DP the privacy guarantees are again joint guarantees on all the clients, since they depend jointly on the local randomisers in addition to the shuffler.

Comparing shuffle DP with LDP and centralised DP, since shuffle DP gives out more information than centralised DP but has an additional layer of uncertainty compared to LDP, we would intuitively expect that it will sit between LDP and centralised DP in capability. This basic intuition turns out to be right, as established by separation results for the single-message ($n_{msg} = 1$) model (Cheu et al., 2019).

There are also known separation results in the shuffle model between single-message and multi-message ($n_{msg} > 1$) models (see e.g. Balle et al. 2019; Cheu 2021; Ghazi et al. 2021a, 2020, 2021b). Various versions of interactive shuffle model have also been considered, which can further strengthen the model. A fully interactive multi-message shuffle DP has been shown to be powerful enough under some assumptions to simulate any randomised algorithm defined in the centralised DP model (see Cheu 2021 for a good discussion on known separation results and various forms of interactivity in shuffle DP).

Another perspective to the shuffle DP model is via privacy amplification: under some assumptions, the privacy of a given LDP protocol can be amplified by adding a shuffler (Erlingsson et al., 2019a; Balle et al., 2019, 2020b; Feldman et al., 2021, 2022; Girgis et al., 2021b,a). In effect, finding the optimal privacy amplification means deriving tight DP bounds in the shuffle DP model (compare this to the discussion on privacy amplification by subsampling in Section 2.4).

In Publication III, we construct dominating pairs of distributions (see Definition 10) for various local randomisers in the shuffle DP model. We derive such pairs for general pure LDP randomisers based on the results of Feldman et al. (2021), as well as for k-randomised response (see Definition 11) under varying adversaries, expanding on the work of Balle et al. (2019). The dominating pairs can then be directly used with any numerical accounting method (see Section 2.5), to find tighter privacy bounds than was possible with the previously existing methods for adaptive sequential compositions in the shuffle DP model.

Chapter 3

Privacy-preserving Bayesian learning

In this section, we first review the basics of Bayesian learning, starting with the high-level objective of learning a posterior distribution in Section 3.1. In Section 3.1.1, we discuss exact Bayesian learning and exponential family distributions, and then continue with approximate Bayesian learning based on Markov chains in Section 3.1.2, and on variational inference in Section 3.1.3. Finally, in Section 3.2 we discuss the main approaches to DP Bayesian learning.

3.1 Bayesian learning

In Bayesian learning we are generally interested in learning a *posterior distribution* on some quantity of interest $\theta \in \Theta$, given some *data* $\mathbf{x} \in \mathcal{X}$, a *prior distribution* $p(\theta)$, and a *likelihood* $p(\mathbf{x}|\theta)$.[1] By Bayes' theorem the posterior distribution can be written as

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \tag{3.1}$$

$$\propto p(\mathbf{x}|\theta)p(\theta), \tag{3.2}$$

where $p(\mathbf{x})$ is the normalizing constant, often also called the partition function, the marginal likelihood, or the evidence of the model. It is important, for example, in model selection problems (see, e.g., MacKay 2003).

The posterior distribution $p(\theta|\mathbf{x})$ describes our beliefs about θ in light of the prior and the data, and can be used for making predictions for future

[1] See e.g. Bernardo and Smith (1994) for a nice general overview of Bayesian theory.

observations $\hat{x}$ via the *predictive distribution*

$$p(\hat{\mathbf{x}}|\mathbf{x}) = \int_\theta p(\hat{\mathbf{x}}|\theta)p(\theta|\mathbf{x})d\theta. \tag{3.3}$$

By privacy-preserving Bayesian learning, we essentially mean learning the posterior $p(\theta|\mathbf{x})$ under DP.

3.1.1 Conjugate-exponential family of distributions

The Bayesian learning approach as described so far is very straightforward: when seeing some new data, we update our beliefs via Equation (3.1), and make predictions via Equation (3.3) when necessary.

The main difficulty is typically computational: the posterior is usually an intractable distribution that is hard to work with. However, for an important if restricted class of problems, namely, for conjugate-exponential family models, the posterior has a closed-form expression and is therefore generally a tractable distribution.

Exponential family distributions have several nice properties (see, for example, Casella and Berger 2001; Bernardo and Smith 1994), the most important for our purposes being that they have *conjugate priors*, which allows for solving the posterior analytically, and that they always have finite *sufficient statistics*.

Before discussing the exponential family further, we first define a sufficient statistic:

Definition 35 (Sufficient statistic, Jordan 2009). *A function $T : \mathcal{X} \to \mathbb{R}^d$ is called a* sufficient statistic *for* $\mathbf{x}$, *if*

$$p(\theta|T(\mathbf{x})) = p(\theta|T(\mathbf{x}), \mathbf{x}). \tag{3.4}$$

Additionally, when $T(\mathbf{x}), T'(\mathbf{x})$ are sufficient statistics, if there always exists a function f s.t. $T(\mathbf{x}) = f(T'(\mathbf{x}))$, then $T(\mathbf{x})$ is called a minimal sufficient statistic*.*

The main point in Definition 35 is that a sufficient statistic contains all the information from the data that is needed for fitting a model. [2]

It turns out that the family of models which have finite sufficient statistics with any data set size is the exponential family:

[2] Note that there are other equivalent definitions of sufficiency, see e.g. Bernardo and Smith (1994); Casella and Berger (2001); Wasserman (2004). Our chosen definition is one of the more Bayesian-friendly approaches, since we tend to think of the parameter θ as random in any case.

Definition 36 (Exponential family)**.** *A probability distribution belongs to an* exponential family*, if it can be written as*

$$p(\mathbf{x}|\theta) = h(\mathbf{x}) \exp\left(\eta(\theta)^T T(\mathbf{x}) - A(\eta(\theta))\right), \tag{3.5}$$

where $\theta \in \Theta \subseteq \mathbb{R}^d$, $h : \mathcal{X} \to \mathbb{R}$, $T(\mathbf{x})$ is a sufficient statistic, and $A(\eta(\theta))$ is called the cumulant function or the log-partition function.

If $\eta_i(\theta) = \eta_i \forall i$ in Definition 36, then the family is said to be in *canonical form*, and the parameters are called *natural parameters.* A given exponential family can always be transformed into canonical form by a suitable transformation (see, e.g., Casella and Berger 2001).

In addition, the parameterisation is said to be *minimal*, when there are no linear constraints between the components of the parameter vector nor (almost surely) between the components of the sufficient statistic. The space $\mathcal{H} = \{\eta : \int h(\mathbf{x}) \exp\left(\eta(\theta)^T T(\mathbf{x})\right) d\mathbf{x} < \infty\}$ is called the *natural parameter space.* When $\mathcal{H}$ is a non-empty open set, the exponential family is called *regular.* In this thesis, we focus on regular exponential families.

Finally, exponential family distributions have conjugate priors, that allow for writing the posterior in a closed-form:

Definition 37 (Conjugacy)**.** *Let $p(\theta|\lambda_0)$ be an exponential family prior, and $p(\mathbf{x}|\theta)$ a likelihood function. If the posterior $p(\theta|\mathbf{x}, \lambda_0) \propto p(\mathbf{x}|\theta)p(\theta|\lambda_0)$ is in the same exponential family as the prior, then the likelihood and the prior are* conjugate distributions*.*

***Example* 38** (Conjugate-exponential family)**.** *Let $X \sim \text{Bin}(n, \theta)$, where $\theta \in (0, 1)$ is the interesting parameter. We immediately have*

$$\begin{aligned} p(x|\theta) &= \binom{n}{x} \theta^x (1-\theta)^{n-x} && (3.6)\\ &= \binom{n}{x} \left(\frac{\theta}{1-\theta}\right)^x (1-\theta)^n && (3.7)\\ &= \binom{n}{x} \exp\left\{\log\left(\frac{\theta}{1-\theta}\right) \cdot x + n\log(1-\theta)\right\}. && (3.8) \end{aligned}$$

Substituting $\eta = \log\left(\frac{\theta}{1-\theta}\right)$ above gives

$$p(x|\eta) = \binom{n}{x} \exp\{\eta \cdot x - n\log(1 + \exp(\eta))\}, \tag{3.9}$$

which is clearly an exponential family distribution with sufficient statistic $T(x) = x$, natural parameter $\eta(\theta) = \log\left(\frac{\theta}{1-\theta}\right)$, and log-partition function $A(\eta) = n\log(1 + \exp(\eta))$.

Considering now setting a prior on θ s.t. the prior and the Binomial likelihood are conjugate, Beta distribution has the correct functional form with density

$$p(\theta|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{3.10}$$

where α, β are prior parameters, and Γ is the gamma-function. The posterior can now be solved analytically:

$$\begin{aligned} p(\theta|x,\alpha,\beta) &\propto p(x|\theta)p(\theta|\alpha,\beta) && (3.11)\\ &\propto \theta^{\alpha+x-1}(1-\theta)^{\beta+n-x-1}, && (3.12) \end{aligned}$$

which is another Beta distribution with parameters α', β' given by

$$\alpha' = \alpha + x, \qquad \beta' = \beta + n - x. \tag{3.13}$$

We need the exponential family theory in Publications I and IV: a finite sufficient statistic allows for an efficient approach to enforcing privacy, based on perturbing the sufficient statistic (see Section 3.2 for a longer discussion on sufficient statistic perturbation). Additionally, in Publication IV we show that in the conjugate-exponential family, our local averaging approach to partitioned variational inference (see Section 3.1.3) does not change the resulting local approximation.

While the conjugate-exponential family contains many important and practically relevant distributions, in most cases the posterior distribution does not have a nice analytical form. We can still work with such distributions, e.g., by using computational methods to approximate the true posterior. We will next focus on the two most common approximation methods: Markov chain Monte Carlo (MCMC, Section 3.1.2) and variational inference (VI, Section 3.1.3).

3.1.2 Markov chain Monte Carlo

The main idea in MCMC is to draw samples from a distribution that is not tractable. In our case, we want to approximate an intractable posterior by drawing samples. These samples can then be used to estimate various quantities of interest, such as the mean or the variance of the true posterior.

We first discuss some basic properties of Markov chains and then move on to introduce the main variants of MCMC used in the dissertation. We refer to Robert and Casella (2004); Meyn and Tweedie (2005) for good general presentations of Markov chain and MCMC theory.

Markov chains

A Markov chain is a random process evolving in (discrete) time. The process is controlled by a *transition kernel*. We take the transition kernel K to be a conditional probability distribution on the state space Θ, giving the probabilities for the next state given the current one.

The defining property of Markov chains is their limited memory: the next state only depends on the most recent value, and knowing the full history beyond this point brings no additional benefit. In this thesis, we only need time-homogeneous chains, for which the transition probabilities do not change over time:

$$\mathbb{P}(Z_{t+m} \in A | z_1, \ldots, z_t) = \mathbb{P}(Z_{t+m} \in A | z_t) = K^m(z_t, A) \ \forall t, m > 0, \quad (3.14)$$

where we write $K^m(z_t, A)$ for the probability that the chain moves from z_t to A in m steps, and writing $\mathcal{B}$ for the Borel σ-algebra, we have $A \in \mathcal{B}(\Theta)$, and $z_i \in \Theta \ \forall i$.

To guarantee that a Markov chain will eventually converge to our chosen target distribution, two properties are sufficient: detailed balance and ergodicity. Detailed balance guarantees that the chain has a (correct) stationary distribution, and ergodicity essentially means that the initial state does not matter for the convergence.

For ergodicity, the Markov kernels considered in this thesis have the *strong irreducibility* property: any positive measure set $A \in \mathcal{B}(\Theta)$ can be reached from any other point in the parameter space in a single step.

As mentioned, the second condition we need is detailed balance:

Definition 39 (Detailed balance)**.** *A Markov chain with transition kernel K satisfies the* detailed balance condition*, if*

$$K(y, z)\pi(y) = K(z, y)\pi(z), y, z \in \Theta$$

where π is a probability density.

For a chain satisfying detailed balance condition (Definition 39), it can be shown that the distribution π is the invariant distribution of the chain, and the chain is time-reversible (see e.g. Robert and Casella 2004). Detailed balance together with strong irreducibility are sufficient conditions to guarantee that the chain convergences asymptotically to the invariant distribution π.

MCMC algorithms

As discussed earlier, the basic idea in MCMC is to construct a Markov chain to approximate an intractable distribution π.[3] Algorithm 2 describes a standard approach, where we construct a Markov chain by proposing the next value for the chain by a transition kernel K, and then decide if the chain moves with an acceptance test ϱ. With some assumptions, the chain can then be shown to converge asymptotically to the chosen target distribution π (see, e.g., Robert and Casella 2004).

Algorithm 2 General MCMC

```
Require: Target distribution $\pi$, Markov kernel $K$, acceptance test $\varrho$,
    number of samples $T$, initial value $\theta_0$.
for $t = 1$ to $T$ do
    Propose new value $\theta'$ from the proposal distribution $K(\theta_{t-1}, \cdot)$.
    Set $\theta_t = \theta'$ with probability $\varrho(\theta_{t-1}, \theta')$, otherwise set $\theta_t = \theta_{t-1}$.
end for
return Generated samples $\theta_1, \dots, \theta_T$.
```

While there are many options for the kernel K when using Algorithm 2, a common choice is $K(\theta, \cdot) \sim \mathcal{N}(\theta, \sigma I_d)$ with some $\sigma > 0$. When $\theta \in \mathbb{R}^d$, this guarantees that the chain is strongly irreducible, since any state can be reached from any other state in a single step.

In turn, the most common choice for the acceptance test ϱ is the Metropolis-Hastings (M-H) test:

Definition 40 (Metropolis-Hastings acceptance test, Metropolis et al. 1953; Hastings 1970)**.** *Given a target distribution π and a Markov kernel K, the* Metropolis-Hastings acceptance test *is given by*

$$\varrho_{MH}(\theta, \theta') = \min\left\{1, \frac{\pi(\theta')K(\theta', \theta)}{\pi(\theta)K(\theta, \theta')}\right\}, \tag{3.15}$$

where θ is the current state and θ' is the proposed new state for the chain.

Note that the target distribution π, is only needed up to the normalising constant to calculate the acceptance probability ϱ_{MH} in Definition 40.

While M-H is the most common acceptance test used in practice, there are other possible choices which lead to valid MCMC samplers. These are less used in standard settings, since the M-H acceptance test can be shown

[3] As also noted earlier, in the Bayesian case this would typically be the posterior: $\pi(\theta) \propto p(\mathbf{x}|\theta)p(\theta)$.

to be the most efficient choice (Peskun, 1973). One alternative choice is the Barker acceptance test:

Definition 41 (Barker acceptance, Barker 1965)**.** *Given a target distribution* π *and a Markov kernel* K*, the* Barker acceptance test *is given by*

$$\varrho_{Barker}(\theta, \theta') = \frac{\pi(\theta')K(\theta', \theta)}{\pi(\theta)K(\theta, \theta') + \pi(\theta')K(\theta', \theta)}, \tag{3.16}$$

where θ *is the current state and* θ' *is the proposed new state for the chain.*

In Publication II we use the Barker acceptance test to construct a general differentially private MCMC method. In our setting, the main advantage in using the Barker acceptance is that it has an equivalent representation via Logistic noise. Write V_{log} for a random variable with a standard logistic distribution, i.e., a random variable V_{log} has the probability density function

$$p_{V_{log}}(y) = \frac{\exp(-y)}{(1 + \exp(-y))^2}, y \in \mathbb{R}. \tag{3.17}$$

Seita et al. (2017) have shown that testing if

$$\log \frac{\pi(\theta')K(\theta', \theta)}{\pi(\theta)K(\theta, \theta')} + V_{log} > 0$$

is equivalent to the acceptance test ϱ_{Barker} in Definition 41. We show how this Logistic noise representation can be used for guaranteeing RDP for the chain without any additional perturbation mechanism. This is in contrast to most standard approaches to differential privacy, which are based on adding extra randomisation steps to learning algorithms (see Section 2.3).

3.1.3 Variational inference

In variational inference, the basic idea is to find a tractable approximating distribution to an intractable posterior by turning the original problem into an optimisation problem than can be solved (see e.g. Jordan et al. 1999; Bishop 2006 for an introduction to VI, Zhang et al. 2019 for a more recent survey).

Writing $q(\theta)$ for the approximation, we try to minimise some notion of distance between the approximation and the true posterior. Most-commonly, the distance is measured in terms of Kullback-Leibler (KL) divergence, which leads to the following optimisation problem:[4]

$$\arg\min_{q \in \mathcal{Q}} \left[D_{\mathrm{KL}}(q(\theta) \| p(\theta | \mathbf{x})) \right], \tag{3.18}$$

[4] Note that KL divergence is not symmetric, and hence is not a proper metric.

where $\mathcal{Q}$ is some tractable family of distributions, and $D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_{o\sim Q}\left[\log(\frac{q(o)}{p(o)})\right]$. The minimisation problem in Equation 3.18 is still not generally tractable. However, we can reformulate Equation 3.18 as a maximisation problem:

$$\begin{aligned}
-D_{\mathrm{KL}}(q(\theta)\|p(\theta|\mathbf{x})) &= \int q(\theta)\log\left(\frac{p(\theta)p(\mathbf{x}|\theta)}{q(\theta)p(\mathbf{x})}\right)d\theta && (3.19)\\
&= \mathbb{E}_q[\log p(\mathbf{x}|\theta)] - \int q(\theta)\left[\log\frac{q(\theta)}{p(\theta)}\right]d\theta - \mathbb{E}_q\left[\log p(\mathbf{x})\right] && (3.20)\\
&= \mathbb{E}_q[\log p(\mathbf{x}|\theta)] - D_{\mathrm{KL}}(q(\theta)\|p(\theta)) - \log p(\mathbf{x}). && (3.21)
\end{aligned}$$

Since $p(\mathbf{x})$ in Equation 3.21 does not depend on θ, it does not change the optimal approximation q. Therefore, solving the following optimisation problem is equivalent to the original problem in Equation 3.18:

$$\arg\max_{q\in\mathcal{Q}}\left[\mathbb{E}_q[\log p(\mathbf{x}|\theta)] - D_{\mathrm{KL}}(q(\theta)\|p(\theta))\right]. \qquad (3.22)$$

The utility function inside the argmax in Equation 3.22 is called the *evidence lower bound* (ELBO).

Ashman et al. (2022) consider VI in the federated learning setting, where the model is learned by a central server and the data are distributed among M clients (see Section 2.6 for more discussion on the federated learning setting), and propose a general partitioned VI (PVI) framework.

The main aim in PVI is to reduce the amount of communication rounds between the server and the clients by pushing more computations to the clients, while maintaining the same global VI solutions. This is enabled by constructing the approximation using client-specific factors, and by modifying the ELBO. The PVI approximation is defined as:

$$q(\theta) = \frac{1}{Z_q}p(\theta)\prod_{j=1}^{M}t_j(\theta) \simeq \frac{1}{Z}p(\theta)\prod_{j=1}^{M}p(\mathbf{x}_j|\theta) = p(\theta|\mathbf{x}), \qquad (3.23)$$

where Z_q, Z are normalising constants. The PVI algorithm works by iteratively updating the t-factors via purely local optimisations, and updating the global approximation by the server incorporating the local changes.

For client m doing local optimisation at global update s, the local ELBO in PVI is defined as

$$\mathbb{E}_q[\log p(\mathbf{x}_m|\theta)] - D_{\mathrm{KL}}(q(\theta)\|p^{(s)}_{\backslash m}(\theta)), \qquad (3.24)$$

where $\mathbf{x}_m$ is the local data at client m, and $p_{\setminus m}^{(s)}(\theta)$ is the *effective prior* or the *cavity distribution*:

$$p_{\setminus m}^{(s)}(\theta) = p(\theta) \prod_{j \neq m} t_j^{(s-1)}(\theta).$$

Compared to the regular ELBO in Equation 3.22, the local ELBO only uses the mth data shard, and replaces the prior with the effective prior $p_{\setminus m}^{(s)}$, which includes t-factors from the other clients.

The main idea with the client-specific t-factors is that during local optimisation at client m, the factors $t_j, j \neq m$ effectively stand in for the log-likelihood terms $p(\mathbf{x}_j|\theta)$. This allows for running the local optimisation without communicating with the other clients, while still having information about the missing log-likelihood terms to prevent the local optimisation from only finding a locally optimal solution.

In Publication IV, we propose a framework for DP federated VI based on PVI. We consider three possible implementations within the general framework, one based on perturbing the local optimisation runs (effectively, using DP-SGD for the local optimisations, see Algorithm 1) and two alternatives based on perturbing the global model updates sent by the clients with the Gaussian mechanism (Definitions 16).

3.2 DP Bayesian learning

Intuitively, there are good reasons why combining Bayesian approaches with DP might make sense. One of the main attractions in Bayesian learning generally is the principled handling of uncertainty: the main interest is the posterior, which is a distribution, not a single parameter value. Since DP is all about introducing carefully tuned uncertainty into the learning process, one could hope that this additional source of noise can be handled with little effort using the same Bayesian principles.

Unfortunately, this turns out to be a hard problem in general. On the positive side, there is already a rich literature on combining Bayesian learning with DP under various settings. The existing approaches are applicable in differing settings, and there exists no universally best method; all DP Bayesian learning methods incur some costs, most typically in terms of model utility, but in many cases also in terms of the amount of compute required, and the most suitable method depends on the setting in question.

In the rest of this section, we review the existing literature focusing on the approaches that are most significant in contextualising the publications included in this dissertation. The main approaches covered in the next sections, as well as the models where each is applicable, are listed in Table 3.1.

	Possible models	Output
Exact posterior sampling	Models where direct sampling is possible	Samples
Sufficient statistic perturbation	Exponential family only	Model parameters
MCMC	Any model	Samples
VI	Exponential family or differentiable approximating models	Model parameters

Table 3.1: Main approaches to DP Bayesian learning

3.2.1 DP via posterior sampling

Considering how one can share an arbitrary posterior distribution, one general way to summarise any distribution is to provide samples from it. Assuming for the moment that we can simply draw samples from the exact posterior distribution, Dimitrakakis et al. (2014, 2017) showed that, under some conditions, the uncertainty from the random posterior sampling without any additional noise mechanism can provide pure DP (Definition 1).

This privacy via posterior sampling was observed to be an instance of exponential mechanism (McSherry and Talwar, 2007), a standard privacy mechanism, already by Mir (2012) in connecting DP with information theory. The connection with the exponential mechanism was also used by Wang et al. (2015) (see also Dimitrakakis et al. 2017), who showed that a sample from the posterior with DP guarantees generally enjoys some nice statistical properties, such as consistency under fairly weak assumptions (see also Zhang et al. 2016).

Minami et al. (2016) introduced (ε, δ)-DP exponential mechanism for providing ADP assuming convex and Lipschitz log-likelihood functions, even when the likelihood itself can be unbounded, while Zhang et al. (2016) used posterior sampling as a method for providing DP for probabilistic graphical models.

Finally, Geumlek et al. (2017) used RDP (see Definition 8) as the chosen privacy definition, and allowed explicitly tuning either the prior strength or the likelihood temperature to guarantee the chosen privacy level even in cases where a direct posterior sample could not provide it.

In Publication II, we use a similar likelihood temperature change with very large data set sizes. Whereas in the posterior sampling approach the temperature is treated as an additional hyperparameter for tuning the privacy level (Wang et al., 2015; Geumlek et al., 2017), we instead change the temperature to improve mixing properties of the Markov chain as well as allow more free use of privacy amplification by subsampling (see Section 2.4) with large data sets.

One major limitation with the DP via posterior sampling framework is that the ability to easily draw samples from the exact posterior is limited to quite specific models, e.g., it is possible in the conjugate-exponential family or with suitable graphical models. However, when the sample instead comes from an approximate posterior, the privacy guarantees weaken with the distance from the true posterior (Wang et al., 2015; Minami et al., 2016), and measuring the distance to an unknown posterior to establish the privacy bounds is generally hard.

Another downside was pointed out by Foulds et al. (2016), based on the observation that using the exponential mechanism for posterior sampling can be alternatively viewed as increasing the model temperature depending on the privacy parameter ε (Huang and Kannan, 2012): posterior sampling via the exponential mechanism is not optimal in the asymptotic relative efficiency sense. Yet another problem follows if we consider changing the prior strength instead of changing the likelihood temperature: under some assumptions, Dimitrakakis et al. (2017) showed that when the prior is chosen to reflect the DP considerations instead of the actual prior knowledge, the posterior utility will decrease with the prior strength.

3.2.2 Sufficient statistic perturbation

One alternative approach for DP Bayesian learning is based on randomising the sufficient statistic (Dwork and Smith, 2010). As discussed in Section 3.1.1, Bayesian learning for conjugate-exponential family models can be done analytically. Since in this case the sufficient statistic contains all the information needed about the sensitive data, DP can be enforced by perturbing the sufficient statistic. This approach was first proposed by Zhang et al. (2016) in the context of probabilistic graphical models, and later generalised and analysed by Foulds et al. (2016).

In particular, Foulds et al. (2016) showed that sufficient statistic perturbation using the Laplace mechanism (see Definition 14) has the optimal asymptotic relative efficiency. On a related note, Honkela et al. (2018) showed that, under some assumptions, sufficient statistic perturbation using the Laplace mechanism can be shown to have an optimal convergence rate in the sense of being asymptotically efficiently private (AEP) as in Definition 43 below.

Definition 42 (Asymptotic consistency, Honkela et al. 2018)**.** *A differentially private algorithm $\mathcal{M}$ is* asymptotically consistent *with respect to an estimated parameter θ if the private estimates $\hat{\theta}_{\mathcal{M}}$ given a data set $\mathbf{x}$ converge in probability to the corresponding non-private estimates $\hat{\theta}_{NP}$ as the number of samples, $n = |\mathbf{x}|$, grows without bound, i.e., if for any $\alpha > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left[\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > \alpha\right] = 0. \tag{3.25}$$

Definition 43 (Asymptotically efficiently private, Honkela et al. 2018)**.** *A differentially private algorithm $\mathcal{M}$ is* asymptotically efficiently private *with respect to an estimated parameter θ, if the algorithm is asymptotically consistent and the private estimates $\hat{\theta}_{\mathcal{M}}$ converge to the corresponding non-private estimates $\hat{\theta}_{NP}$ at the rate $\mathcal{O}(1/n)$, i.e., if for any $\alpha > 0$ there exist constants C, N such that*

$$\mathbb{P}\left[\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > C/n\right] < \alpha \tag{3.26}$$

for all $n \geq N$.

Honkela et al. (2018) also showed that the convergence rate in Definition 43 is optimal for any pure DP algorithm.

In Publication I, we establish corresponding results for the Gaussian mechanism: the convergence rate in Definition 43 is also optimal for any (ε, δ)-DP algorithm, and we show that estimating the means of multivariate Gaussian observations bounded by some constant via sufficient statistic perturbation using Gaussian noise is AEP. In particular, under the same assumptions, estimating the posterior means of a Bayesian linear regression model is AEP.

On a practical level, we show how to do DP Bayesian learning from distributed data, based on the sufficient statistic perturbation with Gaussian noise for exponential family models. For more complex differentiable models, we propose to perturb the model gradients to guarantee ADP.

One problem with the naive sufficient statistic perturbation is that, as noted by Bernstein and Sheldon (2018) (see also Williams and McSherry 2010), when the perturbation is not accounted for in the Bayesian learning, the resulting DP posterior is not properly calibrated. Instead, Bernstein and Sheldon (2018) proposed to include the additional uncertainty due to

DP into the Bayesian estimation. This guarantees that the resulting DP posterior will be properly calibrated.

Unfortunately, the approach proposed by Bernstein and Sheldon (2018) works for some exponential family models, but cannot be directly applied to general models. Efficient methods that result in (nearly) properly calibrated posteriors have subsequently been proposed for Bayesian linear regression (Bernstein and Sheldon, 2019) as well as for Bayesian generalised linear regression models (Kulkarni et al., 2021).

3.2.3 Approximate posterior sampling with DP MCMC

As already noted, the ability to sample from the exact posterior distribution is limited to some special cases like the conjugate-exponential family models. To allow DP Bayesian learning more generally, we need methods for DP learning suitable for approximate posteriors. As discussed in Section 3.1, MCMC is one common approach for approximating more general posterior distributions using samples.

Wang et al. (2015) first noted that the non-private stochastic gradient Langevin dynamics (SGLD) algorithm (Welling and Teh, 2011) is essentially the same as DP-SGD (see Algorithm 1) without the per-example gradient clipping. Hence, assuming that each per-example gradient is bounded and that the step size is chosen appropriately, running SGLD will produce samples from the posterior with DP guarantees.

One major problem with the analysis of Wang et al. (2015) is the step size: since the step size depends on the privacy parameters, and their analysis uses the advanced composition (see Theorem 25), the largest possible step size allowed by the analysis tends to be too small for many practical purposes. To fix this, Li et al. (2019) re-analysed the algorithm using the RDP-based moments accountant, introduced by Abadi et al. (2016), which allows for larger step sizes.

A more general limitation in using DP-SGD for posterior sampling is the reliance on gradients: in case the model is not differentiable, gradient-based sampling is clearly not possible. Foulds et al. (2016) proposed a DP Gibbs sampler, a MCMC variant in which the parameters are updated sequentially dimension by dimension via their conditional distributions. The method is based either on perturbing the sufficient statistic via the Laplace mechanism, which is possible when all the conditional distributions are in the conjugate-exponential family, or more generally on analysing the Gibbs sampler as an exponential mechanism (McSherry and Talwar, 2007). While the more restricted sampler based on perturbing the sufficient statistic can be practical for suitable models, the more general exponential mechanism interpretation still suffers from the fact that sampling from an arbitrary distribution is generally hard even without privacy.

In Publication II, we introduce a general DP MCMC sampler based on the Barker acceptance test. As discussed in Section 3.1.2, we show that the inherent randomness in the accept-reject decision using the Barker acceptance test guarantees RDP under some assumptions. A limitation of the method is that it has a tight upper bound on the noise variance available for DP due to the logistic noise decomposition used in the acceptance test. With large data sets, however, we show how to leverage the subsampling privacy amplification effect (see Section 2.4) to enable running longer chains under tight privacy guarantees.

Concurrently with Publication II, Yıldırım and Ermiş (2019) introduced another general MCMC algorithm based on the penalty MCMC algorithm (Ceperley and Dewing, 1999), which can account for the DP noise added for the accept-reject decision by tuning the acceptance rate according to the noise level. Unlike the sampler introduced in Publication II, the method based on the penalty algorithm enables adding arbitrary amounts of noise in running the sampler. More recently, Räisä et al. (2021) proposed and analysed DP Hamiltonian Monte Carlo, that builds on the DP penalty algorithm of Yıldırım and Ermiş (2019).

3.2.4 DP variational inference

Besides MCMC, VI is another common approach for approximating posteriors using some simpler family of distributions (see Section 3.1.3). Park et al. (2020) proposed the first DP approach for VI based on sufficient statistic perturbation.

Jälkö et al. (2017) introduced DP VI for non-conjugate models, building on doubly-stochastic VI (Titsias and Lázaro-Gredilla, 2014) and on automatic differentiation VI (Kucukelbir et al., 2017). In effect, the privacy results from using DP-SGD (Algorithm 1) in optimising the variational parameters.

More recently, Vinaroz and Park (2022) proposed DP stochastic expectation propagation, which is an alternative but related approximation method to VI (Minka, 2005; Li et al., 2015). In their approach, DP is guaranteed again via sufficient statistic perturbation.

In Publication IV, we consider VI in the cross-silo federated learning setting, where the data are distributed among the clients (see Section 2.6), using the partitioned VI framework (Ashman et al. 2022, see Section 3.1.3). In this DP partitioned VI (DP-PVI) framework, we focus on models where the approximating factors are in the exponential family and consider three alternative implementations: one based on perturbing the local optimisation done by each party independently using DP-SGD (Algorithm 1), and two based on perturbing the natural parameters of the model using the Gaussian mechanism, since this general approach can be easily combined with secure primitives, such as secure aggregation or secure shuffling (see Section 2.6).

Chapter 4

Conclusion

The publications included in this dissertation introduced several novel methods for DP Bayesian machine learning. These include variants of the most common non-DP Bayesian learning methods, namely, learning in the conjugate-exponential family, MCMC, and VI. The second important topic in the publications is to consider learning from distributed data. Taken together, the methods proposed in the articles allow for DP Bayesian machine learning under various practical settings.

Considering open questions for future work, a major open problem is learning properly calibrated posteriors under DP: most of the existing methods do not include the additional uncertainty due to DP in the modelling. This results in overconfident posterior distributions. As discussed in Section 3.2.2, the currently existing solutions mainly work well for suitable exponential family models, including for Bayesian linear and generalised linear regression.

Another mostly open problem concerns optimality of learning algorithms under DP: many of the existing methods for DP Bayesian learning do not have any known optimality guarantees. When considering learning from distributed data, the question about optimality will include additionally the amount of communication necessary. While there has been progress on solving the trilemma of model utility, privacy, and communication (Chen et al., 2023), in many cases this is still an open problem.

Finally, a largely orthogonal research direction involves privacy accounting: considering a given DP Bayesian learning method, any improvement on the privacy accounting translates directly into requiring less randomness in the learning algorithm while keeping the same privacy guarantees.

The problem of privacy accounting has seen rapid progress during the work on this dissertation, which can be seen in the results: when working on Publication I, the classical Gaussian mechanism (see Section 2.3) was state of the art, and the question about tight privacy bounds was generally an important open question in any given setting. In contrast, in Publication IV we simply assume an oracle access to a black-box privacy accountant, and then instantiate the oracle in practice using a readily available numerical accountant implementation (Koskela et al., 2020b).

That said, there are still many open questions in privacy accounting. One prominent example is the shuffle model of DP (see Section 2.6): while we introduce numerical accounting for some common mechanisms in Publication III, there are still standard mechanisms, such as the Gaussian mechanism (see Section 2.3), for which the currently available privacy bounds cannot be computed tightly in many cases due to computational issues.

Chapter 5

Thesis contribution

5.1 Publication I

In Publication I, we propose an approach for DP Bayesian learning on distributed data. The method combines additive secret sharing with DP in order to achieve better noise scaling than is possible with LDP approaches. We introduce a secure aggregation protocol based on secret sharing for settings where there are several servers available. The threat model for the servers assumes that at least one of the servers does not collude with the others.

For privacy, we propose guaranteeing ADP in the conjugate-exponential family by perturbing the sufficient statistic, or more generally, by perturbing the parameter gradients with Gaussian noise. We show that under some assumptions, the sufficient statistic perturbation converges to the non-private estimator with optimal rate as the number of samples increases. We test our proposed method with Bayesian linear regression on several standard open data sets, as well as with genomic data for a cancer cell drug sensitivity prediction task.

5.2 Publication II

In Publication II, we introduce a general method for running MCMC under RDP guarantees. The method is based on utilising the inherent randomness present in the MCMC accept-reject decisions for the privacy analysis. The main idea needed for this analysis is to represent the standard Barker acceptance test as a standard logistic noise addition, and then break the logistic noise into two random variables, where one is Gaussian and provides RDP, and the second one represents a correction from the Gaussian to the logistic distribution.

Since the standard logistic noise requires a strict upper bound on the variance available for the privacy analysis, we enhance the base method by considering privacy amplification by data subsampling. Additionally, to improve the mixing properties of the chain, as well as to enable running the proposed method on larger data sets, we consider raising the likelihood temperature, and connect this temperature change to robust Bayesian methods.

5.3 Publication III

In Publication III, we construct dominating pairs of distributions for some common privacy mechanisms in the shuffle model of DP. By combining these dominating pairs with existing privacy accounting techniques, we achieve tighter privacy parameters than with the existing techniques, including for composing subsampled mechanisms. We also consider improving the runtime of the algorithms at the cost of slightly increasing the privacy parameter δ by applying Hoeffding's inequality in evaluating the required privacy loss distributions.

More specifically, we construct dominating pairs of distributions valid for any pure LDP mechanism, as well as for k-randomised response under two different adversaries. A stronger adversary knows whether the values communicated to the shuffler are randomised or not. We formulate and derive results also for a weaker adversary, who has the same extra knowledge except for a single party in the protocol, and show that this improves the resulting privacy bounds.

5.4 Publication IV

In Publication IV, we propose differentially private partitioned variational inference, a general framework for learning a DP variational approximation to an intractable Bayesian posterior distribution in the federated learning setting, while minimising the number of server-client communication rounds.

Within the general framework, we consider three specific implementations: one based on perturbing the local optimisation runs done by the individual parties, and two based on randomising the global model updates (one using a version of federated averaging, one adding virtual parties to the protocol). Given access to suitable secure primitives, such as secure aggregation or secure shuffling, the privacy guarantees for both of the approaches randomising the global model updates can be improved by all parties guaranteeing privacy jointly.

Finally, we compare the properties of the proposed implementations both theoretically, deriving the main properties especially for the proposed averaging approach, and empirically, using logistic regression and a fully connected Bayesian neural network for classification tasks.

References

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.

Agarwal, N., Kairouz, P., and Liu, Z. (2021). The Skellam mechanism for differentially private federated learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5052–5064. Curran Associates, Inc.

Agarwal, N., Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, B. (2018). cpSGD: Communication-efficient and differentially-private distributed SGD. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7575–7586.

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. (2021). Differentially private learning with adaptive clipping. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17455–17466.

Ashman, M., Bui, T. D., Nguyen, C. V., Markou, S., Weller, A., Swaroop, S., and Turner, R. E. (2022). Partitioned variational inference: A framework for probabilistic federated learning.

Asi, H., Duchi, J. C., and Javidbakht, O. (2019). Element level differential privacy: The right granularity of privacy. *CoRR*, abs/1912.04042.

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6280–6290.

Balle, B., Barthe, G., Gaboardi, M., Hsu, J., and Sato, T. (2020a). Hypothesis testing interpretations and Rényi differential privacy. In Chiappa, S. and Calandra, R., editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2496–2506. PMLR.

Balle, B., Bell, J., Gascón, A., and Nissim, K. (2019). The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, pages 638–667. Springer.

Balle, B., Cherubin, G., and Hayes, J. (2022). Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156.

Balle, B., Kairouz, P., McMahan, B., Thakkar, O. D., and Thakurta, A. (2020b). Privacy amplification via random check-ins. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Balle, B. and Wang, Y. (2018). Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 403–412. PMLR.

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a Proton-Electron plasma. *Aust. J. Phys.*, 18(2):119.

Barocas, S., Hardt, M., and Narayanan, A. (2019). *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org. http://www.fairmlbook.org.

Barthe, G. and Olmedo, F. (2013). Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In Fomin, F. V., Freivalds, R., Kwiatkowska, M., and Peleg, D., editors, *Automata, Languages, and Programming*, pages 49–60, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473.

Beimel, A., Brenner, H., Kasiviswanathan, S. P., and Nissim, K. (2014). Bounds on the sample complexity for private learning and private data release. *Mach. Learn.*, 94(3):401–437.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory.* John Wiley & Sons, Inc.

Bernstein, G. and Sheldon, D. (2019). Differentially private Bayesian linear regression. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 523–533.

Bernstein, G. and Sheldon, D. R. (2018). Differentially private Bayesian inference for exponential families. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2924–2934.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer New York.

Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., and Seefeld, B. (2017). Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. (2005). Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database* *systems*, PODS ’05, pages 128–138, New York, NY, USA. Association for Computing Machinery.

Boneh, D. and Shoup, V. (2023). *A Graduate Course in Applied Cryptography (version 0.6).*

Bun, M., Dwork, C., Rothblum, G. N., and Steinke, T. (2018). Composable and versatile privacy via truncated CDP. In Diakonikolas, I., Kempe, D., and Henzinger, M., editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, 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.

Canonne, C. L., Kamath, G., and Steinke, T. (2020). The discrete Gaussian for differential privacy. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramèr, F. (2022). Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. (2019). The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, Santa Clara, CA. USENIX Association.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. (2021). Extracting training data from large language models. In Bailey, M. and Greenstadt, R., editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.

Casacuberta, S., Shoemate, M., Vadhan, S., and Wagaman, C. (2022). Widespread underestimation of sensitivity in differentially private libraries and how to fix it. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’22, page 471–484, New York, NY, USA. Association for Computing Machinery.

Casella, G. and Berger, R. L. (2001). *Statistical Inference.* Duxbury.

Ceperley, D. M. and Dewing, M. (1999). The penalty method for random walks with uncertain energies. *J. Chem. Phys.*, 110(20):9812–9820.

Chaudhuri, K., Guo, C., and Rabbat, M. (2022). Privacy-aware compression for federated data analysis. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 296–306. PMLR.

Chaudhuri, K. and Mishra, N. (2006). When random sampling preserves privacy. In *Advances in Cryptology - CRYPTO 2006*, pages 198–213. Springer Berlin Heidelberg.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90.

Chen, W., Kairouz, P., and Özgür, A. (2023). Breaking the communication-privacy-accuracy trilemma. *IEEE Trans. Inf. Theory*, 69(2):1261–1281.

Chen, W.-N., Ozgur, A., and Kairouz, P. (2022). The Poisson binomial mechanism for unbiased federated learning with secure aggregation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3490–3506. PMLR.

Cheu, A. (2021). Differential privacy in the shuffle model: A survey of separations. *CoRR*, abs/2107.11839.

Cheu, A., Smith, A., Ullman, J., Zeber, D., and Zhilyaev, M. (2019). Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer.

Corrigan-Gibbs, H. and Boneh, D. (2017). Prio: Private, robust, and scalable computation of aggregate statistics. In Akella, A. and Howell, J., editors, *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*, pages 259–282. USENIX Association.

Dimitrakakis, C., Nelson, B., Mitrokotsa, A., and Rubinstein, B. I. P. (2014). Robust and private Bayesian inference. In Auer, P., Clark, A., Zeugmann, T., and Zilles, S., editors, *Algorithmic Learning Theory - 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, volume 8776 of *Lecture Notes in Computer Science*, pages 291–305. Springer.

Dimitrakakis, C., Nelson, B., Zhang, Z., Mitrokotsa, A., and Rubinstein, B. I. P. (2017). Differential privacy for Bayesian inference through posterior sampling. *Journal of Machine Learning Research*, 18(11):1–39.

Dinur, I. and Nissim, K. (2003). Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '03*, pages 202–210, New York, New York, USA. ACM Press.

Doroshenko, V., Ghazi, B., Kamath, P., Kumar, R., and Manurangsi, P. (2022). Connect the dots: Tighter discrete approximations of privacy loss distributions. *Proc. Priv. Enhancing Technol.*, 2022(4):552–570.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC 2009, pages 371–380, New York, NY, USA. Association for Computing Machinery.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. (2006b). Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T., editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer.

Dwork, C., Naor, M., Pitassi, T., Rothblum, G. N., and Yekhanin, S. (2010a). Pan-private streaming algorithms. In Yao, A. C., editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 66–80. Tsinghua University Press.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407.

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy. *CoRR*, abs/1603.01887.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010b). Boosting and differential privacy. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Washington, DC, USA. IEEE Computer Society.

Dwork, C. and Smith, A. (2010). Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2).

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. (2019a). Amplification by shuffling: From local to central differential privacy via anonymity. In Chan, T. M., editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2468–2479. SIAM.

Erlingsson, Ú., Mironov, I., Raghunathan, A., and Song, S. (2019b). That which we call private. *CoRR*, abs/1908.03566.

Feldman, V., McMillan, A., and Talwar, K. (2021). Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 954–964. IEEE.

Feldman, V., McMillan, A., and Talwar, K. (2022). Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. *CoRR*, abs/2208.04591.

Ferrantino, M. J. and Koten, E. E. (2019). The measurement and analysis of e-commerce : Frameworks for improving data availability. World Bank Group report. Available from http://documents.worldbank.org/curated/en/927771578286460819/The-Measurement-and-Analysis-of-E-Commerce-Frameworks-for-Improving-Data-Availability.

Foulds, J. R., Geumlek, J., Welling, M., and Chaudhuri, K. (2016). On the theory and practice of privacy-preserving Bayesian data analysis. In Ihler, A. T. and Janzing, D., editors, *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, 2016, New York City, NY, USA*. AUAI Press.

Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*.

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. *Proc USENIX Secur Symp*, 2014:17–32.

Geng, J., Mou, Y., Li, F., Li, Q., Beyan, O., Decker, S., and Rong, C. (2021). Towards general deep leakage in federated learning. *CoRR*, abs/2110.09074.

Geumlek, J., Song, S., and Chaudhuri, K. (2017). Rényi differential privacy mechanisms for posterior sampling. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5289–5298.

Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *CoRR*, abs/1712.07557.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. (2020). Pure differentially private summation from anonymous messages. In *1st Conference on Information-Theoretic Cryptography (ITC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. (2021a). On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In Canteaut, A. and Standaert, F., editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 463–488. Springer.

Ghazi, B., Kumar, R., Manurangsi, P., Pagh, R., and Sinha, A. (2021b). Differentially private aggregation in the shuffle model: Almost central accuracy in almost a single message. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3692–3701. PMLR.

Girgis, A. M., Data, D., Diggavi, S. N., Kairouz, P., and Suresh, A. T. (2021a). Shuffled model of differential privacy in federated learning. In Banerjee, A. and Fukumizu, K., editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2521–2529. PMLR.

Girgis, A. M., Data, D., Diggavi, S. N., Kairouz, P., and Suresh, A. T. (2021b). Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE J. Sel. Areas Inf. Theory*, 2(1):464–478.

Google (2023). The size and quality of a data set. Webpage, read on 2.2.2023. Available from https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality.

Gopi, S., Lee, Y. T., and Wutschitz, L. (2021). Numerical composition of differential privacy. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11631–11642.

Goryczka, S. and Xiong, L. (2017). A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Trans. Dependable Secure Comput.*, 14(5):463–477.

Haney, S., Desfontaines, D., Hartman, L., Shrestha, R., and Hay, M. (2022). Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers. *CoRR*, abs/2207.13793.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Heikkilä, M. A., Koskela, A., Shimizu, K., Kaski, S., and Honkela, A. (2020). Differentially private cross-silo federated learning. *CoRR*, abs/2007.05553.

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., and Craig, D. W. (2008). Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.*, 4(8):e1000167.

Honkela, A., Das, M., Nieminen, A., Dikmen, O., and Kaski, S. (2018). Efficient differentially private learning improves drug sensitivity prediction. *Biology Direct*, 13(1):1.

Huang, Z. and Kannan, S. (2012). The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 140–149. IEEE Computer Society.

Jälkö, J., Honkela, A., and Dikmen, O. (2017). Differentially private variational inference for non-conjugate models. In Elidan, G., Kersting, K., and Ihler, A. T., editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press.

Jayaraman, B., Wang, L., Evans, D., and Gu, Q. (2018). Distributed learning without distress: Privacy-preserving empirical risk minimization. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6346–6357.

Jordan, M. I. (2009). The exponential family: Basics. Unpublished lecture notes, available from https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter8.pdf.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233.

Kairouz, P., Liu, Z., and Steinke, T. (2021a). The distributed discrete Gaussian mechanism for federated learning with secure aggregation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5201–5212. PMLR.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021b). Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210.

Kairouz, P., Oh, S., and Viswanath, P. (2015). The composition theorem for differential privacy. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1376–1385. JMLR.org.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kearns, M. and Roth, A. (2019). *The Ethical Algorithm: The Science of Socially Aware Algorithm Design.* Oxford University Press.

Kim, M., Günlü, O., and Schaefer, R. F. (2021). Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2650–2654.

Koskela, A., Jälkö, J., and Honkela, A. (2020a). Computing tight differential privacy guarantees using FFT. In Chiappa, S. and Calandra, R., editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2560–2569. PMLR.

Koskela, A., Jälkö, J., Prediger, L., and Honkela, A. (2021). Tight differential privacy for discrete-valued mechanisms and for the subsampled Gaussian mechanism using FFT. In Banerjee, A. and Fukumizu, K., editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 3358–3366. PMLR.

Koskela, A., Prediger, L., Jälkö, J., and Honkela, A. (2020b). Fourier accountant. Python package, available from https://pypi.org/project/fourier-accountant/.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18:14:1–14:45.

Kulkarni, T., Jälkö, J., Koskela, A., Kaski, S., and Honkela, A. (2021). Differentially private Bayesian inference for generalized linear models. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5838–5849. PMLR.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE.

Li, B., Chen, C., Liu, H., and Carin, L. (2019). On connecting stochastic gradient MCMC and differential privacy. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 557–566. PMLR.

Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2015). Stochastic expectation propagation. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2323–2331.

Lin, J., Dang, L., Rahouti, M., and Xiong, K. (2021). ML attack models: Adversarial attacks and data poisoning attacks. *CoRR*, abs/2112.02797.

Lindell, Y. and Pinkas, B. (2009). Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1).

Liu, K. Z., Hu, S., Wu, Z. S., and Smith, V. (2022). On privacy and personalization in cross-silo federated learning. *CoRR*, abs/2206.07902.

Long, Y., Bindschaedler, V., Wang, L., Bu, D., Wang, X., Tang, H., Gunter, C. A., and Chen, K. (2018). Understanding membership inferences on well-generalized learning models. *CoRR*, abs/1802.04889.

Long, Y., Wang, L., Bu, D., Bindschaedler, V., Wang, X., Tang, H., Gunter, C. A., and Chen, K. (2020). A pragmatic approach to membership inferences on machine learning models. *Proceedings - 5th IEEE European Symposium on Security and Privacy, Euro S and P 2020*, pages 521–534.

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Singh, A. and Zhu, X. J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2018). Learning differentially private recurrent language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103.

Meiser, S. (2018). Approximate and probabilistic differential privacy definitions. Cryptology ePrint Archive, Paper 2018/277. https://eprint.iacr.org/2018/277.

Meiser, S. and Mohammadi, E. (2018). Tight on budget?: Tight bounds for r-fold approximate differential privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 247–264. ACM.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092.

Meyn, S. P. and Tweedie, R. L. (2005). *Markov Chains and Stochastic Stability*. Springer London.

Minami, K., Arai, H., Sato, I., and Nakagawa, H. (2016). Differential privacy without sensitivity. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 956–964.

Minka, T. (2005). Divergence measures and message passing. Technical Report MSR-TR-2005-173.

Mir, D. (2012). Differentially-private learning and information theory. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, pages 206–210, New York, NY, USA. Association for Computing Machinery.

Mironov, I. (2012). On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 650–661, New York, NY, USA. Association for Computing Machinery.

Mironov, I. (2017). Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275.

Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. (2009). Computational differential privacy. In Halevi, S., editor, *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 126–142. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mironov, I., Talwar, K., and Zhang, L. (2019). Rényi differential privacy of the sampled Gaussian mechanism. *CoRR*, abs/1908.10530.

Mohammed, N., Alhadidi, D., Fung, B. C. M., and Debbabi, M. (2014). Secure Two-Party differentially private data release for vertically partitioned data. *IEEE Trans. Dependable Secure Comput.*, 11(1):59–71.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective.* MIT Press.

Murtagh, J. and Vadhan, S. (2016). The complexity of computing the optimal composition of differential privacy. In *Theory of Cryptography Conference*, pages 157–175.

Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 739–753. IEEE.

Nasr, M., Song, S., Thakurta, A., Papernot, N., and Carlini, N. (2021). Adversary instantiation: Lower bounds for differentially private machine learning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 866–882. IEEE.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'99, pages 223–238, Berlin, Heidelberg. Springer-Verlag.

Park, M., Foulds, J. R., Chaudhuri, K., and Welling, M. (2020). Variational Bayes in private settings (VIPS). *J. Artif. Intell. Res.*, 68:109–157.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612.

Räisä, O., Koskela, A., and Honkela, A. (2021). Differentially private Hamiltonian Monte Carlo. *CoRR*, abs/2106.09376.

Rastogi, V. and Nath, S. (2010). Differentially private aggregation of distributed time-series with transformation and encryption. In Elmagarmid, A. K. and Agrawal, D., editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 735–746. ACM.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York, 2. edition.

Rogers, R. M., Vadhan, S. P., Roth, A., and Ullman, J. R. (2016). Privacy odometers and filters: Pay-as-you-go composition. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1921–1929.

Roser, M., Ritchie, H., and Ortiz-Ospina, E. (2015). Internet. *Our World in Data*. Webpage, read on 3.2.2023. Available from https://ourworldindata.org/internet.

Roussi, A. (2020). Resisting the rise of facial recognition. *Nature*, 587, 350-353. Available from https://doi.org/10.1038/d41586-020-03188-2.

Sabater, C., Bellet, A., and Ramon, J. (2022). An accurate, scalable and verifiable protocol for federated differentially private averaging. *Mach. Learn.*, 111(11):4249–4293.

Seita, D., Pan, X., Chen, H., and Canny, J. F. (2017). An efficient minibatch acceptance test for Metropolis-Hastings. In Elidan, G., Kersting, K., and Ihler, A. T., editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press.

Shamir, A. (1979). How to share a secret. *Commun. ACM*, 22(11):612–613.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society.

Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *Proc. GlobalSIP 2013*, pages 245–248.

Steinke, T. (2022). Composition of differential privacy & privacy amplification by subsampling. *CoRR*, abs/2210.00597.

Tajeddine, R., Jälkö, J., Kaski, S., and Honkela, A. (2020). Privacy-preserving data sharing on vertically partitioned data. *CoRR*, abs/2010.09293.

Titsias, M. K. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational Bayes for non-conjugate inference. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1971–1979. JMLR.org.

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*, AISec'19, pages 1–11, New York, NY, USA. Association for Computing Machinery.

Truex, S., Liu, L., Chow, K.-H., Gursoy, M. E., and Wei, W. (2020). LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, EdgeSys '20, pages 61–66, New York, NY, USA. Association for Computing Machinery.

Tschantz, M. C., Sen, S., and Datta, A. (2020). SoK: Differential privacy as a causal property. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE.

Vadhan, S. P. and Wang, T. (2021). Concurrent composition of differential privacy. In Nissim, K. and Waters, B., editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part II*, volume 13043 of *Lecture Notes in Computer Science*, pages 582–604. Springer.

Varshney, K. R. (2022). *Trustworthy Machine Learning.* Independently Published, Chappaqua, NY, USA.

Villalobos, P., Sevilla, J., Besiroglu, T., Heim, L., Ho, A., and Hobbhahn, M. (2022). Machine learning model sizes and the parameter gap. *CoRR*, abs/2207.02852.

Vinaroz, M. and Park, M. (2022). Differentially private stochastic expectation propagation. *Transactions on Machine Learning Research.*

Wang, Y., Balle, B., and Kasiviswanathan, S. P. (2019). Subsampled Rényi differential privacy and analytical moments accountant. In Chaudhuri, K. and Sugiyama, M., editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1226–1235. PMLR.

Wang, Y., Fienberg, S. E., and Smola, A. J. (2015). Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2493–2502. JMLR.org.

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69. PMID: 12261830.

Wasserman, L. (2004). *All of Statistics. A Concise Course in Statistical Inference.* Springer Texts in Statistics. Springer New York.

Watson, L., Guo, C., Cormode, G., and Sablayrolles, A. (2022). On the importance of difficulty calibration in membership inference attacks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., and Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. Omnipress.

Williams, O. and McSherry, F. (2010). Probabilistic inference and differential privacy. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2451–2459. Curran Associates, Inc.

Yao, A. C. (1982). Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164. IEEE Computer Society.

Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. (2022). Enhanced membership inference attacks against machine learning models. In Yin, H., Stavrou, A., Cremers, C., and Shi, E., editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 3093–3106. ACM.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 268–282. IEEE Computer Society.

Yıldırım, S. and Ermiş, B. (2019). Exact MCMC with differentially private moves. *Statistics and Computing*, 29(5):947–963.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2019). Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Zhang, Z., Rubinstein, B. I. P., and Dimitrakakis, C. (2016). On the differential privacy of Bayesian inference. In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2365–2371. AAAI Press.

Zhao, B., Mopuri, K. R., and Bilen, H. (2020). iDLG: Improved deep leakage from gradients. *CoRR*, abs/2001.02610.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14747–14756.

Zhu, Y., Dong, J., and Wang, Y.-X. (2022). Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*, pages 4782–4817. PMLR.

Zhu, Y. and Wang, Y.-X. (2019). Poisson subsampled Rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642.

Paper I

Mikko A. Heikkilä, Eemil Lagerspetz, Samuel Kaski, Kana Shimizu, Sasu Tarkoma and Antti Honkela

Differentially private Bayesian learning on distributed data

In *Annual Conference on Neural Information Processing Systems (NIPS) 2017, December 4-9, 2017, Long Beach, CA, USA*,
Advances in Neural Information Processing Systems 30, pages 3226–3235.

Errata for Publication I

In Figures 2 and 3 in Publication I, the results are plotted also for values $\varepsilon > 1$. This shows how the performance scales with increasing noise variance, but none of the algorithms are guaranteed to be (ε, δ)-DP with the stated privacy parameters when $\varepsilon > 1$ due to the properties of the classical Gaussian mechanism.

Differentially private Bayesian learning on distributed data

Mikko Heikkilä[1]
mikko.a.heikkila@helsinki.fi

Eemil Lagerspetz[2]
eemil.lagerspetz@helsinki.fi

Samuel Kaski[3]
samuel.kaski@aalto.fi

Kana Shimizu[4]
shimizu.kana.g@gmail.com

Sasu Tarkoma[2]
sasu.tarkoma@helsinki.fi

Antti Honkela[1,5]
antti.honkela@helsinki.fi

[1] Helsinki Institute for Information Technology HIIT,
Department of Mathematics and Statistics, University of Helsinki
[2] Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki
[3] Helsinki Institute for Information Technology HIIT,
Department of Computer Science, Aalto University
[4] Department of Computer Science and Engineering, Waseda University
[5] Department of Public Health, University of Helsinki

Abstract

Many applications of machine learning, for example in health care, would benefit from methods that can guarantee privacy of data subjects. Differential privacy (DP) has become established as a standard for protecting learning results. The standard DP algorithms require a single trusted party to have access to the entire data, which is a clear weakness, or add prohibitive amounts of noise. We consider DP Bayesian learning in a distributed setting, where each party only holds a single sample or a few samples of the data. We propose a learning strategy based on a secure multi-party sum function for aggregating summaries from data holders and the Gaussian mechanism for DP. Our method builds on an asymptotically optimal and practically efficient DP Bayesian inference with rapidly diminishing extra cost.

1 Introduction

Differential privacy (DP) [9, 11] has recently gained popularity as the theoretically best-founded means of protecting the privacy of data subjects in machine learning. It provides rigorous guarantees against breaches of individual privacy that are robust even against attackers with access to additional side information. DP learning methods have been proposed e.g. for maximum likelihood estimation [24], empirical risk minimisation [5] and Bayesian inference [e.g. 8, 13, 16, 17, 19, 25, 29]. There are DP versions of most popular machine learning methods, including linear regression [16, 28], logistic regression [4], support vector machines [5], and deep learning [1].

Almost all existing DP machine learning methods assume that some trusted party has unrestricted access to all the data in order to add the necessary amount of noise needed for the privacy guarantees.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

This is a highly restrictive assumption for many applications, e.g. for learning with data on mobile devices, and creates huge privacy risks through a potential single point of failure.

In this paper we introduce a general strategy for DP Bayesian learning in the distributed setting with minimal overhead. Our method builds on the asymptotically optimal sufficient statistic perturbation mechanism [13, 16] and shares its asymptotic optimality. The method is based on a DP secure multi-party communication (SMC) algorithm, called Distributed Compute algorithm (DCA), for achieving DP in the distributed setting. We demonstrate good performance of the method on DP Bayesian inference using linear regression as an example.

1.1 Our contribution

We propose a general approach for privacy-sensitive learning in the distributed setting. Our approach combines SMC with DP Bayesian learning methods, originally introduced for the non-distributed setting including a trusted party, to achieve DP Bayesian learning in the distributed setting.

To demonstrate our framework in practice, we combine the Gaussian mechanism for (ϵ, δ)-DP with efficient DP Bayesian inference using sufficient statistics perturbation (SSP) and an efficient SMC approach for secure distributed computation of the required sums of sufficient statistics. We prove that the Gaussian SSP is an efficient (ϵ, δ)-DP Bayesian inference method and that the distributed version approaches this quickly as the number of parties increases. We also address the subtle challenge of normalising the data privately in a distributed manner, required for the proof of DP in distributed DP learning.

2 Background

2.1 Differential privacy

Differential privacy (DP) [11] gives strict, mathematically rigorous guarantees against intrusions on individual privacy. A randomised algorithm is differentially private if its results on adjacent data sets are likely to be similar. Here adjacency means that the data sets differ by a single element, i.e., the two data sets have the same number of samples, but they differ on a single one. In this work we utilise a relaxed version of DP called (ϵ, δ)-DP [9, Definition 2.4].

Definition 2.1. A randomised algorithm $\mathcal{A}$ is (ϵ, δ)-DP, if for all $\mathcal{S} \subseteq \text{Range}\ (\mathcal{A})$ and all adjacent data sets D, D',

$$P(\mathcal{A}(D) \in \mathcal{S}) \leq \exp(\epsilon) P(\mathcal{A}(D') \in \mathcal{S}) + \delta.$$

The parameters ϵ and δ in Definition 2.1 control the privacy guarantee: ϵ tunes the amount of privacy (smaller ϵ means stricter privacy), while δ can be interpreted as the proportion of probability space where the privacy guarantee may break down.

There are several established mechanisms for ensuring DP. We use the Gaussian mechanism [9, Theorem 3.22]. The theorem says that given a numeric query f with ℓ_2-sensitivity $\Delta_2(f)$, adding noise distributed as $N(0, \sigma^2)$ to each output component guarantees (ϵ, δ)-DP, when

$$\sigma^2 > 2 \ln(1.25/\delta)(\Delta_2(f)/\epsilon)^2. \tag{1}$$

Here, the ℓ_2-sensitivity of a function f is defined as

$$\Delta_2(f) = \sup_{D \sim D'} \|f(D) - f(D')\|_2, \tag{2}$$

where the supremum is over all adjacent data sets D, D'.

2.2 Differentially private Bayesian learning

Bayesian learning provides a natural complement to DP because it inherently can handle uncertainty, including uncertainty introduced to ensure DP [26], and it provides a flexible framework for data modelling.

Three distinct types of mechanisms for DP Bayesian inference have been proposed:

1. Drawing a small number of samples from the posterior or an annealed posterior [7, 25];

2. Sufficient statistics perturbation (SSP) of an exponential family model [13, 16, 19]; and
3. Perturbing the gradients in gradient-based MCMC [25] or optimisation in variational inference [17].

For models where it applies, the SSP approach is asymptotically efficient [13, 16], unlike the posterior sampling mechanisms. The efficiency proof of [16] can be generalised to (ϵ, δ)-DP and Gaussian SSP as shown in the Supplementary Material.

The SSP (#2) and gradient perturbation (#3) mechanisms are of similar form in that the DP mechanism ultimately computes a perturbed sum

$$z = \sum_{i=1}^{N} z_i + \eta \tag{3}$$

over quantities z_i computed for different samples $i = 1, \dots, N$, where η denotes the noise injected to ensure the DP guarantee. For SSP [13, 16, 19], the z_i are the sufficient statistics of a particular sample, whereas for gradient perturbation [17, 25], the z_i are the clipped per-sample gradient contributions. When a single party holds the entire data set, the sum z in Eq. (3) can be computed easily, but the case of distributed data makes things more difficult.

3 Secure and private learning with distributed data

Let us assume there are N data holders (called clients in the following), who each hold a single data sample. We would like to use the aggregate data for learning, but the clients do not want to reveal their data as such to anybody else. The main problem with the distributed setting is that if each client uses a trusted aggregator (TA) DP technique separately, the noise η in Eq. (3) is added by each client, increasing the total noise variance by a factor of N compared to the non-distributed single TA setting, effectively reducing to naive input perturbation. To reduce the noise level without compromising on privacy, the individual data samples need to be combined without directly revealing them to anyone.

Our solution to this problem uses an SMC approach based on a form of secret sharing: each client sends their term of the sum, split to separate messages, to M servers such that together the messages sum up to the desired value, but individually they are just random noise. This can be implemented efficiently using a fixed-point representation of real numbers which allows exact cancelling of the noise in the addition. Like any secret sharing approach, this algorithm is secure as long as not all M servers collude. Cryptography is only required to secure the communication between the client and the server. Since this does not need to be homomorphic as in many other protocols, faster symmetric cryptography can be used for the bulk of the data. We call this the Distributed Compute Algorithm (DCA), which we introduce next in detail.

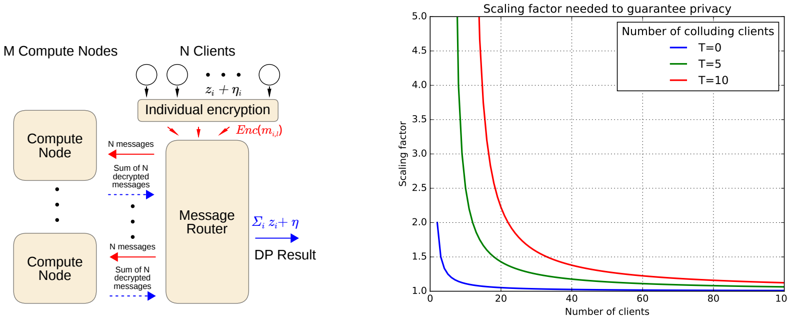
3.1 Distributed compute algorithm (DCA)

In order to add the correct amount of noise while avoiding revealing the unperturbed data to any single party, we combine an encryption scheme with the Gaussian mechanism for DP as illustrated in Fig. 1(a). Each individual client adds a small amount of Gaussian noise to his data, resulting in the aggregated noise to be another Gaussian with large enough variance. The details of the noise scaling are presented in the Section 3.1.2.

The scheme relies on several independent aggregators, called Compute nodes (Algorithm 1). At a general level, the clients divide their data and some blinding noise into shares that are each sent to one Compute. After receiving shares from all clients, each Compute decrypts the values, sums them and broadcasts the results. The final results can be obtained by summing up the values from all Computes, which cancels the blinding noise.

3.1.1 Threat model

We assume there are at most T clients who may collude to break the privacy, either by revealing the noise they add to their data samples or by abstaining from adding the noise in the first place. The rest are honest-but-curious (HbC), i.e., they will take a peek at other people's data if given the chance, but they will follow the protocol.



(a) DCA setting

(b) Extra scaling factor

Figure 1: 1(a): Schematic diagram of the Distributed Compute algorithm (DCA). Red refers to encrypted values, blue to unencrypted (but blinded or DP) values. 1(b) Extra scaling factor needed for the noise in the distributed setting with T colluding clients as compared to the trusted aggregator setting.

Algorithm 1 Distributed Compute Algorithm for distributed summation with independent Compute nodes

```
Input: $d$-dimensional vectors $\mathbf{z}_i$ held by clients $i \in \{1, \dots, N\}$;
    Distributed Gaussian mechanism noise variances $\sigma_j^2$, $j = 1, \dots, d$ (public);
    Number of parties $N$ (public);
    Number of Compute nodes $M$ (public);
Output: Differentially private sum $\sum_{i=1}^{N} (\mathbf{z}_i + \boldsymbol{\eta}_i)$, where $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \mathrm{diag}(\sigma_j^2))$
Each client $i$ simulates $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \mathrm{diag}(\sigma_j^2))$ and $M-1$ vectors $\mathbf{r}_{i,k}$ of uniformly random fixed-point data with $\mathbf{r}_{i,M} = -\sum_{k=1}^{M-1} \mathbf{r}_{i,k}$ to ensure that $\sum_{k=1}^{M} \mathbf{r}_{i,k} = \mathbf{0}_d$ (a vector of zeros).
Each client $i$ computes the messages $\mathbf{m}_{i,1} = \mathbf{z}_i + \boldsymbol{\eta}_i + \mathbf{r}_{i,1}$, $\mathbf{m}_{i,k} = \mathbf{r}_{i,k}, k = 2, \dots M$, and sends them securely to the corresponding Compute $k$.
After receiving messages from all of the clients, Compute $k$ decrypts the values and broadcasts the noisy aggregate sums $\mathbf{q}_k = \sum_{i=1}^{N} \mathbf{m}_{i,k}$. A final aggregator will then add these to obtain $\sum_{k=1}^{M} \mathbf{q}_k = \sum_{i=1}^{N} (\mathbf{z}_i + \boldsymbol{\eta}_i)$.
```

To break the privacy of individual clients, all Compute nodes need to collude. We therefore assume that at least one Compute node follows the protocol. We further assume that all parties have an interest in the results and hence will not attempt to pollute the results with invalid values.

3.1.2 Privacy of the mechanism

In order to guarantee that the sum-query results returned by Algorithm 1 are DP, we need to show that the variance of the aggregated Gaussian noise is large enough.

Theorem 1 (Distributed Gaussian mechanism)**.** *If at most T clients collude or drop out of the protocol, the sum-query result returned by Algorithm 1 is (ϵ, δ)-DP, when the variance of the added noise σ_j^2 fulfils*

$$\sigma_j^2 \geq \frac{1}{N - T - 1} \sigma_{j,std}^2,$$

where N is the number of clients and $\sigma_{j,std}^2$ is the variance of the noise in the standard (ϵ, δ)-DP Gaussian mechanism given in Eq. (1).

Proof. See Supplement. □

In the case of all HbC clients, $T = 0$. The extra scaling factor increases the variance of the DP, but this factor quickly approaches 1 as the number of clients increases, as can be seen from Figure 1(b).

3.1.3 Fault tolerance

The Compute nodes need to know which clients' contributions they can safely aggregate. This feature is simple to implement e.g. with pairwise-communications between all Compute nodes. In order to avoid having to start from scratch due to insufficient noise for DP, the same strategy used to protect against colluding clients can be utilized: when $T > 0$, at most T clients in total can drop or collude and the scheme will still remain private.

3.1.4 Computational scalability

Most of the operations needed in Algorithm 1 are extremely fast: encryption and decryption can use fast symmetric algorithms such as AES (using slower public key cryptography just for the key of the symmetric system) and the rest is just integer additions for the fixed point arithmetic. The likely first bottlenecks in the implementation would be caused by synchronisation when gathering the messages as well as the generation of cryptographically secure random vectors $\mathbf{r}_{i,k}$.

3.2 Differentially private Bayesian learning on distributed data

In order to perform DP Bayesian learning securely in the distributed setting, we use DCA (Algorithm 1) to compute the required data summaries that correspond to Eq. (3). In this Section we consider how to combine this scheme with concrete DP learning methods introduced for the trusted aggregator setting, so as to provide a wide range of possibilities for performing DP Bayesian learning securely with distributed data.

The aggregation algorithm is most straightforward to apply to the SSP method [13, 16] for exact and approximate posterior inference on exponential family models. [13] and [16] use Laplacian noise to guarantee ϵ-DP, which is a stricter form of privacy than the (ϵ, δ)-DP used in DCA [9]. We consider here only (ϵ, δ)-DP version of the methods, and discuss the possible Laplace noise mechanism further in Section 7. The model training in this case is done in a single iteration, so a single application of Algorithm 1 is enough for learning. We consider a more detailed example in Section 3.2.1.

We can also apply DCA to DP variational inference [17, 19]. These methods rely on possibly clipped gradients or expected sufficient statistics calculated from the data. Typically, each training iteration would use only a mini-batch instead of the full data. To use variational inference in the distributed setting, an arbitrary party keeps track of the current (public) model parameters and the privacy budget, and asks for updates from the clients.

At each iteration, the model trainer selects a random mini-batch of fixed public size from the available clients and sends them the current model parameters. The selected clients then calculate the clipped gradients or expected sufficient statistics using their data, add noise to the values scaled reflecting the batch size, and pass them on using DCA. The model trainer receives the decrypted DP sums from the output and updates the model parameters.

3.2.1 Distributed Bayesian linear regression with data projection

As an empirical example, we consider Bayesian linear regression (BLR) with data projection in the distributed setting. The standard BLR model depends on the data only through sufficient statistics and the approach discussed in Section 3.2 can be used in a straightforward manner to fit the model by running a single round of DCA.

The more efficient BLR with projection (Algorithm 2) [16] reduces the data range, and hence sensitivity, by non-linearly projecting all data points inside stricter bounds, which translates into less added noise. We can select the bounds to optimize bias vs. DP noise variance. In the distributed setting, we need to run an additional round of DCA and use some privacy budget to estimate data standard deviations (stds). However, as shown by the test results (Figures 2 and 3), this can still achieve significantly better utility with a given privacy level.

The assumed bounds in Step 1 of Algorithm 2 would typically be available from general knowledge of the data. The initial projection in Step 1 ensures the privacy of the scheme even if the bounds are invalid for some samples. We determine the optimal final projection thresholds p_j in Step 3 using the same general approach as [16]: we create an auxiliary data set of equal size as the original with data

Algorithm 2 Distributed linear regression with projection

```
Input: Data and target values $(x_{ij}, y_i), j = 1, \dots, d$ held by clients $i \in \{1, \dots, N\}$;
    Number of clients $N$ (public);
    Assumed data and target bounds $(-c_j, c_j), j = 1, \dots, d+1$ (public);
    Privacy budget $(\epsilon, \delta)$ (public);
Output: DP BLR model sufficient statistics of projected data $\sum_{i=1}^{N} \hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^T + \boldsymbol{\eta}^{(1)}$, $\sum_{i=1}^{N} \hat{\mathbf{x}}_i^T \hat{y}_i + \boldsymbol{\eta}^{(2)}$,
    calculated using projection to estimated optimal bounds
  Each client projects his data to the assumed bounds $(-c_j, c_j)\ \forall j$.
  Calculate marginal std estimates $\sigma^{(1)}, \dots, \sigma^{(d+1)}$ by running Algorithm 1 using the assumed
    bounds for sensitivity and a chosen share of the privacy budget.
  Estimate optimal projection thresholds $p_j, j = 1, \dots, d+1$ as fractions of std on auxiliary
    data. Each client then projects his data to the estimated optimal bounds $(-p_j\sigma^{(j)}, p_j\sigma^{(j)}), j =$
    $1, \dots, d+1$.
  Aggregate the unique terms in the DP sufficient statistics by running Algorithm 1 using the
    estimated optimal bounds for sensitivity and the remaining privacy budget, and combine the
    DP result vectors into the symmetric $d \times d$ matrix and $d$-dimensional vector of DP sufficient
    statistics.
```

generated as

$$x_i \sim N(0, I_d) \tag{4}$$

$$\beta \sim N(0, \lambda_0 I) \tag{5}$$

$$y_i | x_i \sim N(x_i^T \beta, \lambda). \tag{6}$$

We then perform grid search on the auxiliary data with varying thresholds to find the one providing optimal prediction performance. The source code for our implementation is available through GitHub[1] and a more detailed description can be found in the Supplement.

4 Experimental setup

We demonstrate the secure DP Bayesian learning scheme in practice by testing the performance of the BLR with data projection, the implementation of which was discussed in Section 3.2.1, along with the DCA (Algorithm 1) in the all HbC clients distributed setting ($T = 0$).

With the DCA our primary interest is scalability. In the case of BLR implementation, we are mostly interested in comparing the distributed algorithm to the trusted aggregator version as well as comparing the performance of the straightforward BLR to the variant using data projection, since it is not clear a priori if the extra cost in privacy necessitated by the projection in the distributed setting is offset by the reduced noise level.

We use simulated data for the DCA scalability testing, and real data for the BLR tests. As real data, we use the Wine Quality [6] (split into white and red wines) and Abalone data sets from the UCI repository[18], as well as the Genomics of Drug Sensitivity in Cancer (GDSC) project data [2]. The measured task in the GDSC data is to predict drug sensitivity of cancer cell lines from gene expression data (see Supplement for a more detailed description). The datasets are assumed to be zero-centred. This assumption is not crucial but is done here for simplicity; non-zero data means can be estimated like the marginal stds at the cost of some added noise (see Section 3.2.1).

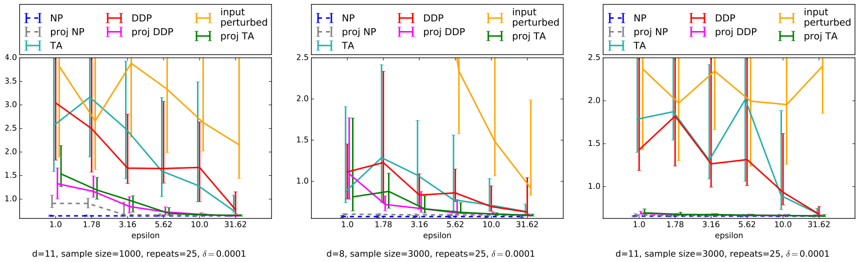
For estimating the marginal std, we also need to assume bounds for the data. For unbounded data, we can enforce arbitrary bounds simply by projecting all data inside the chosen bounds, although very poor choice of bounds will lead to poor performance. With real distributed data, the assumed bounds could differ from the actual data range. In the UCI tests we simulate this effect by scaling each data dimension to have a range of length 10, and then assuming bounds of $[-7.5, 7.5]$, i.e., the assumed bounds clearly overestimate the length of the true range, thus adding more noise to the results. The actual scaling chosen here is arbitrary. With the GDSC data, the true ranges are mostly known due to the nature of the data (see Supplement).

[1] `https://github.com/DPBayes/dca-nips2017`

[2] `http://www.cancerrxgene.org/`, release 6.1, March 2017

	N=10^2	**N**=10^3	**N**=10^4	**N**=10^5
d=10	1.72	1.89	2.99	8.58
d=10^2	2.03	2.86	12.36	65.64
d=10^3	3.43	10.56	101.2	610.55
d=10^4	15.30	84.95	994.96	1592.29

Table 1: DCA experiment average runtimes in seconds with 5 repeats, using M=10 Compute nodes, N clients and vector length d.



(a) Red wine data set (b) Abalone data set (c) White wine data set

Figure 2: Median of the predictive accuracy measured on mean absolute error (MAE) on several UCI data sets with error bars denoting the interquartile range (lower is better). The performance of the distributed methods (DDP, DDP proj) is indistinguishable from the corresponding undistributed algorithms (TA, TA proj) and the projection (proj TA, proj DDP) can clearly be beneficial for prediction performance. NP refers to non-private version, TA to the trusted aggregator setting, DDP to the distributed scheme.

The optimal projection thresholds are searched for using 10 (GDSC) or 20 (UCI) repeats on a grid with 20 points between 0.1 and 2.1 times the std of the auxiliary data set. In the search we use one common threshold for all data dimensions and a separate one for the target.

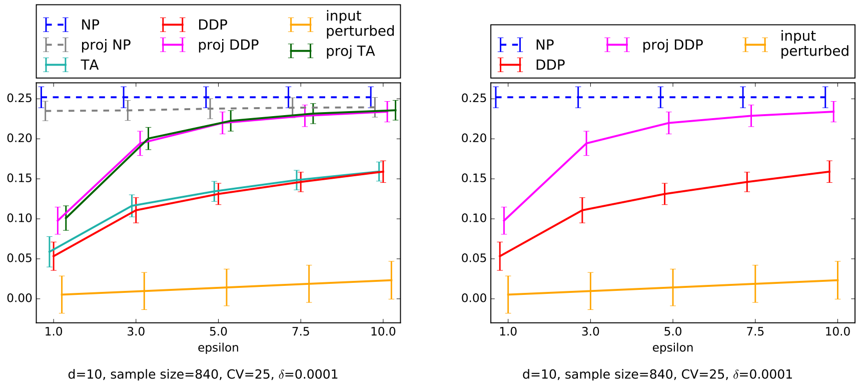
For accuracy measure, we use prediction accuracy on a separate test data set. The size of the test set for UCI in Figure 2 is 500 for red wine, 1000 for white wine, and 1000 for abalone data. The test set size for GDSC in Figure 3 is 100. For UCI, we compare the median performance measured on mean absolute error over 25 cross-validation (CV) runs, while for GDSC we measure mean prediction accuracy to sensitive vs insensitive with Spearman's rank correlation on 25 CV runs. In both cases, we use input perturbation [11] and the trusted aggregator setting as baselines.

5 Results

Table 1 shows runtimes of a distributed Spark implementation of the DCA algorithm. The timing excludes encryption, but running AES for the data of the largest example would take less than 20 s on a single thread on a modern CPU. The runtime modestly increases as N or d is increased. This suggests that the prototype is reasonably scalable. Spark overhead sets a lower bound runtime of approximately 1 s for small problems. For large N and d, sequential communication at the 10 Compute threads is the main bottleneck. Larger N could be handled by introducing more Compute nodes and clients only communicating with a subset of them.

Comparing the results on predictive error with and without projection (Fig. 2 and Fig. 3), it is clear that despite incurring extra privacy cost for having to estimate the marginal standard deviations, using the projection can improve the results markedly with a given privacy budget.

The results also demonstrate that compared to the trusted aggregator setting, the extra noise added due to the distributed setting with HbC clients is insignificant in practice as the results of the distributed and trusted aggregator algorithms are effectively indistinguishable.



(a) Drug sensitivity prediction

(b) Drug sensitivity prediction, selected methods

Figure 3: Mean drug sensitivity prediction accuracy on GDSC dataset with error bars denoting standard deviation over CV runs (higher is better). Distributed results (DDP, proj DDP) do not differ markedly from the corresponding trusted aggregator (TA, proj TA) results. The projection (proj TA, proj DDP) is clearly beneficial for performance. The actual sample size varies between drugs. NP refers to non-private version, TA to the trusted aggregator setting, DDP to the distributed scheme.

6 Related work

The idea of distributed private computation through addition of noise generated in a distributed manner was first proposed by Dwork et al. [10]. However, to the best of our knowledge, there is no prior work on secure DP Bayesian statistical inference in the distributed setting.

In machine learning, [20] presented the first method for aggregating classifiers in a DP manner, but their approach is sensitive to the number of parties and sizes of the data sets held by each party and cannot be applied in a completely distributed setting. [21] improved upon this by an algorithm for distributed DP stochastic gradient descent that works for any number of parties. The privacy of the algorithm is based on perturbation of gradients which cannot be directly applied to the efficient SSP mechanism. The idea of aggregating classifiers was further refined in [15] through a method that uses an auxiliary public data set to improve the performance.

The first practical method for implementing DP queries in a distributed manner was the distributed Laplace mechanism presented in [22]. The distributed Laplace mechanism could be used instead of the Gaussian mechanism if pure ϵ-DP is required, but the method, like those in [20, 21], needs homomorphic encryption which is computationally more demanding, especially for high-dimensional data.

There is a wealth of literature on secure distributed computation of DP sum queries as reviewed in [14]. The methods of [23, 2, 3, 14] also include different forms of noise scaling to provide collusion resistance and/or fault tolerance, where the latter requires a separate recovery round after data holder failures which is not needed by DCA. [12] discusses low level details of an efficient implementation of the distributed Laplace mechanism.

Finally, [27] presents several proofs related to the SMC setting and introduce a protocol for generating approximately Gaussian noise in a distributed manner. Compared to their protocol, our method of noise addition is considerably simpler and faster, and produces exactly instead of approximately Gaussian noise with negligible increase in noise level.

7 Discussion

We have presented a general framework for performing DP Bayesian learning securely in a distributed setting. Our method combines a practical SMC method for calculating secure sum queries with efficient Bayesian DP learning techniques adapted to the distributed setting.

DP methods are based on adding sufficient noise to effectively mask the contribution of any single sample. The extra loss in accuracy due to DP tends to diminish as the number of samples increases and efficient DP estimation methods converge to their non-private counterparts as the number of samples increases [13, 16]. A distributed DP learning method can significantly help in increasing the number of samples because data held by several parties can be combined thus helping make DP learning significantly more effective.

Considering the DP and the SMC components separately, although both are necessary for efficient privacy-aware learning, it is clear that the choice of method to use for each sub-problem can be made largely independently. Assessing these separately, we can therefore easily change the privacy mechanism from the Gaussian used in Algorithm 1 to the Laplace mechanism, e.g. by utilising one of the distributed Laplace noise addition methods presented in [14] to obtain a pure ϵ-DP method. If need be, the secure sum algorithm in our method can also be easily replaced with one that better suits the security requirements at hand.

While the noise introduced for DP will not improve the performance of an otherwise good learning algorithm, a DP solution to a learning problem can yield better results if the DP guarantees allow access to more data than is available without privacy. Our distributed method can further help make this more efficient by securely and privately combining data from multiple parties.

Acknowledgements

This work was funded by the Academy of Finland [Centre of Excellence COIN and projects 259440, 278300, 292334, 294238, 297741, 303815, 303816], the Japan Agency for Medical Research and Development (AMED), and JST CREST [JPMJCR1688].

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proc. CCS 2016*, 2016.

[2] G. Ács and C. Castelluccia. I have a DREAM! (DiffeRentially privatE smArt Metering). In *Proc. 13th International Conference in Information Hiding (IH 2011)*, pages 118–132, 2011.

[3] T. H. H. Chan, E. Shi, and D. Song. Privacy-preserving stream aggregation with fault tolerance. In *Proc. 16th Int. Conf. on Financial Cryptography and Data Security (FC 2012)*, pages 200–214, 2012.

[4] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems 21*, pages 289–296. 2009.

[5] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, 2011.

[6] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[7] C. Dimitrakakis, B. Nelson, A. Mitrokotsa, and B. I. P. Rubinstein. Robust and private Bayesian inference. In *Proc. ALT 2014*, pages 291–305, 2014.

[8] C. Dimitrakakis, B. Nelson, Z. Zhang, A. Mitrokotsa, and B. I. P. Rubinstein. Differential privacy for Bayesian inference through posterior sampling. *Journal of Machine Learning Research*, 18(11):1–39, 2017.

[9] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology (EUROCRYPT 2006)*, page 486–503, 2006.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. 3rd Theory of Cryptography Conference (TCC 2006)*, pages 265–284. 2006.

[12] F. Eigner, A. Kate, M. Maffei, F. Pampaloni, and I. Pryvalov. Differentially private data aggregation with optimal utility. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 316–325. ACM, 2014.

[13] J. Foulds, J. Geumlek, M. Welling, and K. Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pages 192–201, 2016.

[14] S. Goryczka and L. Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2015.

[15] J. Hamm, P. Cao, and M. Belkin. Learning privately from multiparty data. In *ICML*, 2016.

[16] A. Honkela, M. Das, A. Nieminen, O. Dikmen, and S. Kaski. Efficient differentially private learning improves drug sensitivity prediction. 2016. arXiv:1606.02109 [stat.ML].

[17] J. Jälkö, O. Dikmen, and A. Honkela. Differentially private variational inference for non-conjugate models. In *Proc. 33rd Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, 2017.

[18] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[19] M. Park, J. Foulds, K. Chaudhuri, and M. Welling. Variational Bayes in private settings (VIPS). 2016. arXiv:1611.00340.

[20] M. Pathak, S. Rane, and B. Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems 23*, pages 1876–1884, 2010.

[21] A. Rajkumar and S. Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Proc. AISTATS 2012*, pages 933–941, 2012.

[22] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proc. 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD 2010)*, pages 735–746. ACM, 2010.

[23] E. Shi, T. Chan, E. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Proc. NDSS*, 2011.

[24] A. Smith. Efficient, differentially private point estimators. 2008. arXiv:0809.4794 [cs.CR].

[25] Y. Wang, S. E. Fienberg, and A. J. Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proc. ICML 2015*, pages 2493–2502, 2015.

[26] O. Williams and F. McSherry. Probabilistic inference and differential privacy. In *Adv. Neural Inf. Process. Syst. 23*, 2010.

[27] G. Wu, Y. He, J. Wu, and X. Xia. Inherit differential privacy in distributed setting: Multiparty randomized function computation. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 921–928, 2016.

[28] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: Regression analysis under differential privacy. *PVLDB*, 5(11):1364–1375, 2012.

[29] Z. Zhang, B. Rubinstein, and C. Dimitrakakis. On the differential privacy of Bayesian inference. In *Proc. AAAI 2016*, 2016.

Supplement to "Differentially private Bayesian learning on distributed data"

Mikko Heikkilä, Eemil Lagerspetz, Samuel Kaski,
Kana Shimizu, Sasu Tarkoma, and Antti Honkela

This supplement contains proofs and extra discussion omitted from the main text.

1 Privacy and fault tolerance

Theorem 1 (Distributed Gaussian mechanism)**.** *If at most T clients collude or drop out of the protocol, the sum-query result returned by Algorithm 1 is differentially private, when the variance of the added noise σ_j^2 fulfils*

$$\sigma_j^2 \geq \frac{1}{N-T-1}\sigma_{j,std}^2,$$

where N is the number of clients and $\sigma_{j,std}^2$ is the variance of the noise in the standard Gaussian mechanism given in Eq. (1).

Proof. Using the property that a sum of independent Gaussian variables is another Gaussian with variance equal to the sum of the component variances, we can divide the total noise equally among the N clients.

However, in the distributed setting even with all honest-but-curious clients, there is an extra scaling factor needed compared to the standard DP. Since each client knows the noise values she adds to the data, she can also remove them from the aggregate values. In other words, privacy then has to be guaranteed by the noise the remaining $N-1$ clients add to the data. If we further assume the possibility of T colluding clients, then the noise from $N-T-1$ clients must be sufficient to guarantee the privacy.

The added noise can therefore be calculated from the inequality

$$\sum_{i=1}^{N-T-1} \sigma_j^2 \geq \sigma_{j,std}^2 \tag{1}$$

$$\Leftrightarrow \; \sigma_j^2 \geq \frac{1}{N-T-1}\sigma_{j,std}^2. \tag{2}$$

□

2 Bayesian linear regression

In the following, we denote the d-dimensional input data for the ith observation by $\mathbf{x}_i$, the scalar target values by y_i, and the whole $d+1-$dimensional dataset by $D_i = (\mathbf{x}_i, y_i)$. We assume all variable-wise expectations to be zeroes for simplicity. For n observations, we denote the sufficient statistics by $n\overline{xx} = \sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T$ and $n\overline{xy} = \sum_{i=1}^n \mathbf{x}_i y_i$.

For the regression, we assume that

$$y_i|\mathbf{x}_i \sim N(\mathbf{x}_i^T\beta, \lambda I), i = 1, \ldots, n \tag{3}$$

$$\beta \sim N(0, \lambda_0 I), \tag{4}$$

where we want to learn the posterior over β, and λ, λ_0 are hyperparameters (set to 1 in the tests). The posterior can be solved analytically to give

$$\beta|\mathbf{y}, \mathbf{x} \sim N(\hat{\mu}, \hat{\Lambda}), \tag{5}$$

$$\hat{\Lambda} = \lambda_0 I + \lambda n \overline{xx}, \tag{6}$$

$$\hat{\mu} = \hat{\Lambda}^{-1}(\lambda n \overline{xy}). \tag{7}$$

The predicted mean values from the model are $\hat{y} = \mathbf{x}^T \hat{\mu}$.

The DP sufficient statistics are given by $n\widehat{xx} = n\overline{xx} + \eta_{xx}, n\widehat{xy} = n\overline{xy} + \eta_{xy}$, where η_{xx}, η_{xy} consist of suitably scaled Gaussian noise added independently to each dimension. In total, there are $d(d+1)/2 + d$ parameters in the combined sufficient statistic, since $n\overline{xx}$ is a symmetric matrix.

The main idea in the data projection is simply to project the data into some reduced range. Since the noise level is determined by the sensitivity of the data, reducing the sensitivity by limiting the data range translates into less added noise.

With projection threshold c, the projection of data x_i is given by

$$\breve{x}_i = \max(-c, \min(x_i, c)). \tag{8}$$

This data projection obviously discards information, but in various problems it can be beneficial to disregard some information in the data in order to achieve less noisy estimates of the model parameters. From the bias-variance trade-off point of view, this can be seen as increasing the bias while reducing the variance. The optimal trade-off then depends on the actual problem.

To run Algorithm 2 (in the main text), we need to assume initial projection bounds (c_j, d_j) for each dimension $j \in \{1, \dots, d+1\}$ for the data $(\mathbf{x}_i, y_i)_{i=1}^n$. In the paper we assume bounds of the form $(-c_j, c_j)$. To find good final projection bounds, we first find an optimal projection threshold by a grid search on an auxiliary dataset, that is generated from a BLR model similar to the regression model defined above.

This gives us the projection thresholds in terms of std for each dimension. We then estimate the marginal std for each dimension by using Algorithm 1 (in the main text), to fix the actual projection thresholds. For this the data are assumed to lie on some known bounded interval. In practice, the assumed bounds need to be based on prior information. In case the estimates are negative due to noise, they are set to small positive constants (0.5 in all the tests).

The amount of noise each client needs to add to the output depends partly on the sensitivity of the function in question. The query function we are interested in returns a vector of length $d(d+1)/2+d$ that contains all the unique terms in the sufficient statistics needed for linear regression.

Let $\mathbf{x}, \mathbf{x}'$ be the mismatching, maximally different elements over adjacent datasets s.t. dimensions $1, \dots, d$ are the independent variables, and $d+1$ is the target. Assume further that each dimension $j = 1, \dots, d+1$ is bounded by $(-c_j, c_j)$. The squared sensitivity of the query f is then

$$\begin{aligned}
\Delta_2(f)^2 &= ||f(\mathbf{x}) - f(\mathbf{x}')||_2^2 && (9)\\
&= ||(x_j x_k - x'_j x'_k, x_j x_{d+1} - x'_j x'_{d+1})_{j=1,k=j}^d||_2^2 && (10)\\
&= \sum_{j=1}^d \sum_{k=j}^d (x_j x_k - x'_j x'_k)^2 + \sum_{j=1}^d (x_j x_{d+1} - x'_j x'_{d+1})^2 && (11)\\
&\leq \sum_{j=1}^d (c_j^2)^2 + \sum_{j=1}^d \sum_{k>j}^d (2 c_j c_k)^2 + \sum_{j=1}^d (2 c_j c_{d+1})^2. && (12)
\end{aligned}$$

We assume $c_j = c_x \forall j = 1, \dots, d$, so (12) can be further simplified to $d(2d-1)c_x^4 + 4d(c_x c_{d+1})^2$.

3 Asymptotic efficiency of the Gaussian mechanism

The asymptotic efficiency of the sufficient statistics perturbation using Laplace mechanism has been proven before [2, 3]. We show corresponding results for the Gaussian mechanism. The proofs generally follow closely those given in [3]. For convenience, we state the relevant definitions, but mostly focus on those proofs that differ in a non-trivial way from the existing ones for the Laplace mechanism. For the full proofs and related discussion, see [3].

3.1 Definition of asymptotic efficiency

Definition 3.1. A differentially private mechanism $\mathcal{M}$ is *asymptotically consistent with respect to an estimated parameter* θ if the private estimates $\hat{\theta}_{\mathcal{M}}$ given a data set $\mathcal{D}$ converge in probability to the corresponding non-private estimates $\hat{\theta}_{NP}$ as the number of samples, $n = |\mathcal{D}|$, grows without bound, i.e., if for any[1] $\alpha > 0$,

$$\lim_{n\to\infty} \Pr\{\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > \alpha\} = 0.$$

Definition 3.2. A differentially private mechanism $\mathcal{M}$ is *asymptotically efficiently private with respect to an estimated parameter* θ, if the mechanism is asymptotically consistent and the private estimates $\hat{\theta}_{\mathcal{M}}$ converge to the corresponding non-private estimates $\hat{\theta}_{NP}$ at the rate $\mathcal{O}(1/n)$, i.e., if for any $\alpha > 0$ there exist constants C, N such that

$$\Pr\{\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > C/n\} < \alpha$$

for all $n \geq N$.

The first part of Theorem 2 follows closely the corresponding result for the Laplace mechanism [3, Theorem 1]. The theorem shows that the optimal rate for estimating the expectation of exponential family distributions is $\mathcal{O}(1/n)$. This justifies the term asymptotically efficiently private introduced by [3], when we show that sufficient statistics perturbation by the Gaussian mechanism achieves this rate.

Theorem 2. *The private estimates $\hat{\theta}_{\mathcal{M}}$ of an exponential family posterior expectation parameter θ, generated by a differentially private mechanism $\mathcal{M}$ that achieves (ϵ, δ)-differential privacy for any $\epsilon > 0, \delta \in (0, 1)$, cannot converge to the corresponding non-private estimates $\hat{\theta}_{NP}$ at a rate faster than $1/n$. That is, assuming $\mathcal{M}$ is (ϵ, δ)-differentially private, there exists no function $f(n)$ such that $\limsup nf(n) = 0$ and for all $\alpha > 0$, there exists a constant N such that*

$$\Pr\{\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > f(n)\} < \alpha$$

for all $n \geq N$.

Proof. The non-private estimate of an expectation parameter of an exponential family is [1]

$$\hat{\theta}_{NP}|x_1, \ldots, x_n = \frac{n_0 x_0 + \sum_{i=1}^{n} x_i}{n_0 + n}. \tag{13}$$

The difference of the estimates from two neighbouring data sets differing by one element is

$$(\hat{\theta}_{NP}|\mathcal{D}) - (\hat{\theta}_{NP}|\mathcal{D}') = \frac{x - y}{n_0 + n}, \tag{14}$$

where x and y are the corresponding mismatched elements. Let $\Delta = \max(\|x - y\|)$, and let $\mathcal{D}$ and $\mathcal{D}'$ be neighbouring data sets including these maximally different elements.

Let us assume that there exists a function $f(n)$ such that $\limsup nf(n) = 0$ and for all $\alpha > 0$ there exists a constant N such that

$$\Pr\{\|\hat{\theta}_{\mathcal{M}} - \hat{\theta}_{NP}\| > f(n)\} < \alpha \tag{15}$$

for all $n \geq N$.

Fix $\alpha > 0$ and choose $M \geq \max(N, n_0)$ such that $f(n) \leq \Delta/4n$ for all $n \geq M$. This implies that

$$\|(\hat{\theta}_{NP}|\mathcal{D}) - (\hat{\theta}_{NP}|\mathcal{D}')\| = \frac{\Delta}{n_0 + n} \geq \frac{\Delta}{2n} \geq 2f(n). \tag{16}$$

Let us define the region $C_{\mathcal{D}} = \{t \mid \|(\hat{\theta}_{NP}|\mathcal{D}) - t\| < f(n)\}$.

[1] We use α in limit expressions instead of usual ϵ to avoid confusion with ϵ-differential privacy.

Based on our assumptions we have

$$\Pr\left((\hat{\theta}_{\mathcal{M}}|\mathcal{D}) \in C_{\mathcal{D}}\right) > 1 - \alpha. \tag{17}$$

Combining (16) and (15) we have

$$\Pr\left((\hat{\theta}_{\mathcal{M}}|\mathcal{D}') \in C_{\mathcal{D}}\right) < \alpha \tag{18}$$

which implies

$$\Pr\left((\hat{\theta}_{\mathcal{M}}|\mathcal{D}) \in C_{\mathcal{D}}\right) \le \exp(\epsilon)\Pr\left((\hat{\theta}_{\mathcal{M}}|\mathcal{D}') \in C_{\mathcal{D}}\right) + \delta. \tag{19}$$

Together these imply that

$$\begin{align} 1 - \alpha < \exp(\epsilon)\alpha + \delta \tag{20} \\ \Leftrightarrow \ \delta > 1 - (1 + \exp(\epsilon))\alpha. \tag{21} \end{align}$$

Since for fixed ϵ, $\lim_{\alpha \to 0} 1 - (1 + \exp(\epsilon))\alpha = 1$, $\mathcal{M}$ cannot be (ϵ, δ)-differentially private with any ϵ and $\delta < 1$. □

Before the next theorem, we prove Lemma 1, which is not used in [3].

Lemma 1. *Let $x \in \mathbb{R}^d$, $x \sim N(0, \sigma^2 I)$. The tail probability of the ℓ_1 norm of x obeys*

$$\Pr(\|x\|_1 \ge t) \le \frac{d\sigma^2}{\left(t - \sqrt{2/\pi}d\sigma\right)^2}\left(1 - \frac{2}{\pi}\right). \tag{22}$$

Proof. $\|x\|_1 = \sum_{i=1}^d |x_i| = \sum_{i=1}^d y_i$, where $x_i \sim N(0, \sigma^2)$ and y_i follows the half-normal distribution with variance σ^2.

It is known that $\mathrm{E}[y_i] = \sqrt{2/\pi}\sigma$ and $\mathrm{Var}[y_i] = \sigma^2(1 - 2/\pi)$.

Because y_i are independent, $\mathrm{E}[\|x\|_1] = d\,\mathrm{E}[y_i] = \sqrt{2/\pi}d\sigma$ and $\mathrm{Var}[\|x\|_1] = d\,\mathrm{Var}[y_i] = d\sigma^2(1 - 2/\pi)$.

Setting $a = t - \sqrt{2/\pi}d\sigma$ we have

$$\begin{aligned} \Pr(\|x\|_1 \ge t) &= \Pr\left(\|x\|_1 \ge a + \sqrt{2/\pi}d\sigma\right) \\ &\le \Pr\left(\left|\|x\|_1 - \sqrt{2/\pi}d\sigma\right| \ge a\right) \\ &\le \frac{d\sigma^2}{\left(t - \sqrt{2/\pi}d\sigma\right)^2}\left(1 - \frac{2}{\pi}\right). \end{aligned}$$

where the last inequality follows from Chebyshev's inequality. □

3.1.1 Asymptotic efficiency of Gaussian means

Theorem 3, showing one case of asymptotic efficiency of the Gaussian mechanism, corresponds to [3, Theorem 5], although the proof is somewhat different.

Theorem 3. *(ϵ, δ)-differentially private estimate of the mean of a d-dimensional Gaussian variable x bounded by $\|x_i\|_1 \le B$ in which the Gaussian mechanism is used to perturb the sufficient statistics, is asymptotically efficiently private.*

Proof. Following [3, Theorem 3], it is trivial to show that

$$\|\mu_{DP} - \mu_{NP}\|_1 \le \frac{c}{n}\|\delta\|_1,$$

where $\delta = (\delta_1, \ldots, \delta_d)^T \in \mathbb{R}^D$ with $\delta_j \sim \mathrm{N}\left(0, \sigma_j^2\right)$ holds when we utilize the Gaussian mechanism instead of the Laplace mechanism. This allows us to bound the corresponding tail probabilities by using Lemma 1.

Therefore, given $\alpha > 0$, we can guarantee that

$$\Pr\left\{\|\mu_{DP} - \mu_{NP}\|_1 > \frac{C}{n}\right\} \leq \Pr\left\{\frac{1}{n}\|\delta\|_1 > \frac{C}{n}\right\} = \Pr\{\|\delta\|_1 > C\} < \alpha, \tag{23}$$

by choosing C according to Lemma 1. □

3.2 Asymptotic efficiency of DP linear regression

Theorem 4 that establishes asymptotic efficiency for DP linear regression using the Gaussian mechanism, for the most part follows [3, Theorem 8]. We concentrate here more closely only on the differing parts.

Theorem 4. (ϵ, δ)*-differentially private inference of the posterior mean of the weights of linear regression with the Gaussian mechanism used to perturb the sufficient statistics is asymptotically efficiently private.*

Proof. Following the proof of [3, Theorem 7] with minimal changes we have

$$\begin{aligned}\|\mu_{DP} - \mu_{NP}\|_1 \leq &\left\|(\Lambda_0 + \Lambda(n\overline{xx} + \Delta))^{-1}\Lambda\delta\right\|_1 \\ &+ \left\|\left[\left(\frac{1}{n}\Lambda_0 + \Lambda\left(\overline{xx} + \frac{1}{n}\Delta\right)\right)^{-1} - \left(\frac{1}{n}\Lambda_0 + \Lambda\overline{xx}\right)^{-1}\right]\left(\Lambda\overline{xy} + \frac{1}{n}\Lambda_0\beta_0\right)\right\|_1,\end{aligned} \tag{24}$$

where Δ is the noise contribution from the Gaussian mechanism added to the sufficient statistics $\overline{xx}$ (see Section 2 in this supplement).

As in [3, Theorem 7], the first term can be bounded as

$$\left\|(\Lambda_0 + \Lambda(n\overline{xx} + \Delta))^{-1}\Lambda\delta\right\|_1 \leq \frac{c_1}{n}\left\|(\overline{xx})^{-1}\right\|_1 \|\delta\|_1 \tag{25}$$

where $c_1 > 1$, for large enough n.

As done in the proof of Theorem 3, given $\alpha > 0$, Lemma 1 can be used to ensure that

$$\Pr\left\{\|(\Lambda_0 + \Lambda(n\overline{xx} + \Delta))^{-1}\Lambda\delta\|_1 > \frac{C_1}{n}\right\} < \frac{\alpha}{2}, \tag{26}$$

by choosing a suitable C_1.

Again, following [3, Theorem 7], the second term can be bounded as

$$\begin{aligned}&\left\|\left[\left(\frac{1}{n}\Lambda_0 + \Lambda\left(\overline{xx} + \frac{1}{n}\Delta\right)\right)^{-1} - \left(\frac{1}{n}\Lambda_0 + \Lambda\overline{xx}\right)^{-1}\right]\left(\Lambda\overline{xy} + \frac{1}{n}\Lambda_0\beta_0\right)\right\|_1 \\ &\qquad\leq \frac{c_2}{n}\left\|(\overline{xx})^{-1}\right\|_1 \|\Delta\|_1 \left\|(\overline{xx})^{-1}\right\|_1 \|\overline{xy}\|_1,\end{aligned}$$

where, as in Eq. (25), the bound is valid for $c_2 > 1$ as n gets large enough.

$\|\Delta\|_1$ here is the ℓ_1-norm of the symmetric matrix Δ, that is comprised of a vector of $d(d+1)/2$ unique noise terms, each generated independently from a Normal distribution according to the Gaussian mechanism. Denoting this vector by $\mathbf{v}$, a bound to the matrix norm is given by $\|\Delta\|_1 \leq \|\mathbf{v}\|_1$.

Therefore, given $\alpha > 0$, we can again use Lemma 1 to find a suitable C_2 s.t.

$$\Pr\left\{\|\Delta\|_1 > \frac{C_2}{c_2\left\|(\overline{xx})^{-1}\right\|_1^2 \|\overline{xy}\|_1}\right\} \leq \Pr\left\{\|\mathbf{v}\|_1 > \frac{C_2}{c_2\left\|(\overline{xx})^{-1}\right\|_1^2 \|\overline{xy}\|_1}\right\} < \frac{\alpha}{2}. \tag{27}$$

By combining Eqs. (26) and (27) we get

$$\Pr\left\{\|\mu_{DP} - \mu_{NP}\|_1 > \frac{C_1 + C_2}{n}\right\} < \alpha. \tag{28}$$

□

4 GDSC dataset description

The data were downloaded from the Genomics of Drug Sensitivity in Cancer (GDSC) project, release 6.1, March 2017, http://www.cancerrxgene.org/. We use gene expression and drug sensitivity data. The gene expression dimensionality is reduced to 10 genes used for the actual prediction task, based on prior information about their mutation counts in cancer (we use the same procedure as [3]). The dataset used for learning contains 940 cell lines and drug sensitivity data for 265 drugs. Some of the values are missing, so the actual number of observations varies between the drugs. We use a test set of size 100 and the rest of the available data for learning.

Since we want to focus on the relative expression of the genes, each data point is normalized to have ℓ_2-norm of 1. In the distributed setting this can be done by each client without breaching privacy. After the scaling, we know that all dimensions are bounded by $[-1, 1]$, except the target. For the target dimension, the true range varies between the drugs. The average width of the ranges is 8.6.

We assume a range of [-7.5,7.5] for the marginal std estimation needed for the projection, and use a symmetric bound given by $[-\lceil \max |y| \rceil, \lceil \max |y| \rceil]$ for the non-projected baseline methods (DDP, TA). The exact bound for the baseline methods varies between the drugs while the average is 6.8. In other words, the projected methods add somewhat more extra noise to the results on average. We also tested the performance using a fixed bound for the non-projected methods as with the UCI data, but the results did not change markedly (not included in the paper).

References

[1] P. Diaconis and D. Ylvisaker. Conjugate priors for exponential families. *Ann. Stat.*, 7(2):269–281, 1979.

[2] J. Foulds, J. Geumlek, M. Welling, and K. Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pages 192–201, 2016.

[3] A. Honkela, M. Das, A. Nieminen, O. Dikmen, and S. Kaski. Efficient differentially private learning improves drug sensitivity prediction. 2016. arXiv:1606.02109 [stat.ML].

Paper II

Mikko A. Heikkilä, Joonas Jälkö, Onur Dikmen and Antti Honkela

Differentially Private Markov Chain Monte Carlo

In *Annual Conference on Neural Information Processing Systems (NeurIPS) 2019, December 8-14, 2019, Vancouver, BC, Canada*,
Advances in Neural Information Processing Systems 32, 2019, pages 4115–4125.

Differentially Private Markov Chain Monte Carlo

Mikko A. Heikkilä [*]
Helsinki Institute for Information Technology HIIT, Department of Mathematics and Statistics
University of Helsinki, Helsinki, Finland
mikko.a.heikkila@helsinki.fi

Joonas Jälkö [*]
Helsinki Institute for Information Technology HIIT, Department of Computer Science
Aalto University, Espoo, Finland
joonas.jalko@aalto.fi

Onur Dikmen
Center for Applied Intelligent Systems Research (CAISR)
Halmstad University, Halmstad, Sweden
onur.dikmen@hh.se

Antti Honkela
Helsinki Institute for Information Technology HIIT, Department of Computer Science
University of Helsinki, Helsinki, Finland
antti.honkela@helsinki.fi

Abstract

Recent developments in differentially private (DP) machine learning and DP Bayesian learning have enabled learning under strong privacy guarantees for the training data subjects. In this paper, we further extend the applicability of DP Bayesian learning by presenting the first general DP Markov chain Monte Carlo (MCMC) algorithm whose privacy-guarantees are not subject to unrealistic assumptions on Markov chain convergence and that is applicable to posterior inference in arbitrary models. Our algorithm is based on a decomposition of the Barker acceptance test that allows evaluating the Rényi DP privacy cost of the accept-reject choice. We further show how to improve the DP guarantee through data subsampling and approximate acceptance tests.

1 Introduction

Differential privacy (DP) [Dwork et al., 2006, Dwork and Roth, 2014] and its generalisations to concentrated DP [Dwork and Rothblum, 2016, Bun and Steinke, 2016] and Rényi DP [Mironov, 2017] have recently emerged as the dominant framework for privacy-preserving machine learning. There are DP versions of many popular machine learning algorithms, including highly popular and effective DP stochastic gradient descent (SGD) [Song et al., 2013] for optimisation-based learning.

There has also been a fair amount of work in DP Bayesian machine learning, with the proposed approaches falling to three main categories: i) DP perturbation of sufficient statistics for inference in exponential family models [e.g. Zhang et al., 2016, Foulds et al., 2016, Park et al., 2016, Bernstein and Sheldon, 2018], ii) gradient perturbation similar to DP SGD for stochastic gradient Markov chain

[*] These authors contributed equally to this work.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Monte Carlo (MCMC) and variational inference [e.g. Wang et al., 2015, Jälkö et al., 2017, Li et al., 2019], and iii) DP guarantees for sampling from the exact posterior typically realised using MCMC [e.g. Dimitrakakis et al., 2014, Zhang et al., 2016, Geumlek et al., 2017].

None of these provide fully general solutions: i) sufficient statistic perturbation methods are limited to a restricted set of models, ii) stochastic gradient methods lack theoretical convergence guarantees and are limited to models with continuous variables, iii) posterior sampling methods are applicable to general models, but the privacy is conditional on exact sampling from the posterior, which is usually impossible to verify in practice.

In this paper, we present a new generic DP-MCMC method with strict, non-asymptotic privacy guarantees that hold independently of the chain's convergence. Our method is based on a recent Barker acceptance test formulation [Seita et al., 2017].

1.1 Our contribution

We present the first general-purpose DP MCMC method with a DP guarantee under mild assumptions on the target distribution. We mitigate the privacy loss induced by the basic method through a subsampling-based approximation. We also improve on the existing method of Seita et al. [2017] for subsampled MCMC, resulting in a significantly more accurate method for correcting the subsampling induced noise distribution.

2 Background

2.1 Differential privacy

Definition 1 (Differential privacy)**.** A randomized algorithm $\mathcal{M} : \mathcal{X}^N \to \mathcal{I}$ satisfies (ϵ, δ) differential privacy, if for all adjacent datasets $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^N$ and for all measurable $I \subset \mathcal{I}$ it holds that

$$\Pr(\mathcal{M}(\mathbf{x}) \in I) \leq e^{\epsilon} \Pr(\mathcal{M}(\mathbf{x}') \in I) + \delta. \tag{1}$$

Adjacency here means that $|\mathbf{x}| = |\mathbf{x}'|$, and $\mathbf{x}$ differs from $\mathbf{x}'$ by a single element, e.g. by a single row corresponding to one individual's data in a data matrix.

Recently Mironov [2017] proposed a Rényi divergence [Rényi, 1961] based relaxation for differential privacy called *Rényi differential privacy* (RDP).

Definition 2 (Rényi divergence)**.** Rényi divergence between two distributions P and Q defined over $\mathcal{I}$ is defined as

$$D_\alpha(P \,||\, Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_P \left[\left(\frac{p(X)}{q(X)} \right)^{\alpha - 1} \right]. \tag{2}$$

Definition 3 (Rényi differential privacy)**.** A randomized algorithm $\mathcal{M} : \mathcal{X}^N \to \mathcal{I}$ is (α, ϵ)-RDP, if for all adjacent datasets $\mathbf{x}, \mathbf{x}'$ it holds that

$$D_\alpha(\mathcal{M}(\mathbf{x}) \,||\, \mathcal{M}(\mathbf{x}')) \leq \epsilon \triangleq \epsilon(\alpha). \tag{3}$$

Like DP, RDP has many useful properties such as invariance to post-processing. The main advantage of RDP compared to DP is the theory providing tight bounds for doing adaptive compositions, i.e., for combining the privacy losses from several possibly adaptive mechanisms accessing the same data, and subsampling [Wang et al., 2019]. RDP guarantees can always be converted to (ϵ, δ)-DP guarantees. These existing results are presented in detail in the Supplement.

2.2 Subsampled MCMC using Barker acceptance

The fundamental idea in standard MCMC methods [Brooks et al., 2011] is that a distribution $\pi(\theta)$ that can only be evaluated up to a normalising constant, is approximated by samples $\theta_1, \ldots, \theta_t$ drawn from a suitable Markov chain. Denoting the current parameter values by θ, the next value is generated using a proposal θ' drawn from a proposal distribution $q(\theta'|\theta)$. An acceptance test is used to determine if the chain should move to the proposed value or stay at the current one.

Denoting the acceptance probability by $\alpha(\theta', \theta)$, a test that satisfies detailed balance $\pi(\theta)q(\theta'|\theta)\alpha(\theta', \theta) = \pi(\theta')q(\theta|\theta')\alpha(\theta, \theta')$ together with ergodicity of the chain are sufficient conditions to guarantee asymptotic convergence to the correct invariant distribution $\pi(\theta)$. In Bayesian inference, we are typically interested in sampling from the posterior distribution, i.e., $\pi(\theta) \propto p(\mathbf{x}|\theta)p(\theta)$. However, it is computationally infeasible to use e.g. the standard Metropolis-Hastings (M-H) test [Metropolis et al., 1953, Hastings, 1970] with large datasets, since each iteration would require evaluating $p(\mathbf{x}|\theta)$.

To solve this problem in the non-private setting, Seita et al. [2017] formulate an approximate test that only uses a fraction of the data at each iteration. In the rest of this Section we briefly rephrase their arguments most relevant for our approach without too much details. A more in-depth treatment is then presented in deriving DP MCMC in Section 3.

We start by assuming the data are exchangeable, so $p(\mathbf{x}|\theta) = \prod_{x_i \in \mathbf{x}} p(x_i|\theta)$. Let

$$\Delta(\theta', \theta) = \sum_{x_i \in \mathbf{x}} \log \frac{p(x_i|\theta')}{p(x_i|\theta)} + \log \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}, \tag{4}$$

where we suppress the parameters for brevity in the following, and let $V_{log} \sim \text{Logistic}(0, 1)$. Instead of using the standard M-H acceptance probability $\min\{\exp(\Delta), 1\}$, Seita et al. [2017] use a form of Barker acceptance test [Barker, 1965] to show that testing if

$$\Delta + V_{log} > 0 \tag{5}$$

also satisfies detailed balance. To ease the computational burden, we now want to use only a random subset $S \subset \mathbf{x}$ of size b instead of full data of size N to evaluate acceptance. Let

$$\Delta^*(\theta', \theta) = \frac{N}{b} \sum_{x_i \in S} \log \frac{p(x_i|\theta')}{p(x_i|\theta)} + \log \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)}. \tag{6}$$

Omitting the parameters again, Δ^* is now an unbiased estimator for Δ, and assuming x_i are iid samples from the data distribution, Δ^* has approximately normal distribution by the Central Limit Theorem (CLT).

In order to have a test that approximates the exact full data test (5), we decompose the logistic noise as $V_{log} \simeq V_{norm} + V_{cor}$, where V_{norm} has a normal distribution and V_{cor} is a suitable correction. Relying on the CLT and on this decomposition we write $\Delta^* + V_{cor} \simeq \Delta + V_{norm} + V_{cor} \simeq \Delta + V_{log}$, so given the correction we can approximate the full data exact test using a minibatch.

2.3 Tempering

When the sample size N is very large, one general problem in Bayesian inference is that the posterior includes more and more details. This often leads to models that are much harder to interpret while only marginally more accurate than simpler models (see e.g. Miller and Dunson 2019). One way of addressing this issue is to scale the log-likelihood ratios in (4) and (6), so instead of $\log p(x_i|\theta)$ we would have $\tau \log p(x_i|\theta)$ with some τ. The effect of scaling with $0 < \tau < 1$ is then to spread the posterior mass more evenly. We will refer to this scaling as tempering.

As an interesting theoretical justification for tempering, Miller and Dunson [2019] show a relation between tempered likelihoods and modelling error. The main idea is to take the error between the theoretical pure data and the actual observable data into account in the modelling. Denote the observed data with lowercase and errorless random variables with uppercase letters, and let $R \sim \text{Exp}(\beta)$. Then using empirical KL divergence as our modelling error estimator d_N, instead of the standard posterior we are looking for the posterior conditional on the observed data being close to the pure data, i.e., we want $p(\theta|d_N(x_{1:N}, X_{1:N}) < R)$, which is called coarsened posterior or *c-posterior*.

Miller and Dunson [2019] show that with these assumptions

$$p(\theta|d_N(x_{1:N}, X_{1:N}) < R) \underset{\sim}{\propto} p(\mathbf{x}|\theta)^{\xi_N} p(\theta), \tag{7}$$

where $\underset{\sim}{\propto}$ means approximately proportional to, and $\xi_N = 1/(1+N/\beta)$, i.e., a posterior with tempered likelihoods can be interpreted as an approximate c-posterior.

3 Privacy-preserving MCMC

Our aim is to sample from the posterior distribution of the model parameters while ensuring differential privacy. We start in Section 3.2 by formulating DP MCMC based on the exact full data Barker acceptance presented in Section 2.2. To improve on this basic algorithm, we then introduce subsampling in Section 3.3. The resulting DP subsampled MCMC algorithm has significantly better privacy guarantees as well as computational requirements than the full data version.

3.1 Notation

There are multiple different factors that we use in the privacy analysis. Table 1 includes all the necessary factors used.

Notation	Explanation
α	Parameter for RDP
T	Number of MCMC draws
N	Dataset size
$C \in (0, \pi^2/3)$	Noise variance, in Section 3.3 we set $C = 2$
B	Assumed bound for the log-likelihood ratios (llr) w.r.t. data OR the parameters
$b > 5\alpha$	Batch size for subsampled DP-MCMC
β	Parameter for tempering

Table 1: Table of the notation used in Section 3.

3.2 DP MCMC

To achieve privacy-preserving MCMC, we repurpose the decomposition idea mentioned in Section 2.2 with subsampling, i.e., we decompose V_{log} in the exact test (5) into normal and correction variables. Noting that V_{log} has variance $\pi^2/3$, fix $0 < C < \pi^2/3$ a constant and write

$$V_{log} \simeq \mathcal{N}(0, C) + V_{cor}^{(C)}, \tag{8}$$

where $V_{cor}^{(C)}$ is the correction with variance $\pi^2/3 - C$. Now testing if

$$\mathcal{N}(\Delta, C) + V_{cor}^{(C)} > 0 \tag{9}$$

is approximately equivalent to (5).

Since (8) holds exactly for no known distribution $V_{cor}^{(C)}$ with an analytical expression, Seita et al. [2017] construct an approximation by discretising the convolution implicit in (8), and turning the problem into a ridge regression problem which can be solved easily. Unlike Seita et al. [2017], we aim for preserving privacy. We therefore want to work with relatively large values of C for which the ridge regression based solution does not give a good approximation. Instead, we propose to use a Gaussian mixture model approximation, which gives good empirical performance for larger C as well. The details of the approximation with related discussion can be found in the Supplement.

In practice, if $V_{cor}^{(C)}$ is an approximation, the stationary distribution of the chain might not be the exact posterior. However, when the approximation (8) is good, the accept-reject decisions are rarely affected and we can expect to stay close to the true posterior. Clearly, in the limit of decreasing C we recover the exact test (5). We return to this topic in Section 3.3.

Considering privacy, on each MCMC iteration we access the data only through the log-likelihood ratio Δ in the test (9). To achieve RDP, we therefore need a bound for the Rényi divergence between two Gaussians $\mathcal{N}_{\mathbf{x}} = \mathcal{N}(\Delta_{\mathbf{x}}, C)$ and $\mathcal{N}_{\mathbf{x}'} = \mathcal{N}(\Delta_{\mathbf{x}'}, C)$ corresponding to neighbouring datasets $\mathbf{x}, \mathbf{x}'$. The following Lemma states the Rényi divergence between two Gaussians:

Lemma 1. *Rényi divergence between two normals $\mathcal{N}_1$ and $\mathcal{N}_2$ with parameters μ_1, σ_1 and μ_2, σ_2 respectively is*

$$D_\alpha(\mathcal{N}_1 \,||\, \mathcal{N}_2) = \ln \frac{\sigma_2}{\sigma_1} + \frac{1}{2(\alpha - 1)} \ln \frac{\sigma_2^2}{\sigma_\alpha^2} + \frac{\alpha}{2} \frac{(\mu_1 - \mu_2)^2}{\sigma_\alpha^2}, \tag{10}$$

where $\sigma_\alpha^2 = \alpha\sigma_2^2 + (1 - \alpha)\sigma_1^2$.

Proof. See [Gil et al., 2013] Table 2. □

Theorem 1. *Assume either*

$$|\log p(x_i \mid \theta') - \log p(x_i \mid \theta)| \le B \tag{11}$$

or

$$|\log p(x_i \mid \theta) - \log p(x_j \mid \theta)| \le B, \tag{12}$$

for all x_i, x_j and for all θ, θ'. Releasing a result of the accept/reject decision from the test (9) *is* (α, ϵ)*-RDP with* $\epsilon = 2\alpha B^2/C$.

Proof. Follows from Lemma 1. See Supplement for further details. □

Using the composition property of RDP (see Supplement), it is straightforward to get the following Corollary for the whole chain:

Corollary 1. *Releasing an MCMC chain of T iterations, where at each iteration the accept-reject decision is done using the test* (9)*, satisfies* (α, ϵ')*-RDP with* $\epsilon' = T2\alpha B^2/C$.

We can satisfy the condition (11) with sufficiently smooth likelihoods and a proposal distribution with a bounded domain:

Lemma 2. *Assuming the model log-likelihoods are L-Lipschitz over θ and the diameter of the proposal distribution domain is bounded by d_θ, LHS of* (11) *is bounded by Ld_θ.*

Proof.

$$|\log p(x_i \mid \theta) - \log p(x_i \mid \theta')| \le L|\theta - \theta'| \le Ld_\theta. \tag{13}$$

□

Clearly, when $Ld_\theta \le B$ we satisfy the condition in Equation (11).

For some models, using a proposal distribution with a bounded domain could affect the ergodicity of the chain. Considering models that are not Lipschitz or using an unbounded proposal distribution, we can also satisfy the boundedness condition (11) by clipping the log-likelihood ratios to a suitable interval.

3.3 DP subsampled MCMC

In Section 3.2 we showed that we can release samples from the MCMC algorithm under privacy guarantees. However, as already discussed, evaluating the log-likelihood ratios might require too much computation with large datasets. Using the full dataset in the DP MCMC setting might also be infeasible for privacy reasons: the noise variance C in Theorem 1 is upper-bounded by the variance of the logistic random variable, and thus working under a strict privacy budget we might be able to run the chain for only a few iterations before ϵ' in Corollary 1 exceeds our budget. Using only a subsample S of the data at each MCMC iteration allows us to reduce not only the computational cost but also the privacy cost through privacy amplification [Wang et al., 2019].

As stated in Section 2.2, for the subsampled variant according to the CLT we have

$$\Delta^* = \Delta + \tilde{V}_{norm}, \tag{14}$$

where $\tilde{V}_{norm}$ is approximately normal with some variance $\sigma^2_{\Delta^*}$. Assuming

$$\sigma^2_{\Delta^*} < C < \pi^2/3 \tag{15}$$

for some constant C, we now reformulate the decomposition (8) as

$$V_{log} \simeq \underbrace{V_{norm} + V_{nc}}_{\sim\mathcal{N}(0,C)} + V_{cor}^{(C)}, \tag{16}$$

where $V_{norm} \sim \mathcal{N}(0, \sigma^2_{\Delta^*})$ and $V_{nc} \sim \mathcal{N}(0, C - \sigma^2_{\Delta^*})$. We can now write

$$\Delta^* + V_{nc} + V_{cor}^{(C)} \simeq \Delta + V_{norm} + V_{nc} + V_{cor}^{(C)} \simeq \Delta + V_{log}, \tag{17}$$

where the first approximation is justified by the CLT, and the second by the decomposition (16). Therefore, testing if

$$\mathcal{N}(\Delta^*, C - \sigma^2_{\Delta^*}) + V^{(C)}_{cor} > 0 \tag{18}$$

approximates the exact full data test (5).

As in Section 3.2, the approximations used for arriving at the test (18) imply that the stationary distribution of the chain need not be the exact posterior. However, we can expect to stay close to the true posterior when the approximations are good, since the result only changes if the binary accept-reject decision is affected. This is exemplified by the testing in Section 4 (see also Seita et al. 2017). The quality of the first approximation in (17) depends on the batch size b, which should not be too small. As for the second error source, as already noted in Section 3.2 we markedly improve on this with the GMM based approximation, and the resulting error is typically very small (see Supplement). In some cases there are known theoretical upper bounds for the total error w.r.t. the true posterior. These bounds are of limited practical value since they rely on assumptions that can be hard to meet in general, and we therefore defer them to the Supplement.

For privacy, similarly as in Section 3.2, in (18) we need to access the data only for calculating $\Delta^* + V_{nc}$. Thus, it suffices to privately release a sample from $\mathcal{N}_S = \mathcal{N}(\Delta^*_{\mathbf{x}}, C - s^2_{\Delta^*_{\mathbf{x}}})$, where $s^2_{\Delta^*_{\mathbf{x}}}$ denotes the sample variance when sampling from dataset $\mathbf{x}$, i.e., we need to bound the Rényi divergence between $\mathcal{N}_S$ and $\mathcal{N}_{S'}$. We use noise variance $C = 2$ in the following analysis.

Next, we will state our main theorem giving an explicit bound that can be used for calculating the privacy loss for a single MCMC iteration with subsampling:

Theorem 2. *Assuming*

$$|\log p(x_i|\theta') - \log p(x_i|\theta)| \leq \frac{\sqrt{b}}{N}, \tag{19}$$

$$\alpha < \frac{b}{5}, \tag{20}$$

where b is the size of the minibatch S and N is the dataset size, releasing a sample from $\mathcal{N}_S$ satisfies (α, ϵ)-RDP with

$$\epsilon = \frac{5}{2b} + \frac{1}{2(\alpha-1)} \ln \frac{2b}{b - 5\alpha} + \frac{2\alpha}{b - 5\alpha}. \tag{21}$$

Proof. The idea of the proof is straightforward: we need to find an upper bound for each of the terms in Lemma 1, which can be done using standard techniques. Note that for $C = 2$, (19) implies that the variance assumption (15) holds. See Supplement for the full derivation. □

Using the composition [Mironov, 2017] and subsampling amplification [Wang et al., 2019] properties of Rényi DP (see Supplement), we immediately get the following:

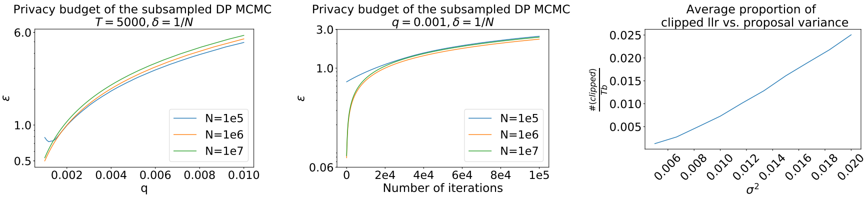
Corollary 2. *Releasing a chain of T subsampled MCMC iterations with sampling ratio q, each satisfying $(\alpha, \epsilon(\alpha))$-RDP with $\epsilon(\alpha)$ from Theorem 2, is $(\alpha, T\epsilon')$-RDP with*

$$\epsilon' = \frac{1}{\alpha - 1} \log \Big(1 + q^2 \binom{\alpha}{2} \min\{4(e^{\epsilon(2)} - 1), 2e^{\epsilon(2)}\} + 2\sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon(j)}\Big). \tag{22}$$

Figures 1(a) and 1(b) illustrate how changing the parameters q and T in Corollary 2 will affect the privacy budget of DP MCMC.

Similarly as in the full data case in Section 3.2, we can satisfy the condition (19) with sufficiently smooth likelihoods or by clipping. Figure 1(c) shows how frequently we need to clip the log-likelihood ratios to maintain the bound in (19) as a function of proposal variance using a Gaussian mixture model problem defined in Section 4. Using smaller proposal variance will result in smaller changes in the log-likelihoods between the previous and the proposed parameter values, which entails fewer clipped values.

However, the bound in (19) gets tighter with increasing N. To counterbalance this, either the proposals need to be closer to the current value (assuming suitably smooth log-likelihood), resulting



(a) Privacy vs. subsampling ratio (b) Privacy vs. iterations (c) Clipping vs. proposal variance

Figure 1: Parameter effects. Calculating total privacy budget from Corollary 2 for different dataset sizes: in Figure 1(a) as a function of subsampling ratio, and in Figure 1(b) as a function of number of iterations. Figure 1(c) shows the proportion of clipped log-likelihood ratios as a function of proposal variance for the GMM example detailed in Section 4.

in a slower mixing chain, or b needs to increase, affecting privacy amplification. For very large N we would therefore like to temper the log-likelihood ratios in a way that we could use sufficiently small batches to benefit from privacy amplification, while still preserving sufficient amount of information from the likelihoods and reasonable mixing properties. Using the c-posterior discussed in Section 2.3 with parameter β s.t. $N_0 = N\beta/(\beta + N)$, instead of condition (19) we then require

$$|\log p(x_i|\theta') - \log p(x_i|\theta)| \leq \frac{\sqrt{b}}{N_0}, \tag{23}$$

which does not depend on N.

4 Experiments

In order to demonstrate our proposed method in practice, we use a simple 2-dimensional Gaussian mixture model[2], that has been used by Welling and Teh [2011] and Seita et al. [2017] in the non-private setting:

$$\theta_j \sim \mathcal{N}(0, \sigma_j^2,), \quad j = 1, 2 \tag{24}$$

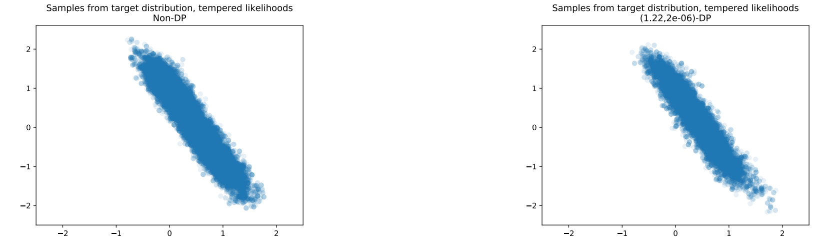
$$x_i \sim 0.5 \cdot \mathcal{N}(\theta_1, \sigma_x^2) + 0.5 \cdot \mathcal{N}(\theta_1 + \theta_2, \sigma_x^2), \tag{25}$$

where $\sigma_1^2 = 10, \sigma_2^2 = 1, \sigma_x^2 = 2$. For the observed data, we use fixed parameter values $\theta = (0, 1)$. Following Seita et al. [2017], we generate 10^6 samples from the model to use as training data. We use $b = 1000$ for the minibatches, and adjust the temperature of the chain s.t. $N_0 = 100$ in (23). This corresponds to the temperature used by Seita et al. [2017] in their non-private test.

If we have absolutely no idea of a good initial range for the parameter values, especially in higher dimensions the chain might waste the privacy budget in moving towards areas with higher posterior probability. In such cases we might want to initialise the chain in at least somewhat reasonable location, which will cost additional privacy. To simulate this effect, we use the differentially private variational inference (DPVI) introduced by Jälkö et al. [2017] with a small privacy budget $(0.22, 10^{-6})$ to find a rough estimate for the initial location.

As shown in Figure 2, the samples from the tempered chain with DP are nearly indistinguishable from the samples drawn from the non-private tempered chain. We also compared our method against DP stochastic gradient Langevin dynamics (DP SGLD) method of Li et al. [2019]. Figure 3 illustrates how the accuracy is affected by privacy. Posterior means and variances are computed from the first t iterations of the private chain alongside the privacy cost ϵ, which increases with t. The baseline is given by a non-private chain after 40000 iterations. The plots show the mean and the standard error of the mean over 20 runs of 20 000 iterations with DP MCMC and 6 000 000 with DP SGLD. The DP MCMC method was burned in for 1 000 iterations and DP SGLD for 100 000 iterations.

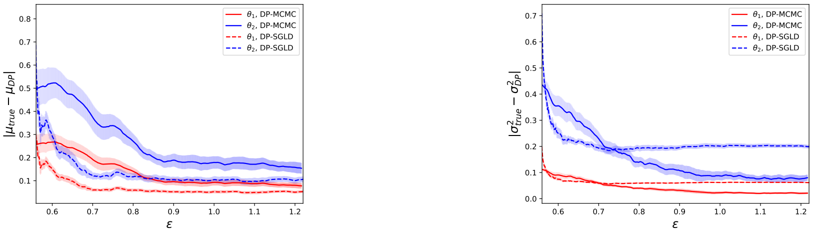
[2] The code for running all the experiments is avalaible in `https://github.com/DPBayes/DP-MCMC-NeurIPS2019`.



(a) Samples without privacy

(b) Samples with DP

Figure 2: Results for the GMM experiment with tempered likelihoods: 2(a) shows 40000 samples from the chain without privacy and 2(b) 20000 samples with privacy. The results with strict privacy are very close to the non-private results.



(a) Posterior mean accuracy

(b) Posterior variance accuracy

Figure 3: Intermediate private posterior statistics from DP SGLD and DP MCMC compared against the baseline given by a non-private chain after 40000 iterations. Lines showing the mean error between 20 runs of the algorithm with errorbars illustrating the standard error of the mean between the runs. DP SGDL converges quickly towards the posterior mean, but does not properly capture posterior variance.

5 Related work

Bayesian posterior sampling under DP has been studied using several different approaches. Recently Yıldırım and Ermiş [2019] proposed a method for drawing samples from exact posterior under DP using a modified MH algorithm. However their solution does not include subsampling and thus suffers the computational cost of the full likelihood. Dimitrakakis et al. [2014] note that drawing a single sample from the posterior distribution of a model where the log-likelihood is Lipschitz or bounded yields a DP guarantee. The bound on ϵ can be strengthened by tempering the posterior by raising the likelihood to a power $\tau \in (0, 1)$ to obtain the tempered posterior

$$\pi_\tau(\theta) \propto p(\theta)p(\mathbf{x} \mid \theta)^\tau. \tag{26}$$

The same principle is discussed and extended by Wang et al. [2015], Zhang et al. [2016] and Dimitrakakis et al. [2017] in the classical DP setting and by Geumlek et al. [2017] in the RDP setting. Wang et al. [2015] dub this the "one posterior sample" (OPS) mechanism. The main limitation of all these methods is that the privacy guarantee is conditional on sampling from the exact posterior, which is in most realistic cases impossible to verify.

The other most widely used approach for DP Bayesian inference is perturbation of sufficient statistics of an exponential family model using the Laplace mechanism. This straightforward application of the Laplace mechanism was mentioned at least by Dwork and Smith [2009] and has been widely applied since by several authors [e.g. Zhang et al., 2016, Foulds et al., 2016, Park et al., 2016, Honkela et al., 2018, Bernstein and Sheldon, 2018]. In particular, Foulds et al. [2016] show that the sufficient statistics perturbation is more efficient than OPS for models where both are applicable. Furthermore, these methods can provide an unconditional privacy guarantee. Many of the early methods ignore the Laplace noise injected for DP in the inference, leading to potentially biased inference results. This weakness is addressed by Bernstein and Sheldon [2018], who include the uncertainty arising from the injected noise in the modelling, which improves especially the accuracy of posterior variances for models where this can be done.

MCMC methods that use gradient information such as Hamiltonian Monte Carlo (HMC) and various stochastic gradient MCMC methods have become popular recently. DP variants of these were first proposed by Wang et al. [2015] and later refined by Li et al. [2019] to make use of the moments accountant [Abadi et al., 2016]. The form of the privacy guarantee for these methods is similar to that of our method: there is an unconditional guarantee for models with a differentiable Lipschitz log-likelihood that weakens as more iterations are taken. Because of the use of the gradients, these methods are limited to differentiable models and cannot be applied to e.g. models with discrete variables.

Before Seita et al. [2017], the problem of MCMC without using the full data has been considered by many authors (see Bardenet et al. 2017 for a recent literature survey). The methods most closely related to ours are the ones by Korattikara et al. [2014] and Bardenet et al. [2014]. From our perspective, the main problem with these approaches is the adaptive batch size: the algorithms may regularly need to use all observations on a single iteration [Seita et al., 2017], which clashes with privacy amplification. Bardenet et al. [2017] have more recently proposed an improved version of their previous technique alleviating the problem, but the batch sizes can still be large for privacy amplification.

6 Discussion

While gradient-based samplers such as HMC are clearly dominant in the non-DP case, it is unclear how useful they will be under DP. Straightforward stochastic gradient methods such as stochastic gradient Langevin dynamics (SGLD) can be fast in initial convergence to a high posterior density region, but it is not clear if they can explore that region more efficiently. We can see this in Figure 3: the gradient adjusted method rapidly converges close to posterior mean, but the posterior variance is not captured. HMC does have a clear advantage at exploration, but Betancourt [2015] demonstrates that HMC is very sensitive to having accurate gradients and therefore a naive DP HMC is unlikely to perform well. We believe that using a gradient-based method such as DP variational inference [Jälkö et al., 2017] as an initiasation for the proposed method can yield overall a very efficient sampler that can take advantage of the gradients in the initial convergence and of MCMC in obtaining accurate posterior variances. Further work in benchmarking different approaches over a number of models is needed, but it is beyond the scope of this work.

The proposed method allows for structurally new kind of assumptions to guarantee privacy through forcing bounds on the proposal instead of or in addition to the likelihood. This opens the door for a lot of optimisation in the design of the proposal. It is not obvious how the proposal should be selected in order to maximise the amount of useful information obtained about the posterior under the given privacy budget, when one has to balance between sampler acceptance rate and autocorrelation as well as privacy. We leave this interesting question for future work.

Acknowledgements

The authors would like to thank Daniel Seita for sharing the original code for their paper.

This work has been supported by the Academy of Finland [Finnish Center for Artificial Intelligence FCAI and grants 294238, 303815, 313124].

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 308–318, New York, NY, USA, 2016. ACM.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.*, 18(1):1515–1557, January 2017.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 405–413, Bejing, China, 22–24 Jun 2014. PMLR.

A. A. Barker. Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18:119, April 1965.

Garrett Bernstein and Daniel R Sheldon. Differentially private Bayesian inference for exponential families. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2924–2934. Curran Associates, Inc., 2018.

Michael Betancourt. The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 533–540, Lille, France, 07–09 Jul 2015. PMLR.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 1 edition, 2011.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 635–658, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Robust and private Bayesian inference. In *ALT 2014*, volume 8776 of *Lecture Notes in Computer Science*, pages 291–305. Springer Science + Business Media, 2014.

Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Differential privacy for Bayesian inference through posterior sampling. *Journal of Machine Learning Research*, 18(11):1–39, 2017.

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

Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. March 2016. arXiv:1603.01887.

Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2):135–154, 2009.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings*, pages 265–284. Springer Berlin Heidelberg, 2006.

James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pages 192–201, Arlington, Virginia, United States, March 2016. AUAI Press.

Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. Rényi differential privacy mechanisms for posterior sampling. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5295–5304, USA, 2017. Curran Associates Inc.

Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Antti Honkela, Mrinal Das, Arttu Nieminen, Onur Dikmen, and Samuel Kaski. Efficient differentially private learning improves drug sensitivity prediction. *Biology Direct*, 13(1):1, 2018.

Joonas Jälkö, Antti Honkela, and Onur Dikmen. Differentially private variational inference for non-conjugate models. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 181–189, Bejing, China, 22–24 Jun 2014. PMLR.

Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient MCMC and differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 557–566, 2019.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Jeffrey W. Miller and David B. Dunson. Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 114(527):1113–1125, 2019.

Ilya Mironov. Rényi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

Mijung Park, James Foulds, Kamalika Chaudhuri, and Max Welling. Variational Bayes in private settings (VIPS). *arXiv preprint arXiv:1611.00340*, 2016.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, Berkeley, Calif., 1961. University of California Press.

Daniel Seita, Xinlei Pan, Haoyu Chen, and John F. Canny. An efficient minibatch acceptance test for Metropolis–Hastings. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2493–2502, Lille, France, 07–09 Jul 2015. PMLR.

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

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, USA, 2011. Omnipress.

Sinan Yıldırım and Beyza Ermiş. Exact MCMC with differentially private moves. *Statistics and Computing*, 29(5):947–963, Sep 2019. ISSN 1573-1375.

Zuhe Zhang, Benjamin I. P. Rubinstein, and Christos Dimitrakakis. On the differential privacy of Bayesian inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2365–2371. AAAI Press, 2016.

Supplement to Differentially Private Markov Chain Monte Carlo

May 23, 2019

1 Useful differential privacy results

Proposition 1. A composition of two RDP algorithms $\mathcal{M}_1$, $\mathcal{M}_2$ with RDP guarantees (α, ϵ_1) and (α, ϵ_2), is $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.

Proof. See Mironov [2017, Proposition 1] . □

The next result follows immediately from Proposition 1.

Corollary 1. *Releasing a result from a T-fold composition of a (α, ϵ)-RDP query is $(\alpha, T\epsilon)$-RDP.*

The following Proposition states the privacy amplification via subsampling result of Wang et al. [2019].

Proposition 2. A randomised algorithm $\mathcal{M}$ which accesses the whole dataset $\mathbf{x}$ only through subset S of the dataset and satisfies (α, ϵ)-RDP w.r.t. to S, is (α, ϵ')-RDP with

$$\begin{aligned}\epsilon' \leq \frac{1}{\alpha - 1} \log \Bigg(1 + q^2 \binom{\alpha}{2} \cdot \min \Big\{4(e^{\epsilon(2)} - 1), e^{\epsilon(2)} \min \Big\{2, (e^{\epsilon(\infty) - 1})^2\Big\}\Big\} \\ + \sum_{j=3}^{\alpha} q^j \binom{\alpha}{j} e^{(j-1)\epsilon(j)} \min \left\{2, (e^{\epsilon(\infty)} - 1)^j\right\} \Bigg),\end{aligned}$$

where $q = |S|/|\mathbf{x}|$, and $\alpha \geq 2$ is an integer, and $\epsilon(\infty) = \lim_{j \to \infty} \epsilon(j)$.

Proof. See Wang et al. [2019, Theorem 10] . □

Finally, we can convert RDP privacy guarantees back to (ϵ, δ)-DP guarantees using the following proposition.

Proposition 3. An (α, ϵ)-RDP algorithm $\mathcal{M}$ also satisfies (ϵ', δ)-DP for all $0 < \delta < 1$ with

$$\epsilon' = \epsilon + \frac{\log(1/\delta)}{\alpha - 1}. \tag{1}$$

Proof. See Mironov [2017, Proposition 3] . □

2 Proof of main text's Theorem 1

Denote the maximally different adjacent datasets by $\mathbf{x}_1, \mathbf{x}_2$. The mechanism releases a sample from $\mathcal{N}_1 = \mathcal{N}(\Delta_1, C)$, and $\mathcal{N}_2 = \mathcal{N}(\Delta_2, C)$, where Δ_1, Δ_2 are calculated with $\mathbf{x}_1, \mathbf{x}_2$, respectively.

We want to show that

$$D_\alpha(\mathcal{N}_1 || \mathcal{N}_2) = \log \frac{\sigma_1}{\sigma_2} + \frac{1}{2(\alpha-1)} \log \frac{\sigma_2^2}{\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2} + \frac{\alpha}{2} \frac{(\mu_1 - \mu_2)^2}{\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2} \tag{2}$$

$$\leq \frac{2\alpha B^2}{C} \tag{3}$$

assuming that either

$$|\log p(x_i|\theta') - \log p(x_i|\theta)| < B \ \forall x_i, \theta, \theta' \tag{4}$$

or

$$|\log p(x_i|\theta) - \log p(x_j|\theta)| < B, \ \forall x_i, x_j, \theta. \tag{5}$$

Proof. W.l.o.g., we can assume that the differing element between $\mathbf{x}_1$ and $\mathbf{x}_2$ is the final one, so $x_{1,i} = x_{2,i}, i = 1, \ldots, N-1$.

Since $\sigma_1^2 = \sigma_2^2 = C$, we immediately have

$$D_\alpha(\mathcal{N}_1 || \mathcal{N}_2) = \log \frac{\sigma_1}{\sigma_2} + \frac{1}{2(\alpha-1)} \log \frac{\sigma_2^2}{\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2} + \frac{\alpha}{2} \frac{(\mu_1 - \mu_2)^2}{\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2} \tag{6}$$

$$= \frac{\alpha}{2C}(\mu_1 - \mu_2)^2 \tag{7}$$

$$= \frac{\alpha}{2C}[\sum_{i=1}^{N} \log \frac{p(x_{1,i}|\theta')}{p(x_{1,i}|\theta)} - \sum_{i=1}^{N} \log \frac{p(x_{2,i}|\theta')}{p(x_{2,i}|\theta)}]^2 \tag{8}$$

$$= \frac{\alpha}{2C} \left| \log \frac{p(x_{1,N}|\theta')}{p(x_{1,N}|\theta)} - \log \frac{p(x_{2,N}|\theta')}{p(x_{2,N}|\theta)} \right|^2. \tag{9}$$

Assuming (4), and continuing from (9)

$$\frac{\alpha}{2C} \left| \log \frac{p(x_{1,N}|\theta')}{p(x_{1,N}|\theta)} - \log \frac{p(x_{2,N}|\theta')}{p(x_{2,N}|\theta)} \right|^2 \tag{10}$$

$$\leq \frac{\alpha}{2C} \Bigg(\left| \log \frac{p(x_{1,N}|\theta')}{p(x_{1,N}|\theta)} \right| + \left| \log \frac{p(x_{2,N}|\theta')}{p(x_{2,N}|\theta)} \right| \Bigg)^2 \tag{11}$$

$$\leq \frac{\alpha}{2C} |2B|^2 \tag{12}$$

$$\leq \frac{2\alpha B^2}{C}. \tag{13}$$

On the other hand, assuming (5), and again continuing from (9) gives

$$\frac{\alpha}{2C}\left|\log\frac{p(x_{1,N}|\theta')}{p(x_{1,N}|\theta)}-\log\frac{p(x_{2,N}|\theta')}{p(x_{2,N}|\theta)}\right|^2 \tag{14}$$

$$=\frac{\alpha}{2C}\left|\log\frac{p(x_{1,N}|\theta')}{p(x_{2,N}|\theta')}-\log\frac{p(x_{1,N}|\theta)}{p(x_{2,N}|\theta)}\right|^2 \tag{15}$$

$$\leq\frac{\alpha}{2C}\left(\left|\log\frac{p(x_{1,N}|\theta')}{p(x_{2,N}|\theta')}\right|+\left|\log\frac{p(x_{1,N}|\theta)}{p(x_{2,N}|\theta)}\right|\right)^2 \tag{16}$$

$$\leq\frac{\alpha}{2C}\left|2B\right|^2 \tag{17}$$

$$\leq\frac{2\alpha B^2}{C}, \tag{18}$$

which is the same bound as before.

□

3 Proof of main text's Theorem 2

The Barker test amounts to checking the following condition:

$$\Delta^* + V_{nc} + V_{cor}^{(2)} > 0, \text{ where} \tag{19}$$

$$\Delta^* = \frac{N}{b}\sum_{i\in S}\underbrace{\log\frac{p(x_i|\theta')}{p(x_i|\theta)}}_{r_i}+\log\frac{q(\theta|\theta')p(\theta)}{q(\theta'|\theta)p(\theta')}, \tag{20}$$

$$V_{nc}\sim\mathcal{N}(0,2-s^2_{\Delta^*}), \tag{21}$$

N is the full dataset size, b is the batch size, $s^2_{\Delta^*}$ is the sample variance, and summation over S here means summing over the elements in the batch, indexed by the element number i.

In other words, with a slight abuse of notation and writing capital letters for random variables the mechanism releases a sample from

$$\mathcal{N}(N\bar{\mathbf{r}},2-\text{Var}(\frac{N}{b}\sum_{i\in S}R_i)) =\mathcal{N}(N\bar{\mathbf{r}},2-\frac{N^2}{b^2}\sum_{i\in S}\text{Var}(R)) \tag{22}$$

$$\approx\mathcal{N}(N\bar{\mathbf{r}},2-\frac{N^2}{b}\text{Var}(\mathbf{r})), \tag{23}$$

where (22) holds because R_i are conditionally iid with a common distribution written as R, and Var($\mathbf{r}$) means the sample variance estimated from the actual iid sample $r_i, i\in S$ we have, i.e., a vector of length b.

Assume that

$$|r_i|\leq\frac{\sqrt{b}}{N}\triangleq c,\forall\, i \text{ and} \tag{24}$$

$$\alpha<\frac{b}{5}. \tag{25}$$

We want to show that

$$D_\alpha(\mathcal{N}_1\,||\,\mathcal{N}_2) = \underbrace{\ln\frac{\sigma_2}{\sigma_1}}_{f_1} + \underbrace{\frac{1}{2(\alpha-1)}\ln\frac{\sigma_2^2}{\alpha\sigma_2^2+(1-\alpha)\sigma_1^2}}_{f_2} + \underbrace{\frac{\alpha}{2}\frac{(\mu_1-\mu_2)^2}{\alpha\sigma_2^2+(1-\alpha)\sigma_1^2}}_{f_3} \tag{26}$$

$$\leq \frac{5}{2b} + \frac{1}{2(\alpha-1)}\ln\frac{2b}{b-5\alpha} + \frac{2\alpha}{b-5\alpha}. \tag{27}$$

Proof. As a first step, we have

$$0 < \text{Var}(\mathbf{r}) = \mathbb{E}(\mathbf{r}^2) - \mathbb{E}(\mathbf{r})^2 \leq \mathbb{E}(\mathbf{r}^2) = 1/b\sum_{i\in S} r_i^2 \leq \frac{b}{N^2} \tag{28}$$

$$\Rightarrow 2 - \frac{N^2}{b}\text{Var}(\mathbf{r}) \in [1,2), \tag{29}$$

where the last inequality in (28) follows from (24).

Denote the maximally different adjacent datasets as $\mathbf{r}_1, \mathbf{r}_2$ that produce draws from $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively, parameterised with means and variances as in (23). W.l.o.g., we can assume that the differing element is the final one, so we have $r_{1,i} = r_{2,i}, i = 1, \ldots, b-1$. We write $i \in S \setminus x_N$ to index a summation over the batch omitting the differing element.

The proof proceeds by bounding each of the terms f_1, f_2, f_3 in (26).

To start with, f_1 can be bounded as follows:

$$f_1 = \frac{1}{2}\ln\frac{\sigma_2^2}{\sigma_1^2} \leq \frac{1}{2}|\ln\frac{\sigma_2^2}{\sigma_1^2}| \leq \frac{1}{2}|\sigma_2^2 - \sigma_1^2| \tag{30}$$

$$= \frac{1}{2}|2 - \frac{N^2}{b}\text{Var}(\mathbf{r}_2) - (2 - \frac{N^2}{b}\text{Var}(\mathbf{r}_1))| \tag{31}$$

$$= \frac{N^2}{2b}|1/b\sum_{i\in S} r_{1,i}^2 - (\bar{\mathbf{r}}_1)^2 - 1/b\sum_{i\in S} r_{2,i}^2 + (\bar{\mathbf{r}}_2)^2| \tag{32}$$

$$= \frac{N^2}{2b}|1/b(r_{1,b}^2 - r_{2,b}^2) + (1/b\sum_{i\in S} r_{2,i})^2 - (1/b\sum_{i\in S} r_{1,i})^2| \tag{33}$$

$$= \frac{N^2}{2b^2}|(r_{1,b}^2 - r_{2,b}^2) + 1/b(r_{2,b}^2 - r_{1,b}^2 + 2(\sum_{i\in S\setminus x_N} r_{2,i}\cdot r_{2,b} - \sum_{i\in S\setminus x_N} r_{1,i}\cdot r_{1,b}))| \tag{34}$$

$$= \frac{N^2}{2b^2}|\frac{b-1}{b}(r_{1,b}^2 - r_{2,b}^2) - \frac{2}{b}(\sum_{i\in S\setminus x_N} r_i)(r_{1,b} - r_{2,b})| \tag{35}$$

$$= \frac{N^2}{2b^3}|(b-1)(r_{1,b}^2 - r_{2,b}^2) - 2(\sum_{i\in S\setminus x_N} r_i)(r_{1,b} - r_{2,b})| \tag{36}$$

$$\leq \frac{N^2}{2b^3}[(b-1)(c^2) + 2(b-1)c(2c)] \tag{37}$$

$$= \frac{N^2}{2b^3}(b-1)5c^2 \tag{38}$$

$$\leq \frac{5}{2b}, \tag{39}$$

where the final inequality in (30) holds because we have (29), and (37) as well as the final bound in (39) follow from (24).

For the common denominator term $\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2$ in f_2 and f_3, we can first repeat essentially the previous calculation to get

$$\begin{aligned}
\sigma_2^2 - \sigma_1^2 &\geq -|\sigma_2^2 - \sigma_1^2| && (40)\\
&= \cdots && (41)\\
&= -\frac{N^2}{b^3}|(b-1)(r_{1,b}^2 - r_{2,b}^2) - 2(\sum_{i\in S\setminus x_N} r_i)(r_{1,b} - r_{2,b})| && (42)\\
&\geq -\frac{N^2}{b^3}[(b-1)c^2 + 2(b-1)c(2c)] && (43)\\
&= -\frac{N^2}{b^3}(b-1)5c^2 && (44)\\
&\geq -\frac{5}{b}. && (45)
\end{aligned}$$

Combining (45) and (29) we get

$$\begin{aligned}
\alpha\sigma_2^2 + (1-\alpha)\sigma_1^2 &= \sigma_1^2 + \alpha(\sigma_2^2 - \sigma_1^2) && (46)\\
&\geq 1 - \alpha\frac{5}{b} > 0, && (47)
\end{aligned}$$

where the final inequality follows from (25).

For the numerator in f_3 we have

$$\begin{aligned}
(\mu_1 - \mu_2)^2 &= \left(\frac{N}{b}\sum_{i\in S} r_{1,i} - \frac{N}{b}\sum_{i\in S} r_{2,i}\right)^2 && (48)\\
&= \left(\frac{N}{b}(r_{1,b} - r_{2,b})\right)^2 && (49)\\
&\leq \left(\frac{2Nc}{b}\right)^2 && (50)\\
&\leq \frac{4}{b}. && (51)
\end{aligned}$$

Finally, using the derived bounds in (39), (47), and (51) with the fact that $\sigma_2^2 \leq 2$ from (29), the bound for the Rényi divergence (26) becomes

$$\begin{aligned}
D_\alpha(\mathcal{N}_1 \,||\, \mathcal{N}_2) &\leq \frac{5}{2b} + \frac{1}{2(\alpha-1)}(\ln 2 - \ln(1 - \frac{5\alpha}{b})) + \frac{\alpha}{2}\frac{4}{b}\frac{1}{1-\frac{5\alpha}{b}} && (52)\\
&\leq \frac{5}{2b} + \frac{1}{2(\alpha-1)}\ln\frac{2b}{b-5\alpha} + \frac{2\alpha}{b-5\alpha}. && (53)
\end{aligned}$$

If we instead use the tempered log-likelihoods with temperature $\tau = \frac{N_0}{N}$, the effect is to replace r_i by τr_i. The same proof then holds when instead of N we write N_0.

□

4 Bounding the approximations errors

As mentioned in the main text, with finite data and $b < N$ the acceptance test (18) in the main text is an approximation. For this case, there are some known theoretical bounds for the errors induced. The general idea with the following Theorems is that by bounding the errors induced by each approximation step, we can find a bound on the error in the stationary distribution of the approximate chain w.r.t. the exact posterior. The references in this Section mostly point to the main text. The exceptions are obvious from the context.

First, Theorem 1 gives an upper bound for the error due to Δ^* having approximately normal instead of exactly normal distribution as in (20):

Theorem 1.

$$\sup_y |\mathbb{P}(\Delta^* < y) - \Phi(\frac{y-\Delta}{s_{\Delta^*}})| \leq \frac{6.4\mathbb{E}[|Z|^3] + 2\mathbb{E}[|Z|]}{\sqrt{b}},$$

where $Z = N(\log \frac{p(X|\theta')}{p(X|\theta)} - \mathbb{E}[\log \frac{p(X|\theta')}{p(X|\theta)}])$.

Proof. See [Seita et al., 2017, Cor. 1] . □

Next, we have a bound for the error in the test quantity (18) relative to the exact test (5) given in Theorem 2. The original proof [Seita et al., 2017, Cor. 2] assumes that $C = 1$ and (16) holds exactly. We present a slightly modified proof that holds for any C and also accounts for the error due to having only an approximate correction to the logistic distribution. We start with a helpful Lemma before the actual modified Theorem.

Lemma 1. *Let $P(x)$ and $Q(x)$ be two CDFs satisfying $\sup_x |P(x) - Q(x)| \leq \epsilon$ with x in some real range. Let $R(y)$ be the density of another random variable Y. Let P' be the convolution $P * R$ and Q' be the convolution $Q * R$. Then $P'(z)$ (resp. $Q'(z)$) is the CDF of sum $Z = X + Y$ of independent random variables X with CDF $P(x)$ (resp. $Q(x)$) and Y with density $R(y)$. Then*

$$\sup_x |P'(x) - Q'(x)| \leq \epsilon.$$

Proof. See [Seita et al., 2017, Lemma 4] . □

Theorem 2. *If* $\sup_y |\mathbb{P}(\Delta^* < y) - \Phi(\frac{y-\Delta}{s_{\Delta^*}})| \leq \epsilon_1(\theta', \theta, b)$ *and* $\sup_y |S'(y) - S(y)| \leq \epsilon_2$, *then* $\sup_y |\mathbb{P}(\Delta^* + V_{nc} + V_{cor}^{(C)} < y) - S(y - \Delta)| \leq \epsilon_1(\theta', \theta, b) + \epsilon_2$, *where s_{Δ^*} is the sample standard deviation of Δ^*, S' is the cdf of the approximate logistic distribution produced by $\mathcal{N}(0, C) + V_{cor}^{(C)}$, and S is the exact logistic function.*

Proof. As in the original proof [Seita et al., 2017, Cor. 2] the main idea is to use Lemma 1 two times. First, take $P(y) = \mathbb{P}(\Delta^* < y), Q(y) = \Phi(\frac{y-\Delta}{s_{\Delta^*}})$ and convolve with V_{nc} which has density $\phi(\frac{x}{\sqrt{C - s_{\Delta^*}^2}})$. For the second step, take the results $P'(y) = \mathbb{P}(\Delta^* + V_{nc} < y), Q'(y) = \Phi(\frac{y-\Delta}{\sqrt{C}})$ and convolve with the density of $V_{cor}^{(C)}$ to get $P''(y) = \mathbb{P}(\Delta^* + V_{nc} + V_{cor}^{(C)} < y), Q''(y) = S'(y - \Delta)$. By Lemma 1, both convolutions preserve the error bound $\epsilon_1(\theta', \theta, b)$, and we therefore have

$$\sup_y |\mathbb{P}(\Delta^* + V_{nc} + V_{cor}^{(C)} < y) - S(y - \Delta)| \tag{54}$$

$$= \sup_y |\mathbb{P}(\Delta^* + V_{nc} + V_{cor}^{(C)} < y) - S'(y - \Delta) + S'(y - \Delta) - S(y - \Delta)| \tag{55}$$

$$\leq \sup_y |\mathbb{P}(\Delta^* + V_{nc} + V_{cor}^{(C)} < y) - S'(y - \Delta)| + \sup_y |S'(y) - S(y)| \tag{56}$$

$$\leq \epsilon_1(\theta', \theta, b) + \epsilon_2, \tag{57}$$

where (56) follows from the triangle inequality. □

Finally, a bound on the test error implies a bound for the stationary distribution of the Markov chain relative to the true posterior, given in Theorem 3. Writing $d_v(P, Q)$ for the total variation distance between distributions P and Q, $\mathcal{T}_0$ for the transition kernel of the exact Markov chain, $\mathcal{S}_0$ for the exact posterior, and $\mathcal{S}_\epsilon$ for the stationary distribution of the approximate transition kernel where ϵ is the error in the acceptance test, we have:

Theorem 3. *If $\mathcal{T}_0$ satisfies the contraction condition $d_v(P\mathcal{T}_0, \mathcal{S}_0) < \eta d_v(P, \mathcal{S}_0)$ for some constant $\eta \in [0, 1)$ and all probability distributions P, then*

$$d_v(\mathcal{S}_0, \mathcal{S}_\epsilon) \leq \frac{\epsilon}{1-\eta},$$

where ϵ is the bound on the error in the acceptance test.

Proof. See [Korattikara et al., 2014, Theorem 1] . □

Generally, especially the contraction condition in Theorem 3 can be hard to meet: it can be shown to hold e.g. for some Gibbs samplers (see e.g. Brémaud 1999, Theorem 6.1) but it is not usually valid for an arbitrary model, and even checking the condition might not be trivial.

5 Numerical approximation of the correction distribution

As noted in the main text, we need to find an approximate distribution $V_{cor}^{(C)}$ s.t.

$$V_{log} \stackrel{d}{=} \mathcal{N}(0, C) + V_{cor}^{(C)}, \tag{58}$$

where V_{log} has a standard logistic distribution. The approximation method of Seita et al. [2017] casts the problem into a ridge regression problem, which can be solved effectively. However, nothing constrains the resulting function from having negative values. In order to use it as an approximate pdf, Seita et al. [2017] set these to zeroes and note that as long as C is small enough, such values are rare and hence do not affect the solution much. In practice, their solution seems to work very well with small values of C, e.g. when $C \leq 1$.

Since we want to use larger C for the privacy, we propose to approximate $V_{cor}^{(C)}$ with a Gaussian mixture model (GMM). Since the result is always a valid pdf, the problem of negative values does not arise.

To find the correction pdf, denote the density of the GMM approximation with K components by $\tilde{f}_{cor}$, the GMM component parameters by π_k, μ_k and σ_k, and the standard normal density by ϕ. We have

$$\begin{aligned} f_{log}(x) &= (f_{norm} * f_{cor})(x) \simeq (f_{norm} * \tilde{f}_{cor})(x) \\ &= \int_{\mathbb{R}} f_{norm}(x) \tilde{f}_{cor}(x-t) dt \\ &= \int_{\mathbb{R}} \phi(\frac{x}{\sqrt{C}}) [\sum_{k=1}^{K} \pi_k \phi(\frac{x-t-\mu_k}{\sigma_k})] dt \\ &= \sum_{k=1}^{K} \pi_k \phi(\frac{x-\mu_k}{\sqrt{C+\sigma_k^2}}) = \tilde{f}_{log}^{(C)}(x; \pi_k, \mu_k, \sigma_k, k = 1, \ldots, K) \end{aligned}$$

As the logistic pdf is symmetric around zero, we require our GMM approximation to be symmetric as well. We achieve this by creating a counterpart for each mixture component with an opposite sign mean and identical variance and weight. To construct the approximation on some interval $[-a, a] \subset \mathbb{R}$, we discretise the interval into n points, and fit the GMM by minimising the loss function

$$\mathcal{L}(\pi, \mu, \sigma) = \|f_{log} - \tilde{f}_{log}^{(C)}\|_2 \tag{59}$$

calculated over the discretisation. Since GMM is a generative model, sampling from the optimised approximation is easy.

Figure 1 shows the approximation error $\max_y |S'(y) - S(y)|$, where S' is the approximate logistic ecdf and S the exact logistic cdf, due to $\tilde{V}_{cor}$ using the ridge regression solution proposed by Seita et al. [2017] and the GMM. The error measure is the same as in Theorem 2 in the Supplement. Empirically, as shown in the Figure, we can have noticeably better approximation especially with larger C values.

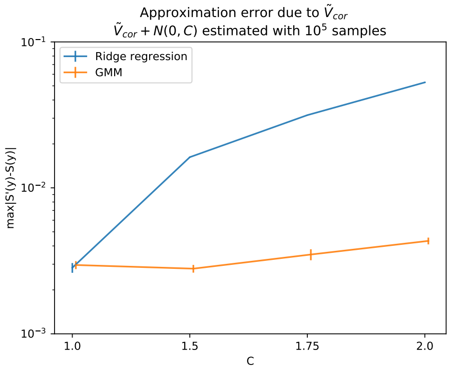
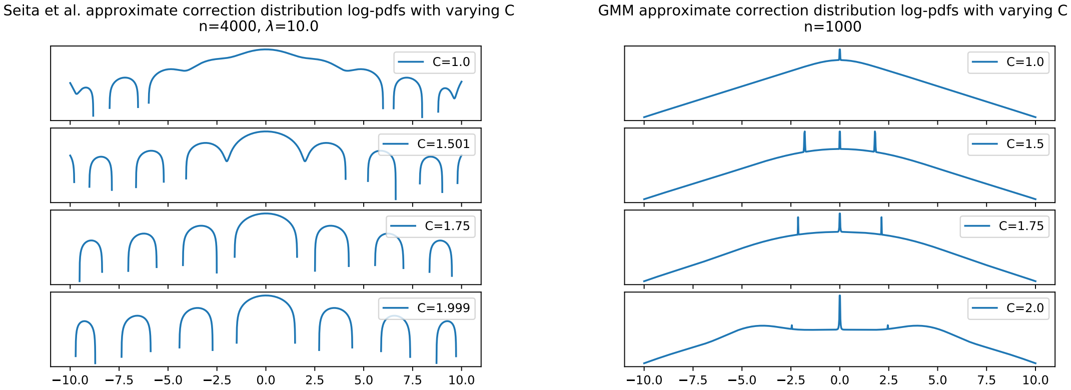


Figure 1: *Approximation error due to $\tilde{V}_{cor}$ with error bars showing the standard error of the mean calculated from 20 runs. With the ridge regression solution proposed by Seita et al. [2017] the error increases quickly when $C > 1$. Using the GMM approximation we can achieve significantly smaller error with $C = 2$.*

Figure 2 shows the two approximations with increasing C. When the negative values in the ridge regression solution are projected to zeroes, the variance of V_{cor} increases and the resulting approximate $\tilde{V}_{log}$ has variance much larger than the actual $\pi^2/3$ it should have. This also shows in the resulting approximation. Figure 3 shows the empirical cdf for both approximations and for the true logistic distribution, and the absolute distance between the approximations S' and the true logistic cdf S.

To calculate the ridge regression solution for $[-10, 10]$, we use the original code of Seita et al. [2017] with parameter values $n = 4000$, $\lambda = 10.0$ used in the original paper. The problems with larger C values persisted with other parameter settings we tested. Note that the discretisation granularity parameter n used in the two methods are not directly comparable.

To fit the GMMs with K components, we take the interval $[-10, 10]$ with $n = 1000$ points for calculating the loss function, and run 20000 optimisation iterations with PyTorch [Paszke et al., 2017]. We use Adam optimiser [Kingma and Ba, 2014] with learning rate $\eta = 0.01$ and otherwise default settings. The approximation is forced to be symmetric about zero by adding mirrored components: for the kth component we add a copy but set the mean as $-\mu_k$, and set



(a) Ridge regression results

(b) GMM results

Figure 2: *Approximate correction distribution log-densities with varying C values. Figure 2(a) shows the results for the ridge regression solution used by Seita et al.: as C increases, the amount of negative values that are projected to zeroes, which show as gaps in the log-pdf, increases markedly. Figure 2(b) shows corresponding results for our GMM solution: the approximation is always a valid pdf over $\mathbb{R}$.*

the weights as $\pi_k/2$ for both, i.e., use the mean of the original and the mirrored component. We use $K = 50$ in the test, which gives 100 components with mirroring.

References

Pierre Brémaud. *Markov Chains: Gibbs fields, Monte Carlo simulation, and Queues*. Springer, 1. edition, 1999.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

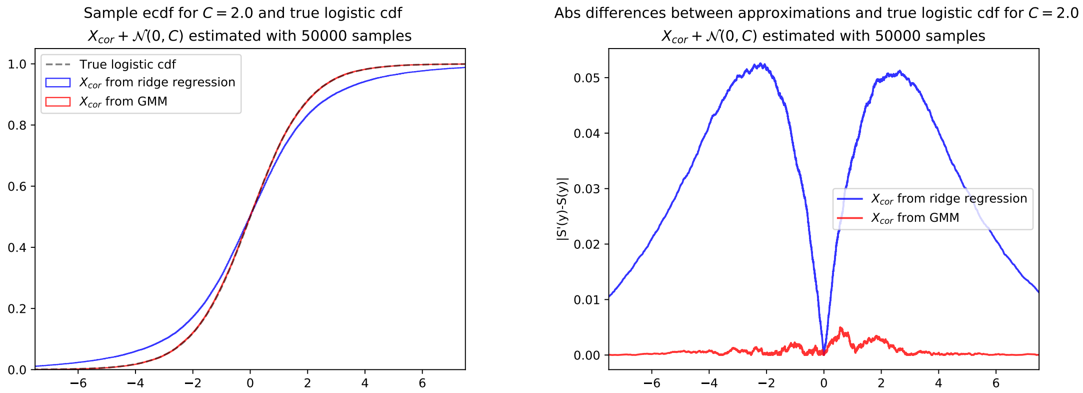
Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 181–189, Bejing, China, 22–24 Jun 2014. PMLR.

Ilya Mironov. Rényi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in Pytorch. In *Workshop in NIPS*, 2017.

Daniel Seita, Xinlei Pan, Haoyu Chen, and John F. Canny. An efficient minibatch acceptance test for Metropolis–Hastings. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In Kamalika Chaudhuri and Masashi



(a) Approximation ecdf and true logistic cdf

(b) Absolute differences from true logistic cdf

Figure 3: *Figure 3(a) shows the empirical cdf for the approximate logistic distributions calculated using the ridge regression solution of Seita et al. and our GMM together with true logistic cdf. The variance of V_{cor} using ridge regression is too high and the resulting $V_{cor} + \mathcal{N}(0, C)$ is clearly off. The ecdf for GMM is almost indistinguishable from the true cdf. Figure 3(b) shows the absolute distances between the approximation ecdf and the true logistic cdf.*

Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1226–1235. PMLR, 16–18 Apr 2019.

Paper III

III

Antti Koskela, Mikko A. Heikkilä and Antti Honkela

Numerical Accounting in the Shuffle Model of Differential Privacy

Transactions on Machine Learning Research, 2023.

Numerical Accounting in the Shuffle Model of Differential Privacy

Antti Koskela *antti.h.koskela@nokia-bell-labs.com*
Nokia Bell Labs
University of Helsinki

Mikko Heikkilä *mikko.a.heikkila@helsinki.fi*
Department of Computer Science
University of Helsinki

Antti Honkela *antti.honkela@helsinki.fi*
Department of Computer Science
University of Helsinki

Reviewed on OpenReview: *https://openreview.net/forum?id=11osftjEbF*

Abstract

Shuffle model of differential privacy is a novel distributed privacy model based on a combination of local privacy mechanisms and a secure shuffler. It has been shown that the additional randomisation provided by the shuffler improves privacy bounds compared to the purely local mechanisms. Accounting tight bounds, however, is complicated by the complexity brought by the shuffler. The recently proposed numerical techniques for evaluating (ε, δ)-differential privacy guarantees have been shown to give tighter bounds than commonly used methods for compositions of various complex mechanisms. In this paper, we show how to utilise these numerical accountants for adaptive compositions of general ε-LDP shufflers and for shufflers of k-randomised response mechanisms, including their subsampled variants. This is enabled by an approximation that speeds up the evaluation of the corresponding privacy loss distribution from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, where n is the number of users, without noticeable change in the resulting $\delta(\varepsilon)$-upper bounds. We also demonstrate looseness of the existing bounds and methods found in the literature, improving previous composition results for shufflers significantly.

1 Introduction

The shuffle model of differential privacy (DP) is a distributed privacy model which sits between the high trust–high utility centralised DP, and the low trust–low utility local DP (LDP). In the shuffle model, the individual results from local randomisers are only released through a secure shuffler. This additional randomisation leads to "amplification by shuffling", resulting in better privacy bounds against adversaries without access to the unshuffled local results.

We consider computing privacy bounds for both single and composite shuffle protocols, where by composite protocol we mean a protocol, where the subsequent user-wise local randomisers depend on the same local datasets and possibly on the previous output of the shuffler, and at each round the results from the local randomisers are independently shuffled. Moreover, using the analysis by Feldman et al. (2023), we provide bounds in the case the subsequent local randomisers are allowed to depend adaptively on the output of the previous ones.

In this paper we show how numerical accounting (Koskela et al., 2020; 2021; Gopi et al., 2021) can be employed for privacy analysis of both single and composite shuffle DP mechanisms. We demonstrate that thus obtained bounds can be up to orders of magnitudes tighter than the existing bounds from the literature. We also evaluate how significantly adversaries with varying capabilities differ in terms of the resulting privacy bounds using the k-randomised response mechanism. For conciseness, most of the proofs are given in the Appendix.

1.1 Related work

DP was originally defined in the central model assuming a trusted aggregator by Dwork et al. (2006), while the fully distributed LDP was formally introduced and analysed by Kasiviswanathan et al. (2011). Closely related to the shuffle model of DP, Bittau et al. (2017) proposed the Encode, Shuffle, Analyze framework for distributed learning, which uses the idea of secure shuffler for enhancing privacy. The shuffle model of DP was formally defined by Cheu et al. (2019), who also provided the first separation result showing that the shuffle model is strictly between the central and the local models of DP. Another direction initiated by Cheu et al. (2019) and continued, e.g., by Balle et al. (2020b); Ghazi et al. (2021) has established a separation between single- and multi-message shuffle protocols.

There exists several papers on privacy amplification by shuffling, some of which are central to this paper. Erlingsson et al. (2019) showed that the introduction of a secure shuffler amplifies the privacy guarantees against an adversary, who is not able to access the outputs from the local randomisers but only sees the shuffled output. Balle et al. (2019) improved the amplification results and introduced the idea of privacy blanket, which we also utilise in our analysis of k-randomised response. Feldman et al. (2021) used a related idea of hiding in the crowd to improve on the previous results, and their analysis was further improved in (Feldman et al., 2023). Girgis et al. (2021) generalised shuffling amplification further to scenarios with composite protocols and parties with more than one local sample under simultaneous communication and privacy restrictions. We use the improved results of Feldman et al. (2023) in the analysis of general LDP mechanisms, and compare our bounds with theirs in Section 3.3. We also calculate privacy bounds in the setting considered by Girgis et al. (2021), namely in the case where a subset of users sending contributions to the shufflers are sampled randomly. This can be seen as a subsampled mechanism and we are able to combine the analysis of Feldman et al. (2023), the privacy loss distribution related subsampling results of Zhu et al. (2022) and FFT accounting to obtain tighter (ε, δ)-bounds than Girgis et al. (2021), as shown in Section 3.4.

2 Background: numerical privacy accounting

Before analysing the shuffled mechanisms we introduce some required theory and notations. In particular, we use the privacy loss distribution formalism, which is based on finding the so-called dominating pairs of distributions for the given mechanisms. For more detailed presentations of the theory, we refer to Koskela et al. (2021); Gopi et al. (2021); Zhu et al. (2022).

2.1 Differential privacy and privacy loss distribution

An input dataset containing n data points is denoted as $X = (x_1, \ldots, x_n) \in \mathcal{X}^n$, where $x_i \in \mathcal{X}$, $1 \leq i \leq n$. We say $X, X' \in \mathcal{X}^n$ are neighbours if we get one by substituting one element in the other (denoted $X \sim X'$).

Definition 1. *Let $\varepsilon > 0$ and $\delta \in [0,1]$. Let P and Q be two random variables taking values in the same measurable space $\mathcal{O}$. We say that P and Q are (ε, δ)-indistinguishable, denoted $P \simeq_{(\varepsilon,\delta)} Q$, if for every measurable set $E \subset \mathcal{O}$ we have*

$$\Pr(P \in E) \leq \mathrm{e}^{\varepsilon} \Pr(Q \in E) + \delta, \qquad \Pr(Q \in E) \leq \mathrm{e}^{\varepsilon} \Pr(P \in E) + \delta.$$

Definition 2. *Let $\varepsilon > 0$ and $\delta \in [0,1]$. Mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{O}$ is (ε, δ)-DP if for every $X \sim X'$: $\mathcal{M}(X) \simeq_{(\varepsilon,\delta)} \mathcal{M}(X')$. We call $\mathcal{M}$ tightly (ε, δ)-DP, if there does not exist $\delta' < \delta$ such that $\mathcal{M}$ is (ε, δ')-DP.*

When the data are distributed among several parties, and the local datasets are only accessed via purely local DP mechanisms, we say that the mechanisms guarantee local DP (LDP) and call the local DP mechanisms local randomisers (Kasiviswanathan et al., 2011).

We rely on the results of Zhu et al. (2022) and characterise (ε, δ)-DP bounds using the hockey-stick divergence, which for $\alpha \geq 0$ is defined as

$$H_\alpha(P||Q) = \int [P(t) - \alpha \cdot Q(t)]_+ \, \mathrm{d}t,$$

where for $x \in \mathbb{R}$, $x_+ = \max\{0, x\}$. Using the hockey-stick divergence, by (Lemma 5, Zhu et al., 2022), tight (ε, δ)-DP bounds can also be characterised as

$$\delta(\varepsilon) = \max_{X \sim X'} H_{\mathrm{e}^\varepsilon}(\mathcal{M}(X)||\mathcal{M}(X')).$$

We can generally find (ε, δ)-bounds by analysing dominating pairs of distributions:

Definition 3 (Zhu et al. 2022)**.** *A pair of distributions (P, Q) is a dominating pair of distributions for mechanism $\mathcal{M}$ if for all $\alpha \geq 0$,*

$$\max_{X \sim X'} H_\alpha(\mathcal{M}(X)||\mathcal{M}(X')) \leq H_\alpha(P||Q).$$

Using dominating pairs of distributions, we can obtain $\delta(\varepsilon)$-upper bounds for adaptive compositions:

Theorem 4 (Zhu et al. 2022)**.** *If (P, Q) dominates $\mathcal{M}$ and (P', Q') dominates $\mathcal{M}'$, then $(P \times P', Q \times Q')$ dominates the adaptive composition $\mathcal{M} \circ \mathcal{M}'$.*

Having dominating pairs of distributions for each individual mechanism in a composition, the hockey-stick divergence can be transformed into a more easily computable form by using the privacy loss random variables (PRVs). PRV for a pair of distributions (P, Q) is defined as follows.

Definition 5. *Let $P(t)$ and $Q(t)$ be probability density functions. We define the PRV $\omega_{P/Q}$ as*

$$\omega_{P/Q} = \log \frac{P(t)}{Q(t)}, \quad t \sim P(t),$$

where $t \sim P(t)$ means that t is distributed according to $P(t)$.

With a slight abuse of notation, we denote the probability density function of the random variable $\omega_{P/Q}$ by $\omega_{P/Q}(t)$, and call it the privacy loss distribution (PLD).

The $\delta(\varepsilon)$-bounds can be stated using the following representation that involves the PRV.

Theorem 6 (Gopi et al. 2021)**.** *We have:*

$$H_{\mathrm{e}^\varepsilon}(P||Q) = \mathbb{E}_{\omega_{P/Q}} \left[1 - \mathrm{e}^{\varepsilon - \omega_{P/Q}}\right]_+, \tag{2.1}$$

Moreover, if $\omega_{P/Q}$ is a PRV for the pair of distributions (P, Q) and $\omega_{P'/Q'}$ a PRV for the pair of distributions (P', Q'), then the PRV for the pair of distributions $(P \times P', Q \times Q')$ is given by $\omega_{P/Q} + \omega_{P'/Q'}$

By identifying dominating pairs of distributions for each mechanism in a composition and by formulating the $\delta(\varepsilon)$-bound via hockey-stick divergence as an integral of the form given in Equation 2.1, the numerical PLD accountants (Koskela et al., 2021; Gopi et al., 2021) can be utilised for computing accurate $\delta(\varepsilon)$-bounds.

We will also use the following subsampling amplification result (Proposition 30, Zhu et al., 2022), which leads to a privacy profile for the composed mechanism $\mathcal{M} \circ S_{Subset}$, where S_{Subset} denotes a subsampling procedure where, from an input of n entries, a fixed sized subset of $q \cdot n$, $0 < q \leq 1$, entries is sampled without replacement.

Lemma 7 (Zhu et al. 2022)**.** *Denote the subsampled mechanism $\widetilde{\mathcal{M}} := \mathcal{M} \circ S_{Subset}$. Suppose a pair of distributions (P, Q) is a dominating pair of distributions for a mechanism $\mathcal{M}$ for all datasets of size $q \cdot n$ under the $\sim$-neighbouring relation (i.e., the substitute relation), where $q > 0$ is the subsampling ratio (size of the subset divided by n). Then, for all neighbouring datasets (under the $\sim$-neighbouring relation) X and Y of size n,*

$$\begin{aligned} H_\alpha\big(\widetilde{\mathcal{M}}(X)||\widetilde{\mathcal{M}}(Y)\big) &\leq H_\alpha\big(q \cdot P + (1-q) \cdot Q||Q\big), \quad \textit{for } \alpha \geq 1, \\ H_\alpha\big(\widetilde{\mathcal{M}}(X)||\widetilde{\mathcal{M}}(Y)\big) &\leq H_\alpha\big(P||q \cdot Q + (1-q) \cdot P\big), \quad \textit{for } 0 \leq \alpha < 1. \end{aligned} \tag{2.2}$$

Considering the assumptions of Lemma 7, if we define a function $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$

$$h(\alpha) = \max\{H_\alpha\big(q \cdot P + (1-q) \cdot Q || Q\big), H_\alpha\big(P || q \cdot Q + (1-q) \cdot P\big)\}, \tag{2.3}$$

we see that $h(\alpha)$ clearly defines a privacy profile: it is convex and has all the other required properties of a privacy profile. Thus we can use an existing numerical method (Doroshenko et al., 2022, Algorithm 1) with the function h to obtain discrete-valued distributions $\widetilde{P}$ and $\widetilde{Q}$, that are a dominating pair for $\widetilde{\mathcal{M}} = \mathcal{M} \circ S_{Subset}$.

We remark that by (Theorem 10, Zhu et al., 2022), the two pairs of distributions on the right-hand side of Equation 2.2 give dominating pairs for remove and add neighbouring relations of datasets in case the pair (P, Q) is a dominating pair of distributions for $\mathcal{M}$ under remove and add neighbouring relations, respectively, and can therefore be used to compute (ε, δ)-upper bounds in case of add/remove neighbouring relations of datasets. Then, the computation is more straightforward since one can simply take the maximum of the $\delta(\varepsilon)$-values obtained under the remove and add neighbouring relations and therefore using the techniques of Doroshenko et al. (2022) is not necessary. We focus on using the $\sim$-relation as the dominating pair (P, Q) obtained using both the post-processing results of (Feldman et al., 2023) and using our analysis for the k-RR local randomiser is a dominating pair under the $\sim$-relation. The $\sim$-relation is also behind the baseline bounds by Girgis et al. (2021). We illustrate in Fig. 1 the accuracy of the numerical construction of (Doroshenko et al., 2022, Algorithm 1) applied to the privacy profile given in Equation 2.3.

2.2 Numerical PLD accounting using FFT

In order to evaluate integrals of the form given in Equation 2.1, we use the Fast Fourier Transform (FFT)-based method by Koskela et al. (2021) called the Fourier Accountant (FA). This means that each PLD is truncated and placed on an equidistant numerical grid over an interval $[-L, L]$, $L > 0$. The distributions for the sums of the PRVs are given by convolutions of the individual PLDs and are evaluated using the FFT algorithm. By a careful error analysis the error incurred by the numerical method can be bounded and an upper $\delta(\varepsilon)$-bound obtained. We note that alternatively, for accurately computing the integrals we could also use the FFT-based method proposed by Gopi et al. (2021).

3 General shuffled ε_0-LDP mechanisms

Feldman et al. (2023) consider general ε_0-LDP local randomisers combined with a shuffler. The analysis allows also sequential adaptive compositions of the user contributions before shuffling. The analysis is based on decomposing individual LDP contributions to mixtures of data dependent part and noise, which leads to finding (ε, δ)-bounds for the pair of 2-dimensional random variables (see Thm. 3.1 of Feldman et al., 2023)

$$P = (A + \Delta_1, C - A + \Delta_2), \qquad Q = (A + \Delta_2, C - A + \Delta_1), \tag{3.1}$$

where for $n \in \mathbb{N}$,

$$C \sim \text{Bin}(n-1, 2p), \quad A \sim \text{Bin}\left(C, \tfrac{1}{2}\right), \quad \Delta_1 \sim \text{Bern}\left(\mathrm{e}^{\varepsilon_0} p\right) \quad \text{and} \quad \Delta_2 \sim \text{Bin}\left(1 - \Delta_1, \tfrac{p}{1 - \mathrm{e}^{\varepsilon_0} p}\right), \tag{3.2}$$

and $p = \frac{1}{\mathrm{e}^{\varepsilon_0}+1}$. Intuitively, C denotes the number of other users whose mechanism outputs are indistinguishable "clones" of the two differing users, with A denoting random split between these. Using the following lemma, we can use the FFT-based numerical accountants to obtain accurate bounds also for adaptive compositions of general ε_0-LDP shuffling mechanisms:

Lemma 8. *Let X and X' be neighbouring datasets and denote by $\mathcal{A}_s(X)$ and $\mathcal{A}_s(X')$ outputs of the shufflers of adaptive ε_0-LDP local randomisers (for a detailed description of $\mathcal{A}_s$, see Thm. 3.1 by Feldman et al., 2023, which uses the same notation). Then, for all $\alpha \geq 0$,*

$$H_\alpha(\mathcal{A}_s(X) || \mathcal{A}_s(X')) \leq H_\alpha(P || Q),$$

where P and Q are given in Equation 3.1.

Proof. By Thm. 3.1 of Feldman et al. (2023) there exists a post-processing algorithm Φ such that $\mathcal{A}_s(X)$ is distributed identically to $\Phi(P)$ and $\mathcal{A}_s(X')$ identically to $\Phi(Q)$. The claim follows then from the data-processing inequality which holds for the hockey-stick divergence (Balle et al., 2020a). □

Corollary 9. *The pair of distributions (P, Q) given in Equation 3.1 is a dominating pair of distributions for the shuffling mechanism $\mathcal{A}_s(X)$.*

Furthermore, using Thm. 4, we can bound the $\delta(\varepsilon)$ of n_c-wise adaptive composition of the shuffler $\mathcal{A}_s$ using product distributions of Ps and Qs:

Corollary 10. *Denote $\mathcal{A}_s^{n_c}(X, z_0) = \mathcal{A}_s(X, \mathcal{A}_s(X, ...\mathcal{A}_s(X, z_0)))$ for some initial state z_0. For all neighbouring datasets X and X' and for all $\alpha \geq 0$,*

$$H_\alpha(\mathcal{A}_s^{n_c}(X)||\mathcal{A}_s^{n_c}(X')) \leq H_\alpha(P \times \ldots \times P||Q \times \ldots \times Q), \tag{3.3}$$

We remark that the case of heterogeneous adaptive compositions (e.g. varying n and ε_0) can be handled analogously using Thm. 4.

Thus, using the bound of Equation 3.3 for $\alpha = \mathrm{e}^\varepsilon$, we get upper bounds for adaptive compositions of general shuffled ε_0-LDP mechanisms with the Fourier accountant by finding the PLD for the distributions P, Q (given in Equation 3.1). Note that even though the resulting (ε, δ)-bound is tight for P's and Q's, it need not be tight for a specific mechanism like the shuffled k-RR. The bound simply gives an upper bound for any shuffled ε_0-LDP mechanisms.

3.1 PLD for shuffled ε_0-LDP mechanisms

To analyse compositions of general shuffled ε_0-LDP mechanisms, we need to form the PLD $\omega_{P/Q}$ determined by P and Q of Equation 3.1. Denoting $q = \mathrm{e}^{\varepsilon_0} p$ and $\widetilde{q} = \frac{p}{1-\mathrm{e}^{\varepsilon_0} p}$, $p = \frac{1}{\mathrm{e}^{\varepsilon_0}+1}$, and writing out the randomness of Δ_1 and Δ_2 as mixtures, we see that the random variables P and Q given in Equation 3.1 can be expressed as

$$P = q \cdot P_0 + (1-q)\widetilde{q} \cdot P_1 + (1-q)(1-\widetilde{q}) \cdot P_2, \quad Q = (1-q)\widetilde{q} \cdot P_0 + q \cdot P_1 + (1-q)(1-\widetilde{q}) \cdot P_2,$$

where

$$P_0 \sim (A+1, C-A), \quad P_1 \sim (A, C-A+1), \quad P_2 = (A, C-A)$$

and A and C are as given in Equation 3.2. In the Appendix, we give the required expressions to determine the discrete-valued PLD

$$\omega_{P/Q}(s) = \sum\nolimits_{a,b} \mathbb{P}(P = (a,b)) \cdot \delta_{s_{(a,b)}}(s), \quad s_{(a,b)} = \log\left(\tfrac{\mathbb{P}(P=(a,b))}{\mathbb{P}(Q=(a,b))}\right), \tag{3.4}$$

where $\delta_s(\cdot)$, $s \in \mathbb{R}$, denotes the Dirac delta function centred at s, and similarly also for $\omega_{Q/P}(s)$.

3.2 Lowering PLD computational complexity using Hoeffding's inequality

The PLD of Equation 3.4 has $\mathcal{O}(n^2)$ terms, which makes its naive evaluation overly expensive for a large number of users n. Using an appropriate tail bound (Hoeffding) for the binomial distribution, we can truncate part of the probability mass and add it directly to δ. More specifically, if each PLD ω_i, $1 \leq i \leq n_c$, in an n_c-wise composition is approximated by a truncated distribution $\widetilde{\omega}_i$ such that the truncated probability masses are $\tau_i \geq 0$, respectively, then $\delta(\varepsilon) = \widetilde{\delta}(\varepsilon) + \delta(\infty)$, where $\widetilde{\delta}(\varepsilon)$ is the value of the integral in Equation 2.1 obtained with the truncated PLDs and $\delta(\infty) = 1 - \prod_i (1-\tau_i) \leq \sum_i \tau_i$, gives an upper bound for the composition without truncations. Using the Hoeffding's inequality we obtain an accurate approximation of $\omega_{P/Q}$ with only $\mathcal{O}(n)$ terms. We formalise this approximation as follows:

Lemma 11. *Let the PLD $\omega_{P/Q}$ be defined as in Equation 3.4 (Equation 3.1 gives P and Q which include $C \sim \mathrm{Bin}(n-1, 2p)$ and $A \sim \mathrm{Bin}\left(C, \frac{1}{2}\right)$) and let $\tau > 0$. Consider the set*

$$S_n = [\max\left(0, (2p - c_n)(n-1)\right), \min\left(n-1, (2p + c_n)(n-1)\right)],$$

where $c_n = \sqrt{\frac{\log(4/\tau)}{2(n-1)}}$ *and the set*

$$S_i = [\max\left(0, (\tfrac{1}{2} - c_i) \cdot i\right), \min\left(n-1, (\tfrac{1}{2} + c_i) \cdot i\right)],$$

where $c_i = \sqrt{\frac{\log(4/\tau)}{2 \cdot i}}$. *Then,* $\widetilde{\omega}_{P/Q}$ *defined by*

$$\widetilde{\omega}_{P/Q}(s) = \sum_{i \in S_n} \sum_{j \in S_i} \mathbb{P}(P = (j+1, i-j)) \cdot \delta_{s_{j+1,i-j}}(s), \quad s_{a,b} = \log\left(\tfrac{\mathbb{P}(P=(a,b))}{\mathbb{P}(Q=(a,b))}\right) \tag{3.5}$$

has $\mathcal{O}(n \cdot \log(4/\tau))$ *terms and differs from* $\omega_{P/Q}$ *at most by mass* τ.

Proof. As A is conditioned on C, we first use a tail bound on C and then on A to reduce the number of terms. Using Hoeffding's inequality for $C \sim \text{Bin}(n-1, 2p)$ states that for $c > 0$,

$$\mathbb{P}\big(C \leq (2p - c)(n-1)\big) \leq \exp\big(-2(n-1)c^2\big),$$
$$\mathbb{P}\big(C \geq (2p + c)(n-1)\big) \leq \exp\big(-2(n-1)c^2\big).$$

Requiring that $2 \cdot \exp\left(-2(n-1)c^2\right) \leq \tau/2$ gives the condition $c \geq \sqrt{\frac{\log(4/\tau)}{2(n-1)}}$ and the expressions for c_n and S_n. Similarly, we use Hoeffding's inequality for $A \sim \text{Bin}(C, \frac{1}{2})$ and get expressions for c_i and S_i. The total neglected mass is at most $\tau/2 + \tau/2 = \tau$. For the number of terms, we see that S_n contains at most $2c_n(n-1) = \sqrt{n-1}\sqrt{2 \cdot \log(4/\tau)}$ terms and for each i, and S_i contains at most $2c_i i = \sqrt{i}\sqrt{2 \cdot \log(4/\tau)} \leq \sqrt{n-1}\sqrt{2 \cdot \log(4/\tau)}$ terms. Thus $\widetilde{\omega}_{P/Q}$ has at most $\mathcal{O}(n \cdot \log(4/\tau))$ terms. We get the form of Equation 3.5 by an appropriate change of variables. □

When evaluating $\widetilde{\omega}_{P/Q}$, we require that the neglected mass is smaller than some prescribed tolerance τ (e.g. $\tau = 10^{-12}$). When computing guarantees for compositions, the cost of FFT for evaluating the convolutions dominates the rest of the computation.

3.3 Experimental comparison to RDP

Figure 1 shows a comparison between the PLD and RDP applied to the pair of distributions P and Q given in Equation 3.1. RDP bounds for composition are computed using standard composition results (Mironov, 2012) and the RDP bounds are converted to DP bounds using the conversion formula given by Canonne et al. (2020). Naive evaluation of RDP-values is $\mathcal{O}(n^2)$ computation. We heuristically speed up RDP evaluation using the Hoeffding inequality (Lemma 11) and check that increasing the accuracy does not change the results.

3.4 Experimental comparison to the subsampled RDP bounds of Girgis et al. (2021)

Girgis et al. (2021) consider a protocol where a randomly sampled, fixed sized subset of users sends contributions to the shuffler on each round, and the local randomisers are assumed to be integer-valued ε_0-LDP mechanisms. This can be seen as a composition of a shuffler and a subsampling mechanism. We can generalise our analysis to this case via Lemma 7, and use Algorithm 1 of Doroshenko et al. (2022) on the function $h(\alpha)$ defined in Equation 2.3 to obtain the dominating pair of distributions for the subsampled shuffler. To this end, we need to define a grid for α: $\{\alpha_0, \ldots, \alpha_{n_\alpha+1}\}$, where $0 = \alpha_0 < \alpha_1 < \ldots < \alpha_{n_\alpha} < \alpha_{n_\alpha+1} = \infty$. We consider a logarithmically equidistant grid between α_1 and α_{n_α}. Thus, in practice this means that we need to determine α_1 and α_{n_α} and the value n_α. Figure 2 illustrates the convergence of the obtained approximation as we refine the α-grid, for a subsampled shuffler, where the dominating pair of distributions P and Q for the non-subsampled shuffler are obtained from (Thm. 3.1 Feldman et al., 2023).

As we see from Figure 3, this approach leads to considerably lower $\varepsilon(\delta)$-bounds than the approach by Girgis et al. (2021). Notice that the tightness of the PLD-based bound is mostly determined by the analysis of Feldman et al. (2023) which gives the dominating pair (P, Q) of Equation 3.1 and that the RDP-based analysis of Girgis et al. (2021) is fundamentally different.

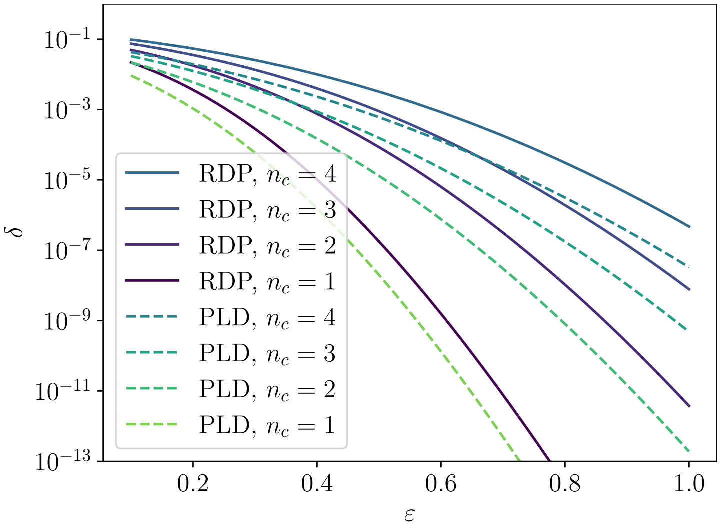


Figure 1: Evaluation of $\delta(\varepsilon)$ for general single and composite shuffle ε_0-LDP mechanisms using RDP accounting and FFT-based numerical accounting (PLD) applied to the pair of distributions P and Q given by the post-processing result of Feldman et al. (2023). Number of users $n = 10^4$ and the LDP parameter $\varepsilon_0 = 4.0$.

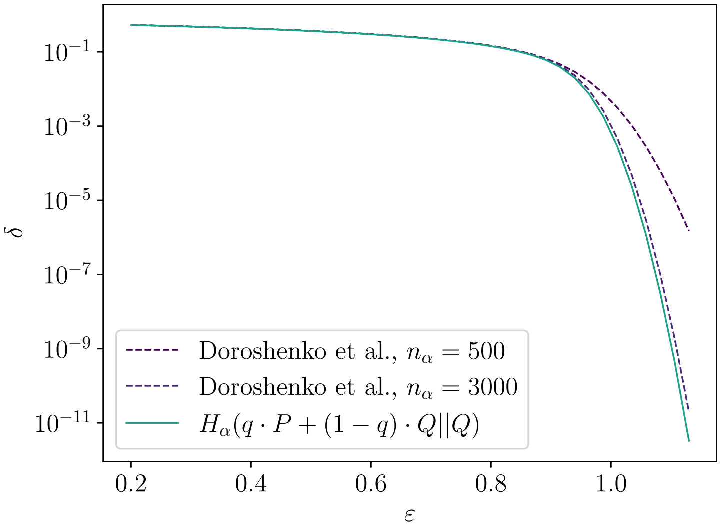


Figure 2: We apply FFT-based method on the dominating pair of distributions given by Algorithm 1 of Doroshenko et al. (2022) applied on the function $h(\alpha)$ that we obtain from Lemma 7, for different sizes of α-grids. Here, the underlying P and Q are obtained from the analysis of Feldman et al. (2023), and we set $\varepsilon_0 = 3.0$, $n = 10^4$, $n_c = 2000$, subsampling ratio $q = 0.01$, $\alpha_1 = \exp(-0.25)$, $\alpha_{n_\alpha} = \exp(0.25)$, and take a logarithmically equidistant α-grid. We also plot $H_{\mathrm{e}^\varepsilon}(q \cdot P + (1-q) \cdot Q||Q)$ for comparison.

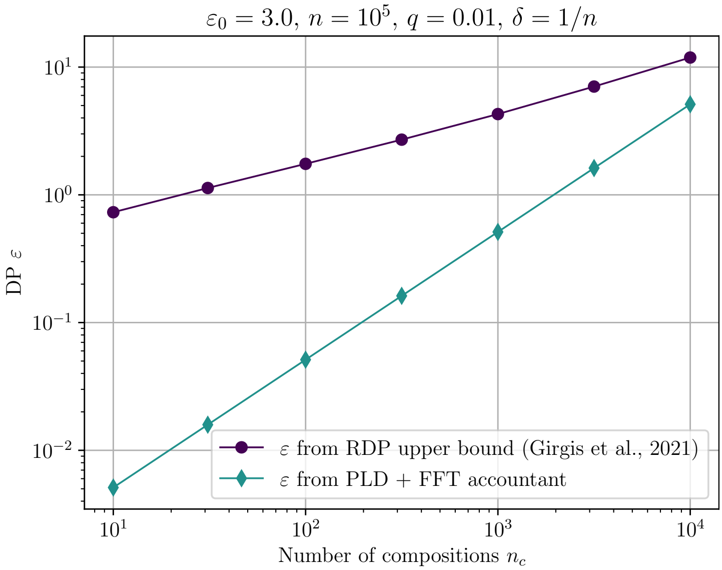


Figure 3: Evaluation of $\varepsilon(\delta)$ for compositions of subsampled shufflers of ε_0-local randomisers. We compare the bounds obtained using the FFT-accounting and the PLD determined by the numerical method of (Doroshenko et al., 2022, Algorithm 1) applied to the dominating pair of Equation 3.1 and the RDP-bounds given in Thm. 2 of (Girgis et al., 2021) that are mapped to $\varepsilon(\delta)$-bounds using Lemma 1 of (Girgis et al., 2021). Here the number of compositions n_c varies and n is fixed. Here q denotes the subsampling ratio.

4 Shuffled k-randomised response

Balle et al. (2019) give a protocol for n parties to compute a private histogram over the domain $[k]$ in the single-message shuffle model. The randomiser is parameterised by a probability γ, and consists of a k-ary randomised response mechanism (k-RR) that returns the true value with probability $1-\gamma$ and a uniformly random value with probability γ. Denote this k-RR randomiser by $\mathcal{R}^{PH}_{\gamma,k,n}$ and the shuffling operation by $\mathcal{S}$. Thus, we are studying the privacy of the shuffled randomiser $\mathcal{M} = \mathcal{S} \circ \mathcal{R}^{PH}_{\gamma,k,n}$.

Consider first the proof of Balle et al. (2019, Thm. 3.1). Assuming without loss of generality that the differing data element between X and X', $X, X' \in [k]^n$, is x_n, the (strong) adversary A_s used by Balle et al. (2019, Thm. 3.1) is defined as follows.

Definition 12. *Let $\mathcal{M} = \mathcal{S} \circ \mathcal{R}^{PH}_{\gamma,k,n}$ be the shuffled k-RR mechanism, and w.l.o.g. let the differing element be x_n. We define adversary A_s as an adversary with the view*

$$View^{A_s}_{\mathcal{M}}(X) = \big((x_1, \dots, x_{n-1}), \quad \beta \in \{0,1\}^n, \quad (y_{\pi(1)}, \dots, y_{\pi(n)})\big),$$

where y are the outputs from the shuffler, β is a binary vector identifying which parties answered randomly, and π is a uniformly random permutation applied by the shuffler.

Assuming w.l.o.g. that the differing element $x_n = 1$ and $x'_n = 2$, the proof then shows that for any possible view V of the adversary A_s,

$$\frac{\mathbb{P}(\mathrm{View}^{A_s}_{\mathcal{M}}(X) = V)}{\mathbb{P}(\mathrm{View}^{A_s}_{\mathcal{M}}(X') = V)} = \frac{N_1 + 1}{N_2}, \tag{4.1}$$

where N_i denotes the number of messages received by the server with value i after removing from the output Y any truthful answers submitted by the first $n-1$ users. The (ε, δ)-analysis of Balle et al. (2019) is based on showing that for all neighbouring X and X',

$$\mathrm{View}_{\mathcal{M}}^{A_s}(X) \simeq_{(\varepsilon,\delta)} \mathrm{View}_{\mathcal{M}}^{A_s}(X') \tag{4.2}$$

for

$$\delta(\varepsilon) = \mathbb{P}\left(\frac{N_1+1}{N_2} \geq \mathrm{e}^{\varepsilon}\right), \tag{4.3}$$

where the randomness of (N_1, N_2) is determined by $\mathrm{View}_{\mathcal{M}}^{A_s}(X)$. Instead of being mutually independent binomially distributed random variables as argued in the proof of Balle et al. (2019, Thm. 3.1), we claim that N_1 and N_2 are distributed as follows.

Lemma 13. *Let the $\mathit{View}_{\mathcal{M}}^{A_s}(X)$ be defined as in Def. 12. N_1 and N_2 denote the number of outcomes of the first $n-1$ local randomisers that are results of randomisation and equal 1 and 2, respectively. Then the counts N_1 and N_2 are distributed as*

$$(N_1, N_2) \sim (A, C),$$

where $A \sim \mathrm{Bin}(n-1, \frac{\gamma}{k})$ and $C \sim \mathrm{Bin}(n-1-A, \frac{\gamma}{k-\gamma})$.

Proof. First, more generally, consider $n-1$ independent trials and random variables for the numbers of observations for three classes: N_1, N_2 and a remainder class, with corresponding probabilities p_1, p_2 and $1-p_1-p_2$. Then, the multinomial probability gives

$$\begin{aligned}
&\mathbb{P}\big((N_1, N_2) = (n_1, n_2)\big) \\
&= \frac{(n-1)!}{n_1! n_2! (n-1-n_1-n_2)!} p_1^{n_1} p_2^{n_2} (1-p_1-p_2)^{n-1-n_1-n_2} \\
&= \frac{(n-1)!}{n_1!(n-1-n_1)!} p_1^{n_1}(1-p_1)^{n-1-n_1} \cdot \frac{(n-1-n_1)!}{n_2!(n-1-n_1-n_2)!} \frac{p_2^{n_2}(1-p_1-p_2)^{n-1-n_1-n_2}}{(1-p_1)^{n-1-n_1}} \\
&= \frac{(n-1)!}{n_1!(n-1-n_1)!} p_1^{n_1}(1-p_1)^{n-1-n_1} \cdot \frac{(n-1-n_1)!}{n_2!(n-1-n_1-n_2)!} \left(\frac{p_2}{1-p_1}\right)^{n_2} \left(\frac{1-p_1-p_2}{1-p_1}\right)^{n-1-n_1-n_2} \\
&= \frac{n!}{n_1!(n-1-n_1)!} p_1^{n_1}(1-p_1)^{n-1-n_1} \cdot \left[\frac{(n-1-n_1)!}{n_2!(n-1-n_1-n_2)!} \left(\frac{p_2}{1-p_1}\right)^{n_2} \left(1-\frac{p_2}{1-p_1}\right)^{n-1-n_1-n_2}\right].
\end{aligned}$$

We recognise the probabilities of binomial distributions, and see that

$$(N_1, N_2) \sim (A, C),$$

where $A \sim \mathrm{Bin}(n-1, p_1)$ and $C \sim \mathrm{Bin}(n-1-A, \frac{p_2}{1-p_1})$. When $V \sim \mathrm{View}_{\mathcal{M}}^{A_s}(X)$, we can think of N_1 and N_2 as numbers of outcomes of $n-1$ independent trials where both classes have probabilities γ/k. Substituting $p_1 = p_2 = \gamma/k$ in the above formula shows the claim.

□

Using the reasoning of Balle et al. (2019, Thm. 3.1) (Equation 4.1), we can explicitly write the PLD which gives us tight (ε, δ)-bounds. Recall from Def. 5 that the privacy loss random variable for $\mathrm{View}_{\mathcal{M}}^{A_s}$ is given by

$$\omega_{X/X'}^{A_s} = \log\left(\frac{\mathbb{P}(\mathrm{View}_{\mathcal{M}}^{A_s}(X) = V)}{\mathbb{P}(\mathrm{View}_{\mathcal{M}}^{A_s}(X') = V)}\right), \quad V \sim \mathrm{View}_{\mathcal{M}}^{A_s}(X). \tag{4.4}$$

Using this definition of PRV, Equation 4.1 and Lemma 13, we get the following.

Theorem 14. *Consider the adversary A_s as given in Def 12. For all neighbouring datasets X and X', the PRV for $\mathit{View}_{\mathcal{M}}^{A_s}$ is given by*

$$\omega^{A_s} = \log\left(\frac{N_1+1}{N_2}\right),$$

where

$$(N_1, N_2) \sim (A, C),$$

and $A \sim \mathrm{Bin}(n-1, \frac{\gamma}{k})$ and $C \sim \mathrm{Bin}(n-1-A, \frac{\gamma}{k-\gamma})$.

Notice that this expression for ω^{A_s} is independent of any input to the local randomisers and holds for any neighbouring datasets X and X'. Therefore it allows computing tight $\delta(\varepsilon)$-bounds for adaptive compositions of the k-RR shuffler in case we assume the adversary of Def. 12.

4.1 Tight bounds for weaker adversaries

Following the reasoning used for analysing the bounds against the adversary A_s of Def. 12, we can compute tight $\delta(\varepsilon)$-bounds also for an adversary that has less information about the local randomisers. Having tight bounds also enables us to evaluate exactly how much different assumptions on the adversary cost us in terms of privacy. Instead of the adversary A_s we analyse a weaker adversary A_w, who has extra information only on the first $n-1$ parties. We formalise this as follows.

Definition 15. *Let $\mathcal{M} = \mathcal{S} \circ \mathcal{R}^{PH}_{\gamma,k,n}$ be the shuffled k-RR mechanism, and w.l.o.g. let the differing element be x_n. Adversary A_w is an adversary with the view*

$$View^{A_w}_{\mathcal{M}}(X) = \left((x_1, \dots, x_{n-1}), \quad \beta \in \{0,1\}^{n-1}, \quad (y_{\pi(1)}, \dots, y_{\pi(n)})\right),$$

where y are the outputs from the shuffler, β is a binary vector identifying which of the first $n-1$ parties answered randomly, and π is a uniformly random permutation applied by the shuffler.

Note that compared to the stronger adversary A_s formalised in Def. 12, the difference is only in the vector β. We write $b = \sum_i \beta_i$, and B for the corresponding random variable in the following. The next theorem gives the random variables we need to calculate privacy bounds for adversary A_w.

Theorem 16. *Consider the adversary A_w as given in Def 15. For all neighbouring datasets X and X', the PRV for $View^{A_w}_{\mathcal{M}}$ is given by*

$$\omega^{A_w} = \log\left(\frac{P_w}{Q_w}\right),$$

where

$$P_w = P_1 + P_2, \quad Q_w = Q_1 + Q_2, \tag{4.5}$$

and

$$\begin{aligned}
&P_1 \sim (1-\gamma) \cdot N_1|B, \quad P_2 \sim \frac{\gamma}{k} \cdot (B+1), \\
&Q_1 \sim (1-\gamma) \cdot N_2|N_1, B, \quad Q_2 \sim \frac{\gamma}{k} \cdot (B+1), \\
&B \sim \mathrm{Bin}(n-1, \gamma), \\
&N_1^B|B \sim \mathrm{Bin}\left(B, \frac{1}{k}\right), \\
&N_1|B \sim N_1^B|B + \mathcal{R}_n, \\
&\mathcal{R}_n \sim \mathrm{Bern}(1-\gamma+\gamma/k), \\
&N_2|N_1, B \sim \mathrm{Bin}\left(B+1-N_1|B, \frac{1}{k-1}\right).
\end{aligned}$$

As a direct corollary to this result, and analogously to the case of the adversary A_s, since the PLD ω^{A_w} is independent of any input to the local randomisers, we obtain tight $\delta(\varepsilon)$-bounds against the adversary A_w for adaptive compositions using ω^{A_w}.

4.2 Experimental comparison between specialized analysis of k-RR (Balle et al., 2019) and specialized Clones - analysis (Feldman et al., 2023)

In Figure 4 we compare the tight bounds obtained using the PRVs ω^{A_s} and ω^{A_w} with numerical FFT-based accounting, and the PLD obtained from the k-RR specific analysis of Feldman et al. (2023, Thm 5.2) combined with numerical accounting. We tune the parameters of the FFT-based numerical accounting so that the discretisation error is negligible. Notice that the underlying analysis of k-RR with the adversaries

A_s, A_w has stronger assumptions about the adversary than the analysis by Feldman et al. (2023), as the adversaries know which of the messages were randomised (except for the differing element in case of the weaker adversary A_w). For the weaker adversary A_w, we already obtain stronger guarantees than by using the analysis of Feldman et al. (2023).

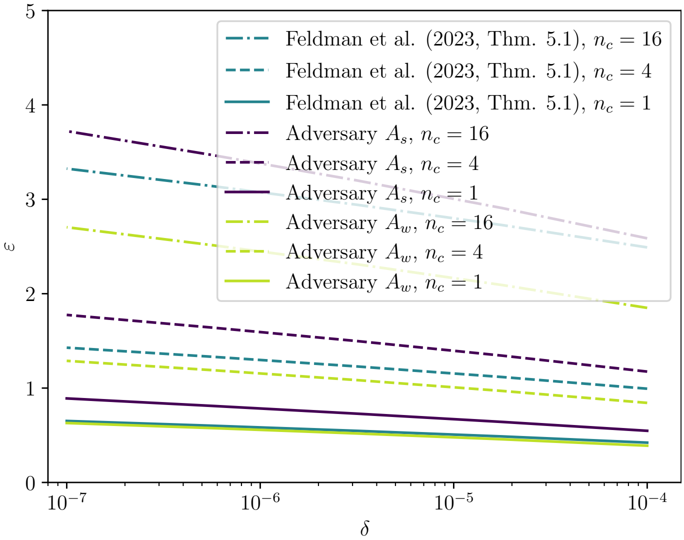


Figure 4: k-RR with the strong adversary A_s (PRV ω^{A_s} determined by Thm 14) and the weak adversary (PRV ω^{A_s} determined by Thm 16) and tight (ε, δ)-DP bounds obtained using FFT-accounting for different numbers of compositions n_c. Here $n = 1000$, probability of randomising $\gamma = 0.25$, and $k = 4$. Also shown are the bounds computed using the k-RR specific result by Feldman et al. (2023, Thm 5.2).

5 On the difficulty of obtaining bounds in the general case

We have provided means to compute accurate (ε, δ)-bounds for the general ε_0-LDP shuffler using the results by Feldman et al. (2023) and tight bounds for the case of k-randomised response. Using the following example, we illustrate the computational difficulty of obtaining tight bounds for arbitrary local randomisers.

Consider neighbouring datasets $X, X' \in \mathbb{R}^n$, where all elements of X are equal, and X' contains one element differing by 1. Without loss of generality (due to shifting and scaling invariance of DP), we may consider the case where X consists of zeros and X' has 1 at some element. Considering a mechanism $\mathcal{M}$ that consists of adding Gaussian noise with variance σ^2 to each element and then shuffling, we see that the adversary sees the output of $\mathcal{M}(X)$ distributed as $\mathcal{M}(X) \sim \mathcal{N}(0, \sigma^2 I_n)$, and the output $\mathcal{M}(X')$ as the mixture distribution $\mathcal{M}(X') \sim \frac{1}{n} \cdot \mathcal{N}(e_1, \sigma^2 I_n) + \ldots + \frac{1}{n} \cdot \mathcal{N}(e_n, \sigma^2 I_n)$, where e_i denotes the ith unit vector.

Determining the hockey-stick divergence $H_{e^\varepsilon}(\mathcal{M}(X')||\mathcal{M}(X))$ cannot be projected to a lower-dimensional problem, unlike in the case of the (subsampled) Gaussian mechanism, for example, which is equivalent to a one-dimensional problem. This means that in order to obtain tight (ε, δ)-bounds, we need to numerically evaluate the n-dimensional hockey-stick integral $H_{e^\varepsilon}(\mathcal{M}(X')||\mathcal{M}(X))$.

Using a numerical grid as in FFT-based accountants is unthinkable due to the curse of the dimensionality. However, we may use the fact that for any dataset X, the density function $f_X(t)$ of $\mathcal{M}(X)$ is a permutation-invariant function, meaning that for any $t \in \mathbb{R}^n$ and for any permutation $\sigma \in \pi_n$, $f_X\big(\sigma(t)\big) = f_X(t)$. This allows reducing the number of required points on a regular grid for the hockey stick integral from $O(m^n)$ to $O(m^n/n!)$, where m is the number of discretisation points in each dimension. Recent research on numerical integration of permutation-invariant functions (e.g. Nuyens et al., 2016) suggests it may be possible to significantly reduce or even eliminate the dependence on n using more advanced integration techniques.

In the Appendix C.2, we give results on experiments where we have computed $H_{\mathrm{e}^{\varepsilon}}(\mathcal{M}(X')||\mathcal{M}(X))$ using Monte Carlo integration on a hypercube $[-L, L]^n$ which requires $\approx 5 \cdot 10^7$ samples for getting two correct significant figures already for $n = 7$.

6 Discussion

We have shown how numerical privacy accounting with privacy loss distributions can be used to calculate accurate upper bounds for the compositions of various (ε, δ)-DP mechanisms, as well as for different adversaries in the shuffle model. An alternative accounting approach uses Rényi differential privacy (Mironov, 2017). We have carried out experimental comparisons between the RDP and the PLD approaches. As illustrated by the comparison against the results of Girgis et al. (2021) in Fig. 3, numerical PLD accounting can sometimes lead to considerably tighter bounds.

When comparing numerical and analytical privacy bounds, they are in many cases complementary and serve different purposes. Numerical accountants allow finding the tightest possible bounds for implementations and enable more unbiased comparison of algorithms when accuracy of accounting is not a factor. Analytical bounds enable theoretical research and understanding of scaling properties of algorithms, but the inaccuracy of the bounds raises the risk of misleading conclusions about privacy claims.

While our results provide improvements over previous state-of-the-art, they only provide optimal accounting for k-randomised response. Developing optimal accounting for more general mechanisms as well as extending the results to $(\varepsilon_0, \delta_0)$-LDP base mechanisms are important topics for future research.

Acknowledgments

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI; and grant 325573), the Strategic Research Council at the Academy of Finland (Grant 336032) as well as the European Union (Project 101070617). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

The authors would also like to thank the anonymous reviewers and the action editor at TMLR for noticing potential points of confusion in an earlier version, as well as for the helpful discussions.

References

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

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy profiles and amplification by subsampling. *Journal of Privacy and Confidentiality*, 10(1), 2020a.

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

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

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

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

Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *Proceedings on Privacy Enhancing Technologies*, 4:552–570, 2022.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

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

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science*. IEEE, 2021.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4966–4981. SIAM, 2023.

Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In Anne Canteaut and François-Xavier Standaert (eds.), *Advances in Cryptology – EUROCRYPT 2021*, pp. 463–488, Cham, 2021. Springer International Publishing. ISBN 978-3-030-77883-5.

Antonious Girgis, Deepesh Data, and Suhas Diggavi. Rényi differential privacy of the subsampled shuffle model in distributed learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems*, 2021.

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

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *International Conference on Artificial Intelligence and Statistics*, pp. 2560–2569. PMLR, 2020.

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

Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 650–661, 2012.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, Aug 2017. doi: 10.1109/CSF.2017.11.

Dirk Nuyens, Gowri Suryanarayana, and Markus Weimar. Rank-1 lattice rules for multivariate integration in spaces of permutation-invariant functions: Error bounds and tractability. *Advances in Computational Mathematics*, 42:55–84, 2016.

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

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*, pp. 4782–4817. PMLR, 2022.

A Auxiliary results for determining the PLD of general ε_0 shufflers

We recall the following from Section 3.1. Denoting $q = \mathrm{e}^{\varepsilon_0} p$ and $\widetilde{q} = \frac{p}{1-\mathrm{e}^{\varepsilon_0} p}$, $p = \frac{1}{\mathrm{e}^{\varepsilon_0}+1}$, the dominating pair of distributions (P, Q) is determined are given by the mixtures

$$P = q \cdot P_0 + (1-q)\widetilde{q} \cdot P_1 + (1-q)(1-\widetilde{q}) \cdot P_2, \quad Q = (1-q)\widetilde{q} \cdot P_0 + q \cdot P_1 + (1-q)(1-\widetilde{q}) \cdot P_2,$$

where

$$P_0 \sim (A+1, C-A), \quad P_1 \sim (A, C-A+1), \quad P_2 = (A, C-A)$$

and for $n \in \mathbb{N}$, A and C are as

$$C \sim \mathrm{Bin}(n-1, 2p), \quad A \sim \mathrm{Bin}\left(C, \tfrac{1}{2}\right), \quad \Delta_1 \sim \mathrm{Bern}\left(\mathrm{e}^{\varepsilon_0} p\right) \quad \text{and} \quad \Delta_2 \sim \mathrm{Bin}\left(1-\Delta_1, \tfrac{p}{1-\mathrm{e}^{\varepsilon_0} p}\right).$$

In this section we give the expressions needed to determine the PLD

$$\omega_{P/Q}(s) = \sum\nolimits_{a,b} \mathbb{P}(P = (a,b)) \cdot \delta_{s_{a,b}}(s), \quad s_{a,b} = \log\left(\frac{\mathbb{P}(P=(a,b))}{\mathbb{P}(Q=(a,b))}\right), \tag{A.1}$$

and similarly also $\omega_{Q/P}$.

A.1 Determining the log ratios $s_{a,b}$

To determine $s_{a,b}$'s, we need the following auxiliary results.

Lemma A.1. *When $b > 0$ and $a > 0$, we have:*

$$\mathbb{P}(P_0 = (a,b)) = \frac{a}{b} \cdot \mathbb{P}(P_1 = (a,b)).$$

Proof. We see that $P_0 = (a,b)$ if and only if $A = a-1$ and $C = a+b-1$. Since

$$\begin{aligned} \mathbb{P}(A = a-1 \,|\, C = a+b-1) &= \binom{a+b-1}{a-1} \frac{1}{2^{a+b-1}} \\ &= \frac{a}{b} \cdot \binom{a+b-1}{a} \frac{1}{2^{a+b-1}} \\ &= \frac{a}{b} \cdot \mathbb{P}(A = a \,|\, C = a+b-1), \end{aligned}$$

we see that

$$\begin{aligned} \mathbb{P}(P_0 = (a,b)) &= \mathbb{P}(C = a+b-1) \cdot \mathbb{P}(A = a-1 \,|\, C = a+b-1) \\ &= \mathbb{P}(C = a+b-1) \cdot \frac{a}{b} \cdot \mathbb{P}(A = a \,|\, C = a+b-1) \\ &= \frac{a}{b} \cdot \mathbb{P}(P_1 = (a,b)), \end{aligned}$$

since $P_1 = (a,b)$ if and only if $A = a$ and $C = a+b-1$. □

Lemma A.2. *When $b > 0$ and $a > 0$, we have:*

$$\mathbb{P}(P_0 = (a,b)) = \frac{(1-2p)a}{(n-a-b)p} \cdot \mathbb{P}(P_2 = (a,b)).$$

Proof. We see that $P_2 = (a,b)$ if and only if $A = a$ and $C = a+b$. Since

$$\begin{aligned} \mathbb{P}(C = a+b) &= \binom{n-1}{a+b} (2p)^{a+b} (1-2p)^{n-1-a-b} \\ &= \frac{2p}{1-2p} \binom{n-1}{a+b} (2p)^{a+b-1} (1-2p)^{n-1-a-b+1} \\ &= \frac{2p}{1-2p} \frac{n-a-b}{a+b} \binom{n-1}{a+b-1} (2p)^{a+b-1} (1-2p)^{n-1-a-b+1} \\ &= \frac{2p}{1-2p} \frac{n-a-b}{a+b} \cdot \mathbb{P}(C = a+b-1) \end{aligned}$$

and since

$$\mathbb{P}(A = a \,|\, C = a+b-1) = \binom{a+b-1}{a} \frac{1}{2^{a+b-1}} = \frac{2b}{a+b} \binom{a+b}{a} \frac{1}{2^{a+b}} = \frac{2b}{a+b} \cdot \mathbb{P}(A = a \,|\, C = a+b),$$

we see that

$$\begin{aligned}
\mathbb{P}(P_0 = (a,b)) &= \mathbb{P}(C = a+b-1) \cdot \mathbb{P}(A = a-1 \,|\, C = a+b-1) \\
&= \mathbb{P}(C = a+b-1) \cdot \frac{a}{b} \cdot \mathbb{P}(A = a \,|\, C = a+b-1) \\
&= \frac{(1-2p)a}{(n-a-b)p} \cdot \mathbb{P}(C = a+b) \cdot \frac{a}{b} \cdot \mathbb{P}(A = a \,|\, C = a+b) \\
&= \frac{(1-2p)a}{(n-a-b)p} \cdot \mathbb{P}(P_2 = (a,b)).
\end{aligned}$$

□

As a corollary of Lemmas A.1 and A.2 we get the following expressions with which we can also determine the log ratios $s_{a,b}$.

Corollary A.3. *We have:*

$$\mathbb{P}\big(P = (a,b)\big) = \left[q + (1-q)\widetilde{q} \cdot \frac{b}{a} + (1-q)(1-\widetilde{q}) \frac{(n-a-b)p}{(1-2p)a} \right] \cdot \mathbb{P}\big(P_0 = (a,b)\big)$$

and

$$\mathbb{P}\big(Q = (a,b)\big) = \left[q \cdot \frac{b}{a} + (1-q)\widetilde{q} + (1-q)(1-\widetilde{q}) \frac{(n-a-b)p}{(1-2p)a} \right] \cdot \mathbb{P}\big(P_0 = (a,b)\big).$$

Probabilities for the cases $a = 0$ and $b = 0$ become extremely small already for moderate values of n. When using the Hoeffding inequality based $O(n)$-approximation to determine the PLDs, we do not need to evaluate these probabilities so we do not consider writing them out.

Corollary A.3 gives $\mathbb{P}\big(P = (a,b)\big)$ and $\mathbb{P}\big(Q = (a,b)\big)$ in terms of $\mathbb{P}(P_0 = (a,b))$, and that is given by the following expression which we get by change of variables.

Lemma A.4. *When $a > 0$,*

$$\mathbb{P}(P_0 = (a,b)) = \binom{n-1}{i} \binom{i}{j} p^i (1-p)^{n-1-i} \frac{1}{2^i},$$

where $(a,b) = (j+1, i-j)$ (i.e., $C = i$ and $A = j$).

B More detailed proof of the Lemma: Lowering PLD computational complexity using Hoeffding's inequality

Using an appropriate tail bound (Hoeffding) for the binomial distribution, we can truncate part of the probability mass and add it directly to δ. More specifically, if each PLD ω_i, $1 \leq i \leq n_c$, in an n_c-composition is approximated by a truncated distribution $\widetilde{\omega}_i$ such that the truncated probability masses are $\tau_i \geq 0$, respectively, then

$$\delta(\varepsilon) = \widetilde{\delta}(\varepsilon) + \delta(\infty),$$

where $\widetilde{\delta}(\varepsilon)$ is the value of the hockey-stick divergence obtained with the truncated PLDs $\widetilde{\omega}_i$, $1 \leq i \leq n_c$, and where

$$\delta(\infty) = 1 - \prod_i (1 - \tau_i) \leq \sum_i \tau_i,$$

gives an upper bound for the composition without truncations, see e.g. Thm 1 in Sommer et al. (2019). Using the Hoeffding inequality we obtain an accurate approximation of $\omega_{P/Q}$ (or $\omega_{Q/P}$) with only $\mathcal{O}(n)$ terms. We formalise this approximation as follows.

Lemma 11. *Let $\tau > 0$. Consider the set*

$$S_n = [\max\left(0, (2p - c_n)(n-1)\right), \min\left(n-1, (2p + c_n)(n-1)\right)],$$

where $c_n = \sqrt{\frac{\log(4/\tau)}{2(n-1)}}$ and the set

$$S_i = [\max\left(0, (\tfrac{1}{2} - c_i) \cdot i\right), \min\left(n-1, (\tfrac{1}{2} + c_i) \cdot i\right)],$$

where $c_i = \sqrt{\frac{\log(4/\tau)}{2 \cdot i}}$. Then, the distribution $\widetilde{\omega}_{P/Q}$ defined by

$$\widetilde{\omega}_{P/Q}(s) = \sum_{i \in S_n} \sum_{j \in S_i} \mathbb{P}(P = (j+1, i-j)) \cdot \delta_{s_{j+1,i-j}}(s), \quad s_{a,b} = \log\left(\tfrac{\mathbb{P}(P=(a,b))}{\mathbb{P}(Q=(a,b))}\right) \tag{B.1}$$

has $\mathcal{O}(n \cdot \log(4/\tau))$ terms and differs from $\omega_{P/Q}$ at most mass τ.

Proof. Using Hoeffding's inequality for $C \sim \text{Bin}(n-1, 2p)$ states that for $c > 0$,

$$\begin{aligned} \mathbb{P}\big(C \leq (2p - c)(n-1)\big) &\leq \exp\big(-2(n-1)c^2\big), \\ \mathbb{P}\big(C \geq (2p + c)(n-1)\big) &\leq \exp\big(-2(n-1)c^2\big). \end{aligned}$$

Requiring that $2 \cdot \exp\left(-2(n-1)c^2\right) \leq \tau/2$ gives the condition $c \geq \sqrt{\frac{\log(4/\tau)}{2(n-1)}}$ and the expressions for c_n and S_n. Similarly, we use Hoeffding's inequality for $A \sim \text{Bin}(C, \frac{1}{2})$ and get expressions for c_i and S_i. The total neglegted mass is at most $\tau/2 + \tau/2 = \tau$. For the number of terms, we see that S_n contains at most $2c_n(n-1) = \sqrt{n-1}\sqrt{2 \cdot \log(4/\tau)}$ terms and for each i, S_i contains at most $2c_i i = \sqrt{i}\sqrt{2 \cdot \log(4/\tau)} \leq \sqrt{n-1}\sqrt{2 \cdot \log(4/\tau)}$ terms. Thus $\widetilde{\omega}_{P/Q}$ has at most $\mathcal{O}(n \cdot \log(4/\tau))$ terms. We get the expression of Equation B.1 by the change of variables $a = i + 1$ ($A = i$) and $b = i - j$ ($C = j$). □

C Auxiliary results for Section 4

C.1 Proof of Theorem 16

Theorem C.1. *Consider the adversary A_w as given in Def 15. For all neighbouring datasets X and X', the PRV for View$_{\mathcal{M}}^{A_w}$ is given by*

$$\omega^{A_w} = \log\left(\frac{P_w}{Q_w}\right),$$

where

$$P_w = P_1 + P_2, \quad Q_w = Q_1 + Q_2, \tag{C.1}$$

and

$$\begin{aligned} &P_1 \sim (1-\gamma) \cdot N_1 | B, \quad P_2 \sim \frac{\gamma}{k} \cdot (B+1), \\ &Q_1 \sim (1-\gamma) \cdot N_2 | N_1, B, \quad Q_2 \sim \frac{\gamma}{k} \cdot (B+1), \\ &B \sim \text{Bin}(n-1, \gamma), \\ &N_1^B | B \sim \text{Bin}\left(B, \frac{1}{k}\right), \\ &N_1 | B \sim N_1^B | B + \mathcal{R}_n, \\ &\mathcal{R}_n \sim \text{Bern}(1 - \gamma + \gamma/k), \\ &N_2 | N_1, B \sim \text{Bin}\left(B + 1 - N_1 | B, \frac{1}{k-1}\right). \end{aligned}$$

Proof. Assume w.l.o.g. that the differing elements are $x_n = 1, x_n' = 2$. Notice that for k-RR, seeing the shuffler output is equivalent to seeing the total counts for each class resulting from applying the local randomisers to X or X'. The adversary A_w can remove all truthfully reported values by client $j, j \in [n-1]$. Denote the observed counts after this removal by $n_i, i = 1, \dots, k$, so $\sum_{i=1}^{k} n_i = b + 1$.

We now have

$$\begin{aligned}
\mathbb{P}(\text{View}_{\mathcal{M}}^{A_w}(\mathbf{x}) = V) &= \sum_{i=1}^{k} \mathbb{P}(N_1 = n_1, \dots, N_i = n_i - 1, N_{i+1} = n_{i+1}, \dots N_k = n_k | B) \cdot \mathbb{P}(\mathcal{R}(x_n) = i) \cdot \mathbb{P}(B = b) \\
&= \binom{b}{n_1 - 1, n_2, \dots, n_k} \left(\frac{1}{k}\right)^b \cdot \left(1 - \gamma + \frac{\gamma}{k}\right) \cdot \gamma^b (1-\gamma)^{n-1-b} \\
&\quad + \sum_{i=2}^{k} \binom{b}{n_1, \dots, n_i - 1, n_{i+1}, \dots, n_k} \left(\frac{1}{k}\right)^b \cdot \frac{\gamma}{k} \cdot \gamma^b (1-\gamma)^{n-1-b} \\
&= \binom{b}{n_1, n_2, \dots, n_k} \frac{\gamma^b (1-\gamma)^{n-1-b}}{k^b} \left[n_1 (1 - \gamma + \frac{\gamma}{k}) + \sum_{i=2}^{k} n_i \frac{\gamma}{k} \right] \\
&= \binom{b}{n_1, n_2, \dots, n_k} \frac{\gamma^b (1-\gamma)^{n-1-b}}{k^b} \left[n_1 (1 - \gamma + \frac{\gamma}{k}) + (b + 1 - n_1) \frac{\gamma}{k} \right] \\
&= \binom{b}{n_1, n_2, \dots, n_k} \frac{\gamma^b (1-\gamma)^{n-1-b}}{k^b} \left[n_1 (1 - \gamma) + \frac{\gamma}{k}(b + 1) \right].
\end{aligned} \tag{C.2}$$

Noting then that $\mathbb{P}(\mathcal{R}_{\gamma,k,n}^{PH}(x_n') = i) = (1 - \gamma + \frac{\gamma}{k})$ when $i = 2$ and $\frac{\gamma}{k}$ otherwise, repeating essentially the same steps gives

$$\mathbb{P}(\text{View}_{\mathcal{M}}^{A_w}(X') = V) = \binom{b}{n_1, n_2, \dots, n_k} \frac{\gamma^b (1-\gamma)^{n-1-b}}{k^b} \left[n_2 (1 - \gamma) + \frac{\gamma}{k}(b + 1) \right]. \tag{C.3}$$

Looking at the ratio of the two final probabilities given in Equation C.2 and in Equation C.3, we get

$$\frac{\mathbb{P}(\text{View}_{\mathcal{M}}^{A_w}(X) = V)}{\mathbb{P}(\text{View}_{\mathcal{M}}^{A_w}(X') = V)} = \frac{N_1 | B \cdot (1-\gamma) + \frac{\gamma}{k}(B+1)}{N_2 | N_1, B \cdot (1-\gamma) + \frac{\gamma}{k}(B+1)},$$

where we write, e.g., $N_1|B$ for the random variables N_1 conditional on B. This shows that for DP bounds, the adversaries' full view is equivalent to only considering the joint distribution of (N_1, N_2, B), and we can therefore look at the neighbouring random variables

$$P_w = P_1 + P_2, \quad Q_w = Q_1 + Q_2, \tag{C.4}$$

where

$$\begin{aligned}
P_1 &\sim (1-\gamma) \cdot N_1 | B, \quad P_2 \sim \frac{\gamma}{k} \cdot (B+1), \\
Q_1 &\sim (1-\gamma) \cdot N_2 | N_1, B, \quad Q_2 \sim \frac{\gamma}{k} \cdot (B+1).
\end{aligned}$$

Writing N_i^B, $i = 1, 2$, for the count in class i resulting from the noise sent by the $n-1$ parties, and denoting by $\mathcal{R}_n$ a Bernoulli random variable s.t. $\mathcal{R}_n = 1$, if $\mathcal{R}(x_n) = 1$, similarly to the proof in case of the strong adversary, we have

$$B \sim \text{Bin}(n-1, \gamma), \qquad N_1^B | B \sim \text{Bin}\left(B, \frac{1}{k}\right), \qquad \mathcal{R}_n \sim \text{Bern}(1 - \gamma + \gamma/k). \tag{C.5}$$

As $V \sim \text{View}_{\mathcal{M}}^{A_w}(X)$, and

$$N_2 | N_1, \mathcal{R}_n, B \sim \begin{cases} \text{Bin}(B + 1 - N_1 | B, \frac{1}{k-1}), \text{ if } \mathcal{R}_n = 1, \\ \text{Bin}(B - N_1 | B, \frac{1}{k-1}) + \text{Bern}(\frac{1}{k-1}), \text{ if } \mathcal{R}_n = 0, \end{cases}$$

we finally have

$$N_1|B = N_1^B|B + \mathcal{R}_n, \quad N_2|N_1, B = \mathrm{Bin}(B + 1 - N_1|B, \frac{1}{k-1}). \tag{C.6}$$

The distributions of Equation C.5 and Equation C.6 determine the neighbouring distributions P_w and Q_w that are given in Equation C.4. This completes the proof. □

C.2 Experiment for Section 5

Consider neighbouring datasets $X, X' \in \mathbb{R}^n$, where all elements of X are equal, and X' contains one element differing by 1. Without loss of generality (due to shifting and scaling invariance of DP), we may consider the case where X consists of zeros and X' has 1 at some element. Considering a mechanism $\mathcal{M}$ that consists of adding Gaussian noise with variance σ^2 to each element and then shuffling, we see that the adversary sees the output of $\mathcal{M}(X)$ distributed as $\mathcal{M}(X) \sim \mathcal{N}(0, \sigma^2 I_n)$, and the output $\mathcal{M}(X')$ as the mixture distribution $\mathcal{M}(X') \sim \frac{1}{n} \cdot \mathcal{N}(e_1, \sigma^2 I_n) + \ldots + \frac{1}{n} \cdot \mathcal{N}(e_n, \sigma^2 I_n)$, where e_i denotes the ith unit vector. In order to obtain tight (ε, δ)-bounds, we need to numerically evaluate the n-dimensional hockey-stick integral $H_{\mathrm{e}^\varepsilon}(\mathcal{M}(X')||\mathcal{M}(X))$.

In Figure 5 we have computed $H_{\mathrm{e}^\varepsilon}(\mathcal{M}(X')||\mathcal{M}(X))$ up to $n = 7$ using Monte Carlo integration on a hypercube $[-L, L]^n$ which requires $\approx 5 \cdot 10^7$ samples for getting two correct significant figures for $n = 7$.

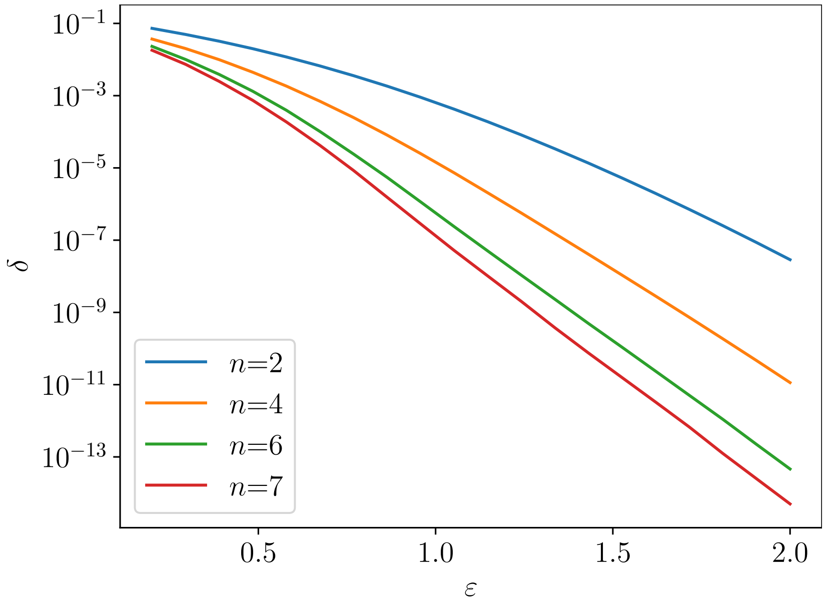


Figure 5: Approximation of tight $\delta(\varepsilon)$ for shuffled outputs of Gaussian mechanisms ($\sigma = 2.0$) by Monte Carlo integration of the hockey-stick divergence $H_{\mathrm{e}^\varepsilon}(\mathcal{M}(X')||\mathcal{M}(X))$, using $5 \cdot 10^7$ samples (two correct significant figures).

Paper IV

Mikko A. Heikkilä, Matthew Ashman, Siddharth Swaroop, Richard E. Turner and Antti Honkela

Differentially private partitioned variational inference

Transactions on Machine Learning Research, 2023.

IV

Differentially private partitioned variational inference

Mikko A. Heikkilä *mikkoaaro.heikkila@telefonica.com*
Telefónica Research

Matthew Ashman *mca39@cam.ac.uk*
Department of Engineering
University of Cambridge

Siddharth Swaroop *siddharth@seas.harvard.edu*
School of Engineering and Applied Sciences
Harvard University

Richard E. Turner *ret26@cam.ac.uk*
Department of Engineering
University of Cambridge

Antti Honkela *antti.honkela@helsinki.fi*
Department of Computer Science
University of Helsinki

Reviewed on OpenReview: https://openreview.net/forum?id=55BcghgicI

Abstract

Learning a privacy-preserving model from sensitive data which are distributed across multiple devices is an increasingly important problem. The problem is often formulated in the federated learning context, with the aim of learning a single global model while keeping the data distributed. Moreover, Bayesian learning is a popular approach for modelling, since it naturally supports reliable uncertainty estimates. However, Bayesian learning is generally intractable even with centralised non-private data and so approximation techniques such as variational inference are a necessity. Variational inference has recently been extended to the non-private federated learning setting via the partitioned variational inference algorithm. For privacy protection, the current gold standard is called differential privacy. Differential privacy guarantees privacy in a strong, mathematically clearly defined sense.

In this paper, we present differentially private partitioned variational inference, the first general framework for learning a variational approximation to a Bayesian posterior distribution in the federated learning setting while minimising the number of communication rounds and providing differential privacy guarantees for data subjects.

We propose three alternative implementations in the general framework, one based on perturbing local optimisation runs done by individual parties, and two based on perturbing updates to the global model (one using a version of federated averaging, the second one adding virtual parties to the protocol), and compare their properties both theoretically and empirically. We show that perturbing the local optimisation works well with simple and complex models as long as each party has enough local data. However, the privacy is always guaranteed independently by each party. In contrast, perturbing the global updates works best with relatively simple models. Given access to suitable secure primitives, such as secure aggregation or secure shuffling, the performance can be improved by all parties guaranteeing privacy jointly.

1 Introduction

Communication-efficient distributed methods that protect user privacy are a basic requirement for many machine learning tasks, where the performance depends on having access to sensitive personal data. Federated learning (Brendan McMahan et al., 2016; Kairouz et al., 2019) is a common approach for increasing communication efficiency with distributed data by pushing computations to the parties holding the data, thereby leaving the data where they are. While it has been convincingly demonstrated that federated learning by itself does not guarantee any kind of privacy (Zhu et al., 2019), it can be combined with differential privacy (DP, Dwork et al. 2006b; Dwork & Roth 2014), which does provide formal privacy guarantees.

In settings which require uncertainty quantification as well as high prediction accuracy, Bayesian methods are a natural approach. However, Bayesian posterior distributions are often intractable and need to be approximated. Variational inference (VI, Jordan et al. 1999; Wainwright et al. 2008) is a well-known and widely used approximation method based on solving a related optimisation problem; the most common formulation minimises the Kullback-Leibler divergence between the approximation and the true posterior.

In this paper, we focus on privacy-preserving federated VI in the cross-silo setting (Kairouz et al., 2019). We consider a common and important setup, where the parties (or 'clients') have sensitive horizontally partitioned data, i.e., local data with shared features, and the aim is to learn a single model on all the data. Such a setting arises, for example, when several hospitals want to train a joint model on common features without sharing their actual patient data with any other party. Our main problem is to learn a posterior approximation from the partitioned data under DP, while trying to minimise the amount of server-client communications.

We propose a general framework to solve the problem based on partitioned variational inference (PVI, Ashman et al. 2022). On a conceptual level, the main steps of PVI are the following: i) server sends current global model to clients, ii) clients perform local optimisation using their own data, iii) clients send updates to server, iv) server updates the global approximation. In our solution, called differentially private partitioned variational inference (DP-PVI), the clients enforce DP either independently or, given access to some suitable secure primitive, jointly with the other clients. We consider three different implementations of DP-PVI, one based on perturbing the local optimisation at step ii) of PVI (called DP optimisation), and two on perturbing the model updates at step iii) of PVI (called local averaging and virtual PVI clients).

Crucially, we empirically demonstrate that with our approaches the number of communication rounds between the server and the clients can be kept significantly lower than in the existing DP VI baseline solution, that requires communicating gradients for each regular VI optimisation step, while achieving nearly identical performance in terms of accuracy and held-out likelihood. Additionally, while the baseline requires communicating the gradients using some suitable secure primitive to achieve good utility, our approaches do not require any such primitives, although they can be easily combined with two of our approaches (local averaging and virtual PVI clients).[1] Finally, compared to the communication minimising baselines given by DP Bayesian committee machines, that require a single communication round between the server and the clients, our solutions provide clearly better prediction accuracies and held-out likelihoods in a variety of settings.

Our contribution Our main contributions are the following:

- We introduce DP-PVI, a communication-efficient general approach for DP VI on horizontally partitioned data.
- Within the general framework, we propose three differing implementations of DP-PVI, one based on perturbing local optimisation, termed DP optimisation, and two based on perturbing model updates, that we call local averaging and virtual PVI clients.
- Compared to the baseline of standard (global) DP VI, we experimentally show that our proposed implementations need orders of magnitude fewer server-client communication rounds, and can be

[1] In this paper, instead of considering any specific secure primitive implementation, such as a secure aggregator or a secure shuffler, we assume access to a black-box trusted aggregator in the comparisons.

trained without any secure primitives, while achieving comparable model utility under various settings. Compared to the DP Bayesian committee machine baselines, all our implementations can significantly improve on the resulting approximation quality under various models and datasets.

- We compare the relative advantages and disadvantages of our methods, both theoretically and experimentally, and make recommendations on how to choose the best method based on the task at hand:
  - We demonstrate that no single implementation outperforms the others in all settings.
  - We show that DP optimisation works well when there is enough local data available, on both simple and complex models. However, it not benefit from access to secure primitives. It can therefore lag behind the other methods when there are many clients without much local data, but with access to a trusted aggregator.
  - In contrast, local averaging and virtual PVI clients work best with relatively simple models, but can struggle with more complex ones. Since they can directly benefit from access to a trusted aggregator, they can outperform DP optimisation in a setting with many clients, little local data on each, and a trusted aggregator available. We find that using virtual PVI clients tends to be more stable than local averaging.

2 Related work

In the non-Bayesian setting, federated learning, with and without privacy concerns, has seen a lot of recent research (Kairouz et al., 2019).

VI without privacy has been considered in a wide variety of configurations, a considerable proportion of which can be interpreted as specific implementations of PVI. With just a single client, the local free-energy is equivalent to the global free-energy and global VI is recovered (Hinton & Van Camp, 1993). PVI with multiple clients unifies many existing local VI methods, in which each factor involves a subset of datapoints and may also include only a subset of variables over which the posterior is defined. This includes variational message passing (Winn et al., 2005) and its extensions (Knowles & Minka, 2011; Archambeau & Ermis, 2015; Wand, 2014), and is related to conjugate-computation VI (Khan & Lin, 2017). When only a single pass is made through the clients, PVI recovers online VI (Ghahramani & Attias, 2000), streaming VI (Broderick et al., 2013) and variational continual learning (Nguyen et al., 2018). See Ashman et al. (2022) for a more detailed overview of the relationships between PVI and these methods.

There is a rich literature on Bayesian learning with DP guarantees in various settings. Perturbing sufficient statistics in exponential family models has been applied both with centralised data (Dwork & Smith, 2010; Foulds et al., 2016; Zhang et al., 2016; Honkela et al., 2018) as well as with distributed data (Heikkilä et al., 2017). In the centralised setting, Dimitrakakis et al. (2014) showed that under some conditions, drawing samples from the true posterior satisfies DP. Posterior sampling under DP has been extended (Zhang et al., 2016; Geumlek et al., 2017; Dimitrakakis et al., 2017), and generalised also based on, e.g., Langevin and Hamiltonian dynamics (Wang et al., 2015; Li et al., 2019; Räisä et al., 2021) as well as on general Markov chain Monte Carlo (Heikkilä et al., 2019; Yıldırım & Ermiş, 2019). As an orthogonal direction for combining Bayesian learning with DP, Bernstein & Sheldon (2018) proposed a method for taking the DP noise into account when estimating the posterior with exponential family models to avoid model overconfidence.

DP-VI has been previously considered in the centralised setting. Jälkö et al. (2017) first introduced DP-VI for non-conjugate models based on DP-SGD, while Foulds et al. (2020) proposed a related variational Bayesian expectation maximization approach based on sufficient statistics perturbation for conjugate exponential family models.

Also in the centralised setting, Vinaroz & Park (2021) recently proposed DP stochastic expectation propagation, an alternative approximation method to VI that also has some close ties to the PVI framework (see e.g. Ashman et al. 2022), based on natural parameter perturbation. While there are several technical differences in how DP is guaranteed, and Vinaroz & Park (2021) do not discuss the distributed setting or propose a federated algorithm, we would expect that there is no fundamental reason why their approach could not be made to work in the federated setting as well, given enough changes to the centralised algorithm. With reasonable privacy parameters, we would expect comparisons to reflect the general properties of the underlying non-DP approaches (see e.g. Minka 2005; Li et al. 2015; Ashman et al. 2022 and the references therein for a discussion on the properties of non-DP variational and EP-style algorithms). The application of such methods to the private federated setting would be an interesting direction for future work.

In the non-Bayesian DP literature, the basic idea in our local averaging and virtual client approaches is close to the subsample and aggregate approach proposed by Nissim et al. (2007). In the same vein as our local averaging, Wei et al. (2020) proposed training a separate model for each single data point and combining the models by averaging the parameters in the context of learning neural network weights under DP. They do not consider other possible partitions or the trade-off between DP noise and the estimator variance.

3 Background

In this section we give a short overview of the most important background knowledge, starting with PVI in Section 3.1 and continuing with DP in Section 3.2.

3.1 Partitioned variational inference (PVI)

In Bayesian learning, we are interested in the posterior distribution

$$p(\theta|x) \propto p(x|\theta)p(\theta),$$

where $p(\theta)$ is a prior distribution, $p(x|\theta)$ is a likelihood, $x \in \mathcal{X}^n$ is some dataset of size n, and θ are the model parameters. Note that these are all different distributions, but we overload the notation in a standard way and identify the distributions by their arguments to keep the writing less cumbersome. For example, instead of $p_\theta(\theta)$ we simply write $p(\theta)$ for the prior. In this paper, we typically have $\theta \in \mathbb{R}^d$ for some d, and each element $x_i \in \mathbb{R}^{d'}, i = 1, \ldots, n$ for some d'.

When the posterior is in the exponential family of distributions, it is always tractable (see, e.g., Bernardo & Smith 1994):

Definition 1. *A distribution over $x \in \mathcal{X}^n$, indexed by a parameter vector $\theta \in \Theta \subset \mathbb{R}^d$ is an exponential family distribution, if it can be written as*

$$p(x|\theta) = h(x) \exp\left(T(x) \cdot \eta(\theta) - A(\eta(\theta))\right) \tag{3.1}$$

for some functions $h : \mathcal{X}^n \to \mathbb{R}, T : \mathcal{X}^n \to \mathbb{R}^d, A : \Theta \to \mathbb{R}$. When $\eta(\theta) = \eta$, the parameters η are called natural parameters, T are sufficient statistics, A is the log-partition function, and h is a base measure.

When the posterior is not in the exponential family, however, we need to resort to approximations. VI is a method for approximating the true posterior by solving an optimization problem over some tractable family of distributions (see e.g. Jordan et al. 1999 for an introduction to VI and Zhang et al. 2019 for a survey of more recent developments).

Writing $q(\theta|\lambda)$ for the approximating distribution parameterised with variational parameters $\lambda \in \mathbb{R}^{d_{VI}}$ for some d_{VI}, the main idea in VI is to find optimal parameters λ^*_{VI} that minimise some notion of distance between the approximation and the true posterior, with the most common choice being Kullback-Leibler divergence:

$$\lambda^*_{VI} = \arg\min_{q \in \mathcal{Q}} \left[D_{\mathrm{KL}}(q(\theta|\lambda) \| p(\theta|x))\right], \tag{3.2}$$

where $\mathcal{Q}$ is some tractable family of distributions. However, since the optimisation problem in Equation 3.2 is usually still not easy-enough for solving directly, the actual optimisation is typically done by maximising the so-called evidence lower bound (ELBO, also called negative variational free-energy):

$$\lambda^*_{VI} = \arg\max_{q \in \mathcal{Q}} \left[\mathbb{E}_q[\log p(x|\theta)] - D_{\mathrm{KL}}(q(\theta|\lambda) \| p(\theta))\right]. \tag{3.3}$$

It can be shown that the optimal solution λ^*_{VI} that solves Equation 3.3 also solves the original minimization problem in Equation 3.2.

In the setting we consider, there are M clients with shared features, and client j holds n_j samples. In the PVI framework (Ashman et al., 2022), this federated learning problem is solved iteratively. We start by defining the following variational approximation:

$$q(\theta|\lambda) = \frac{1}{Z_q} p(\theta) \prod_{j=1}^{M} t_j(\theta|\lambda_j) \simeq \frac{1}{Z} p(\theta) \prod_{j=1}^{M} p(x_j|\theta) = p(\theta|x), \tag{3.4}$$

where λ, λ_j are variational parameters, Z_q, Z are normalizing constants, x_j is the jth data shard, and t_j are client-specific factors refined over the algorithm run. The basic structure of the general PVI algorithm is given in Algorithm 1.

Note that in Algorithm 1, during the local optimisation (Equation 3.5) the cavity distribution (Equation 3.6) works as an effective prior: the variational parameters for the factors $t_j, j \neq m$ are kept fixed to their previous values.

Algorithm 1 Non-private PVI (Ashman et al., 2022)

```
Require: Number of global updates $S$, prior $p(\theta)$, initial client-specific factors $t_j^{(0)}, j = 1, \dots, M$.
for $s = 1$ to $S$ do
    Server chooses a subset $b^{(s)} \subseteq \{1, \dots, M\}$ of clients according to an update schedule and sends the current global model parameters $\lambda^{(s-1)}$.
    Each chosen client $m \in b^{(s)}$ finds a new set of parameters by optimising the local ELBO:
        $$\lambda^* = \arg\max_{q \in \mathcal{Q}} \left[ \mathbb{E}_q[\log p(x_m|\theta)] - D_{\mathrm{KL}}(q(\theta|\lambda) \| p_{\setminus m}^{(s-1)}(\theta)) \right], \tag{3.5}$$
    where $p_{\setminus m}^{(s-1)}$ is the so-called cavity distribution:
        $$p_{\setminus m}^{(s-1)}(\theta) \propto p(\theta) \prod_{j \neq m}^{M} t_j(\theta|\lambda_j^{(s-1)}). \tag{3.6}$$
    Each chosen client sends an update $\Delta t_m^{(s)}(\theta)$, given by
        $$\Delta t_m^{(s)}(\theta) \propto \frac{t_m(\theta|\lambda_m^*)}{t_m(\theta|\lambda_m^{(s-1)})} \propto \frac{q(\theta|\lambda^*)}{q(\theta|\lambda^{(s-1)})}, \tag{3.7}$$
    to the server.
    Server updates the global model by incorporating the updated local factors:
        $$q(\theta|\lambda^{(s)}) \propto q(\theta|\lambda^{(s-1)}) \prod_{m \in b^{(s)}} \Delta t_m^{(s)}(\theta).$$
end for
return Final variational approximation $q(\theta|\lambda^{(S)})$.
```

As a high-level overview, the PVI learning loop consists of the server sending current model to clients, the clients finding new local parameters via local optimisation, the clients sending an update to the server, and the server updating the global approximation.

Depending on the update schedule different variants of PVI are possible. In this paper, we use *sequential* PVI, where each client is visited in turn, and *synchronous* PVI, where all clients update in parallel (see Ashman et al. 2022 for more discussion on the PVI variants). The main idea in PVI is that the information from other sites is transmitted to client m via the other t-factors, while client m runs optimisation with purely local data. This reduces the number of server-client communications by pushing more computation to the clients.

Ashman et al. (2022) show that PVI has several desirable properties that connect it to the standard non-distributed (global) VI. Most importantly, optimising the local ELBO as in Equation 3.5 can be shown to be equivalent to a variational KL optimisation, and a fixed point of PVI is guaranteed to be a fixed point of the global VI.

3.2 Differential privacy (DP)

DP is essentially a robustness guarantee for stochastic algorithms (see e.g. Dwork & Roth 2014 for an introduction to DP and discussion on the definition of privacy). Formally we have the following:

Definition 2 (Dwork et al. 2006b;a)**.** *Let $\varepsilon > 0$ and $\delta \in [0, 1]$. A randomised algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{O}$ is (ε, δ)-DP if for every neighbouring $x, x' \in \mathcal{X}^n$ and every measurable set $E \subset \mathcal{O}$,*

$$\Pr(\mathcal{A}(x) \in E) \leq \mathrm{e}^{\varepsilon} \Pr(\mathcal{A}(x') \in E) + \delta.$$

The basic idea in the definition is that any single individual should only have a limited effect on the output. When this is guaranteed, the privacy of any given individual is protected, since the result would have been nearly the same even if that individual's data had been replaced by an arbitrary sample. Definition 2 formalises this idea by requiring that the probability of seeing any given output is nearly the same with any closely-related input dataset (the neighbouring datasets x, x'). The actual level of protection depends on the privacy parameters ε, δ: larger values mean less privacy.

The type and granularity of the privacy guarantee can be tuned by choosing an appropriate neighbourhood definition. Typical examples include sample-level (x, x' differ by a single sample) and user-level (x, x' differ by a single user's data) neighbourhoods. In this work, we use the bounded neighbourhood definition, which is also known as substitution neighbourhood, and assume that each individual has a single sample in the full combined training data, i.e., datasets x, x' are neighbours, if $|x| = |x'|$, and they differ by a single sample. With these definitions, individual privacy guarantees correspond to sample-level DP.

DP has several nice properties as a privacy guarantee, but the most important ones for our purposes are composability (repeated use of the same sensitive data erodes the privacy guarantees in a controllable manner), and immunity to post-processing (if the output of a stochastic algorithm is DP, then any stochastic or deterministic post-processing results in the same or stronger DP guarantees).

We use the well-known Gaussian mechanism, that is, adding i.i.d. Gaussian noise with equal variance to each component of a query vector, as a basic privacy mechanism:

Definition 3 (Gaussian mechanism, Dwork et al. 2006a)**.** *Let $f : \mathcal{X}^n \to R^d$ be a function s.t. for neighbouring $x, x' \in \mathcal{X}^n$, there exists a constant $C > 0$ satisfying*

$$\sup_{x,x'} \|f(x) - f(x')\|_2 \leq C.$$

A randomised algorithm $\mathcal{G} : \mathcal{G}(x) = f(x) + \xi$, where $\xi \sim \mathcal{N}(0, \sigma I_d)$ is called the Gaussian mechanism.

When a privacy mechanism, e.g., the Gaussian mechanism, is run by first subsampling a minibatch of the full data, and running the mechanism using only the minibatch instead of the full data, the mechanism is referred to as a *subsampled mechanism.* For subsampling, we use sampling without replacement:

Definition 4 (Sampling without replacement)**.** *A randomised function $WOR_b : \mathcal{X}^n \to \mathcal{X}^b$ is a* sampling without replacement *subsampling function, if it maps a given dataset into a uniformly random subset of size b of the input data.*

The main benefit for privacy when using data subsampling is the effect of privacy amplification, i.e., the additional randomisation due to the subsampling enhances the privacy guarantees depending on the subsampling method and the *subsampling fraction* given by $q_{sample} = \frac{b}{n}$, where b is the minibatch size and n

is the total data size in Definition 4. Given a base mechanism $\mathcal{A}_\sigma$ and a minibatch size b, the subsampled mechanism using sampling without replacement is the combined mechanism $\mathcal{A}_\sigma \circ WOR_b$.

To quantify the total privacy resulting from (iteratively) running (subsampled) DP algorithms, we use the following privacy accounting oracle:

Definition 5 (Accounting Oracle)**.** *An* accounting oracle *is a function $\mathbb{O}$ that evaluates (ϵ, δ)-DP privacy bounds for compositions of (subsampled) mechanisms. Specifically, given δ, a sub-sampling ratio $q_{sample} \in (0, 1]$, the number of iterations $T \geq 1$ and a base mechanism $\mathcal{A}_\sigma$, the oracle gives an ϵ, such that a T-fold composition of $\mathcal{A}_\sigma$ using sub-sampling with ratio q_{sample} is (ϵ, δ)-DP, i.e.,*

$$\mathbb{O} : (\delta, q_{sample}, T, \mathcal{A}_\sigma) \mapsto \epsilon.$$

In the experiments, we use the Fourier accountant (Koskela et al., 2020) as an accounting oracle to keep track of the privacy parameters, since it can numerically establish an upper bound for the total privacy loss with a given precision level.

4 Differentially private partitioned variational inference

In the setting we consider, there are M parties or clients connected to a central server, with client j holding some amount n_j of data (we assume there is exactly one sample per individual protected by DP in the full joint data) with common features (horizontal data partitioning). The clients do not want to share their data directly but agree to train a model given DP guarantees. The DP guarantees are enforced on a sample-level, that is, we assume that any given individual we want to protect has a single data sample that is held by exactly one client. The central server aims to learn a single model from the clients' data, while minimising the number of communication rounds between the server and the clients.

As discussed in Section 3.1, the PVI framework allows for effectively reducing the number of global communication rounds by pushing more computation to the clients. This also enables several options for guaranteeing DP on the client side, either by each client alone or jointly with the other clients via secure primitives.

We consider two general approaches the clients can use for enforcing DP in PVI learning:

1. Perturbing the local optimisation (step 3 in Algorithm 1),
2. Perturbing the model parameter updates (step 4 in Algorithm 1).

The first option, which we term *DP optimisation*, relies on the fact that at each optimisation step in Algorithm 1, the local ELBO in Equation 3.5 only depends on the local data at the given client. To guarantee DP independently of others, each client can therefore perturb the local optimisation with a suitable DP mechanism. In practice, this approach can be implemented, e.g., using DP-stochastic gradient descent (DP-SGD) as we show in Section 4.1.

For the second option, since a given client only affects the global model through the parameter updates at step 4 in Algorithm 1, each client can enforce DP by perturbing the update, either independently or jointly with the other clients. Besides the naive *parameter perturbation*, we propose two improved alternatives in Section 4.2. We call these approaches *local averaging* and adding *virtual PVI clients*.

In this paper, instead of considering any particular secure primitive like secure aggregation (see e.g. Shamir 1979; Rastogi & Nath 2010) or secure shuffling (see e.g. Chaum 1981; Cheu et al. 2019), we assume a black-box trusted aggregator capable of summing reals, where necessary. In these cases we also assume that the clients themselves are honest, i.e., they follow the protocol and do not try to gain additional information during the protocol run (or honest but curious, that is, they follow the protocol but will try to gain information such as actual noise values used for DP randomisation during the protocol run, with minor modifications to the relevant bounds). Any actual implementation would need to handle problems arising, for example, from finite precision (see e.g. Agarwal et al. 2021; Chen et al. 2022 and references therein for a discussion on implementing distributed DP). These considerations apply equally to all variants, and hence do not affect their comparisons or our main conclusions. We leave these issues to future work.

Table 1 highlights the most important DP noise properties of our proposed solutions: whether the DP noise level can be affected by the local data size (intuitively, we could hope that guaranteeing DP with plenty of local data gives better utility), and whether the approach can benefit from access to a trusted aggregator (this enables the clients to guarantee DP jointly, so the total noise level can be less than when every client enforces DP independently).

	noise scale affected by local data size	benefit from a trusted aggregator
DP optimisation	✓	x
parameter perturbation	x	✓
local averaging	✓	✓
virtual PVI clients	✓	✓

Table 1: Properties of DP-PVI approaches

In the rest of this section we state the formal DP guarantees for each approach and discuss their properties. For ease of reading, since the proofs are fairly straight-forward, all proofs as well as the properties of non-DP local averaging can be found in Appendix A.

4.1 Privacy via local optimisation: DP optimisation

To guarantee DP during local optimisation, one option is to use differentially private stochastic gradient descent (DP-SGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016): for every local optimisation step, we clip each per-example gradient and then add Gaussian noise with covariance $\sigma^2 I$ to the sum. The formal privacy guarantees are stated in Theorem 6.

Theorem 6. *Running DP-SGD for client-level optimisation in Algorithm 1, using subsampling fraction $q_{sample} \in (0, 1]$ on the local data level for T local optimisation steps in total, with S global updates interleaved with the local steps, the resulting model is (ε, δ)-DP, with $\delta \in (0, 1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample}, T, \mathcal{G}_\sigma)$.*

Proof. See proof A.1 in the Appendix. □

Although DP-SGD in general is not guaranteed to converge, there are some known utility guarantees in the empirical risk minimization (ERM) framework, e.g., for convex and strongly convex loss functions (Bassily et al., 2014). It has also been empirically shown to work well on a number of problems with non-convex losses, such as learning neural network weights (Abadi et al., 2016).

In our setting, DP-SGD can directly benefit from increasing local data size on a given client via the subsampling amplification property: adding more local data while keeping the batch size fixed results in a smaller sampling fraction q_{sample} and hence gives better privacy.

In contrast, when using DP-SGD with a limited communication budget, it is non-trivial to derive direct privacy benefits from adding more clients to the setting. This is the case even when we assume access to a trusted aggregator, since the gradients of the local ELBO in Equation 3.5 only depend on a single client's data.

4.2 Privacy via model updates

To guarantee DP when communicating an update from client m to the server at global update s, we can clip and perturb the change in model parameters corresponding to $\Delta t_m^{(s)}$ at step 4 in Algorithm 1 directly. This naive *parameter perturbation* approach often results in having to add unpractical amounts of noise to each query, which severely degrades the model utility. The problem arises because the local data size in this case will typically have no direct effect either on the DP noise level or on the query sensitivity.

To improve the results by allowing the local data size to have a direct effect on the noise addition, we propose two possible approaches that generalise the naive parameter perturbation: i) *local averaging* and ii) adding *virtual PVI clients*. Both are based on partitioning the local data into non-overlapping shards and optimising a separate local model on each, but they differ on the objective functions and on how the local results are combined after training for a global model update. Additionally, virtual PVI clients with DP requires all virtual factors to be in a common exponential family.[2] As a limiting case, when using a single local data partition both methods are equivalent to the naive parameter perturbation.

Assuming a trusted aggregator, with both of our proposed methods we can scale the noise level with the total number of clients in the protocol using $\mathcal{O}(MS)$ server-client communications, where M is the number of clients and S the total number of global updates, the same number as running non-DP PVI with synchronous updates.

Next, we present the methods and show that they guarantee DP, starting with local averaging in Section 4.2.1 and continuing with virtual PVI clients in Section 4.2.2.

4.2.1 Local averaging

Algorithm 2 describes the main steps needed for running (non-private) PVI with local averaging.

Algorithm 2 PVI with local averaging

```
1: Each client $m = 1, \dots, M$ partitions its local data into $N_m$ non-overlapping shards.
2: for $s = 1$ to $S$ do
3:     Server chooses a subset $b^{(s)} \subseteq \{1, \dots, M\}$ of clients according to an update schedule and sends the current global model parameters $\lambda^{(s-1)}$.
4:     Each chosen client $m \in b^{(s)}$ finds $N_m$ sets of new parameters by optimising the local objectives all starting from a common initial value (the previous global model parameters $\lambda^{(s-1)}$):
           $$\lambda^*_{m_k} = \arg\max_{q \in \mathcal{Q}} \left[ \mathbb{E}_q[\log p(x_{m,k}|\theta)] - \frac{1}{N_m} D_{\mathrm{KL}}(q(\theta|\lambda) \| p^{(s-1)}_{\setminus m}(\theta)) \right], \quad k = 1, \dots, N_m, \qquad (4.1)$$
       where $p^{(s-1)}_{\setminus m}$ is the cavity distribution as in Equation 3.6. The new parameters used for calculating an update for client $m$ in PVI are given by the local average:
           $$\lambda^* = \frac{1}{N_m} \sum_{k=1}^{N_m} \lambda^*_{m_k}. \qquad (4.2)$$
5:     Each chosen client sends the update $\Delta t^{(s)}_m(\theta)$, defined as
           $$\Delta t^{(s)}_m(\theta) \propto \frac{t_m(\theta|\lambda^*_m)}{t_m(\theta|\lambda^{(s-1)}_m)} \propto \frac{q(\theta|\lambda^*)}{q(\theta|\lambda^{(s-1)})},$$
       to the server.
6:     Server updates the global model by incorporating the updated local factors:
           $$q(\theta|\lambda^{(s)}) \propto q(\theta|\lambda^{(s-1)}) \prod_{m \in b^{(s)}} \Delta t^{(s)}_m(\theta).$$
7: end for
8: return Final variational approximation $q(\theta|\lambda^{(S)})$.
```

[2] Non-DP PVI with synchronous updates has a similar restriction: all factors need to be in the same exponential family as the prior due to issues with proper normalization, see Ashman et al. 2022. Therefore, when using synchronous PVI server all our approaches, including DP optimisation, inherit this assumption as well.

Note that the objective in Equation 4.1 is the regular PVI local ELBO where the KL-term is re-weighted to reflect the local partitioning. This is equivalent to using the PVI objective with a tempered (cold) likelihood $p(x_{m,k}|\theta)^{N_m}$. In Appendix A, we show that PVI with local averaging has the same fundamental properties, with minor modifications, as regular PVI. For example, with local averaging the local ELBO optimisation is equivalent to a variational KL optimisation, and a (local) optimum for local averaging is also an optimum for global VI.

DP with local averaging Assuming $t_j, j = 1, \dots, M$ are exponential family factors, in the client update at step 5 in Algorithm 2, we can write

$$\begin{aligned}
\Delta t_m^{(s)}(\theta) &= \Delta \lambda_m^* && (4.3)\\
&= \lambda^* - \lambda^{(s-1)} && (4.4)\\
&= \frac{1}{N_m} \sum_{k=1}^{N_m} \lambda_{m_k}^* - \lambda^{(s-1)} && (4.5)\\
&= \frac{1}{N_m} \sum_{k=1}^{N_m} \left(\lambda_{m_k}^* - \lambda^{(s-1)} \right). && (4.6)
\end{aligned}$$

We then have the following for guaranteeing DP with local averaging:

Theorem 7. *Assume the change in the model parameters $\|\lambda_{m_k}^* - \lambda^{(s-1)}\|_2 \leq C, k = 1, \dots, N_m$ for some known constant C, where $\lambda_{m_k}^*$ is a proposed solution to Equation 4.1, and $\lambda^{(s-1)}$ is the vector of common initial values. Then releasing $\Delta\hat{\lambda}_m^*$ is (ε, δ)-DP, with $\delta \in (0,1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, 1, \mathcal{G}_\sigma)$, when*

$$\Delta\hat{\lambda}_m^* = \frac{1}{N_m} \Big[\sum_{k=1}^{N_m} \left(\lambda_{m_k}^* - \lambda^{(s-1)} \right) + \xi \Big], \tag{4.7}$$

where $\xi \sim \mathcal{N}(0, \sigma^2 \cdot I)$.

Proof. See A.6 in the Appendix. □

For quantifying the total privacy for S global updates using local averaging, we immediately have the following:

Corollary 8. *A composition of S global updates with local averaging using a norm bound C for clipping is (ε, δ)-DP, with $\delta \in (0,1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, S, \mathcal{G}_\sigma)$.*

Proof. See A.7 in the Appendix. □

As is clear from Corollary 8, with local averaging we pay a privacy cost for each global update, while the local optimisation steps are free. This the opposite of the DP optimisation result in Theorem 6. As mentioned in Corollary 8, in practice we generally need to guarantee the norm bound in Theorem 7 by clipping the change in the model parameters.[3]

Considering how increasing the local data size affects the DP noise level, we have the following:

Theorem 9. *With local averaging, the DP noise standard deviation can be scaled as $\mathcal{O}(\frac{1}{N_m})$, where N_m is the number of local partitions. Therefore, the effect of DP noise will vanish on the local factor level when the local dataset size and the number of local partitions grow.*

Proof. See A.8 in the Appendix. □

[3] We could also enforce DP (including without exponential family factors) by clipping and adding noise directly to the parameters instead of privatising the change in parameters; the clipping would then enforce the parameters to an ℓ_2-norm ball of radius C around the origin.

Note that Theorem 9 does not say that the DP noise will necessarily vanish on the global approximation level if one client gets more data and does more local partitions, since the total noise level depends on the factors from all the clients. Looking only at Theorem 9, it would seem like increasing the number of local partitions is always beneficial as it decreases the DP noise effect. However, this is not generally the full picture. Zhang et al. (2013) have shown that under some assumptions, the convergence rate of mean estimators (similar to the one we propose) will deteriorate when the number of partitions increases too much. In effect, having fewer samples from which to estimate each local set of parameters increases the estimator variance, which hurts convergence. We have experimentally confirmed this effect with local averaging (see Figure 5 in the Appendix).

The optimal number of local partitions therefore usually balances the decreasing DP noise level with the increasing estimator variance. However, as we show in Theorem 10, there are important special cases, such as the exponential family, where there is no trade-off since the number of local partitions can be increased without affecting the non-DP posterior.

Theorem 10. *Assume the effective prior $p_{\setminus j}(\eta)$, and the likelihood $p(x_j|\eta), j \in \{1, \dots, M\}$ are in a conjugate exponential family, where η are the natural parameters. Then the number of partitions used in local averaging does not affect the non-DP posterior.*

Proof. See A.9 in the Appendix. □

Finally, Theorem 11 shows that assuming a trusted aggregator, the global approximation noise level can stay constant when adding clients to the protocol, i.e., increasing the number of clients allows every individual client to add less noise while getting the same global DP guarantees.

Theorem 11. *Using local averaging with M clients and a shared number of local partitions $N_j = N \ \forall j$ assume the clients have access to a trusted aggregator. Then for any given privacy parameters ε, δ, the noise standard deviation added by a single client can be scaled as $\mathcal{O}(\frac{1}{\sqrt{M}})$ while guaranteeing the same privacy level.*

Proof. See A.10 in the Appendix. □

4.2.2 Virtual PVI clients

Running (non-private) PVI with virtual clients is described Algorithm 3.

The full local factor for client m is now $t_m = \prod_{k=1}^{N_m} t_{m,k}$, which is updated only through the virtual factors, and only the change in the full product is ever communicated to the server. This means that when all the virtual factors for client m are in the same exponential family,[4] the parameters for the full local factor t_m are given by

$$\lambda_m = \sum_{k=1}^{N_m} \lambda_{m,k}, \tag{4.10}$$

where $\lambda_{m,k}$ are the parameters for the kth virtual factor, and $\Delta t_m^{(s)}(\theta)$ at step 5 in Algorithm 2 can be written as

$$\Delta \lambda_m^* = \sum_{k=1}^{N_m} (\lambda_{m_k}^* - \lambda^{(s-1)}).$$

With virtual PVI clients without DP, doing both local and global updates synchronously corresponds to a regular non-DP PVI run with a synchronous server and $\sum_{j=1}^{M} N_j$ clients. Therefore, all the regular PVI properties (Ashman et al., 2022) derived with a synchronous server immediately hold for non-DP PVI with added virtual clients. In particular, the local ELBO optimisation in this case is equivalent to a variational KL optimisation, and any optimum of the algorithm is also an optimum for global VI.

[4] With DP, having all factors from a single exponential family is required to bound the sensitivity.

Algorithm 3 PVI with virtual clients

```
Each client $m = 1, \dots, M$ partitions it's local data into $N_m$ non-overlapping shards and creates corresponding virtual clients, i.e., separate factors $t_{m,k}$ with parameters $\lambda_{m,k}, k = 1, \dots, N_m$.
for $s = 1$ to $S$ do
    Server chooses a subset $b^{(s)} \subseteq \{1, \dots, M\}$ of clients according to an update schedule and sends the current global model parameters $\lambda^{(s-1)}$.
    Each chosen client $m \in b^{(s)}$ updates its virtual clients by locally simulating a single regular PVI update (steps 3-4 in Algorithm 1) with synchronous update schedule $b_m^{(s)} = \{1, \dots, N_m\}$. The optimised parameters for the $k$th virtual client are given by
        $$\lambda^*_{m_k} = \arg\max_{q \in \mathcal{Q}} \left[ \mathbb{E}_q[\log p(x_{m,k}|\theta)] - D_{\mathrm{KL}}(q(\theta|\lambda) \| p^{(s-1)}_{\setminus m,k}(\theta)) \right], \quad k = 1, \dots, N_m, \qquad (4.8)$$
    where $p^{(s-1)}_{\setminus m,k}$ is the cavity distribution:
        $$p^{(s-1)}_{\setminus m,k}(\theta) \propto p(\theta) \prod_{j \neq m}^{M} t_j(\theta|\lambda_j^{(s-1)}) \prod_{k' \neq k}^{N_m} t_{m,k'}(\theta|\lambda^{(s-1)}_{m,k'}). \qquad (4.9)$$
    Each chosen client $m$ updates the local factor and sends the update $\Delta t_m^{(s)}(\theta)$, defined as
        $$\Delta t_m^{(s)}(\theta) \propto \frac{t_m(\theta|\lambda^*_m)}{t_m(\theta|\lambda_m^{(s-1)})} \propto \prod_{k=1}^{N_m} \frac{q(\theta|\lambda^*_{m_k})}{q(\theta|\lambda^{(s-1)})},$$
    to the server.
    Server updates the global model by incorporating the updated local factors:
        $$q(\theta|\lambda^{(s)}) \propto q(\theta|\lambda^{(s-1)}) \prod_{m \in b^{(s)}} \Delta t_m^{(s)}(\theta).$$
end for
return Final variational approximation $q(\theta|\lambda^{(S)})$.
```

DP with virtual PVI clients For ensuring DP with virtual clients, again via noising the change in the model parameters as in Section 4.2.1, we have:

Theorem 12. *Assume the change in the model parameters $\|\lambda^*_{m_k} - \lambda^{(s-1)}\|_2 \leq C, k = 1, \dots, N_m$ for some known constant C, where $\lambda^*_{m_k}$ is a proposed solution to Equation 4.8, and $\lambda^{(s-1)}$ is the vector of common initial values. Then releasing $\Delta\tilde{\lambda}^*_m$ is (ε, δ)-DP, with $\delta \in (0, 1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, 1, \mathcal{G}_\sigma)$, when*

$$\Delta\tilde{\lambda}^*_m = \sum_{k=1}^{N_m} \left(\lambda^*_{m_k} - \lambda^{(s-1)} \right) + \xi, \qquad (4.11)$$

where $\xi \sim \mathcal{N}(0, \sigma^2 \cdot I)$.

Proof. See A.11 in the Appendix. □

As an immediate result, Corollary 13 quantifies the total privacy when doing S global updates using virtual PVI clients:

Corollary 13. *A composition of S global updates with virtual PVI clients using a norm bound C for clipping is (ε, δ)-DP, with $\delta \in (0, 1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, S, \mathcal{G}_\sigma)$.*

Proof. See A.12 in the Appendix. □

As with local averaging in Corollary 8, and contrasting with DP optimisation in Theorem 6, using Corollary 13 we pay a privacy cost for each global update, but the local optimisation steps are free. And as with local averaging, we usually need to guarantee the assumed norm bound by clipping.

Note that unlike with local averaging in Theorem 9, the noise variance in Equation 4.11 will stay constant with increasing number of local partitions N_m. Increasing the number of partitions will decrease the relative effect of the noise if it increases the non-DP sum.

Assuming access to a trusted aggregator, Theorem 14 is a counterpart to Theorem 11 with local averaging: again, the global approximation noise level can stay constant when adding clients to the protocol, meaning that each individual client needs to add less noise while maintaining the same global DP guarantees. The main difference is that with virtual PVI clients each client can choose the number of local partitions freely.

Theorem 14. *Assume there are M real clients adding virtual clients, and access to a trusted aggregator. Then for any given privacy parameters ε, δ, the noise standard deviation added by a single client can be scaled as $\mathcal{O}(\frac{1}{\sqrt{M}})$ while guaranteeing the same privacy level.*

Proof. See proof A.13 in the Appendix. □

4.3 Summary of technical contributions

We have presented three different implementations of DP-PVI: *DP optimisation*, that is based on perturbing the local optimisation in Section 4.1, as well as *local averaging* in Section 4.2.1 and adding *virtual PVI clients* in Section 4.2.2, which are both based on perturbing the global model updates.

The main idea in DP optimisation is to replace the non-DP optimisation procedure by a DP variant, our main choice being DP-SGD. Hence, DP optimisation inherits all the properties of standard DP-SGD, such as utility guarantees with convex and strongly convex losses. In the more general case of non-convex losses, DP optimisation has no known utility guarantees. Considering the privacy guarantees, with DP optimisation each client enforces DP independently (see Theorem 6), while the global model guarantees result from parallel composition.

In contrast, local averaging and virtual PVI clients are both based on the general idea of adding local data partitioning to mitigate the utility loss from DP noise: each client trains several models on disjoint local data shards, and then combines them for a single global update.

We first showed that local averaging does not fundamentally break the general properties of PVI (see Appendix A): the local ELBO optimisation can be interpreted as a variational KL optimisation, and an optimum of the local averaging algorithm is an optimum for global VI. Privacy for local averaging can be guaranteed by clipping and noising (the change in) the local parameters (see Theorem 7). We showed that the local average under DP approaches the non-DP average on the local factor level when the number of local data shards increases (Theorem 9), and that in the special case of conjugate-exponential family there is no price for increasing the number of local data shards (Theorem 10). Finally, we showed that given access to a suitable secure primitive, we can leverage the other clients to guarantee DP jointly, thereby reducing the amount of noise added by each client while keeping the global model guarantees unchanged (Theorem 11).

Adding virtual PVI clients without DP inherits the properties of non-DP PVI with synchronous updates (e.g., the local ELBO optimisation can be interpreted as a variational KL optimisation, and an optimum of the virtual PVI clients algorithm is an optimum for global VI). We can guarantee privacy via clipping and noising (the change in) the local parameters (Theorem 12). We also showed that, as with local averaging, assuming a suitable secure primitive and guaranteeing DP jointly, the amount of noise added by each client can be reduced while keeping the global model guarantees unchanged (Theorem 14).

Next, we test our proposed methods in Section 5 under various settings to see how they perform in practice.

5 Experiments

In this Section we empirically test our proposed methods using logistic regression and Bayesian neural network (BNN) models. Our code for running all the experiments is openly available from `https://github.com/DPBayes/DPPVI`.

We utilise a mean-field Gaussian variational approximation in all experiments. For datasets and prediction tasks, we employ UCI Adult (Kohavi, 1996; Dua & Graff, 2017) predicting whether an individual has income $> 50k$, as well as balanced MIMIC-III health data (Johnson et al., 2016b;a; Goldberger et al., 2000) with an in-hospital mortality prediction task (Harutyunyan et al., 2019).

An important and common challenge in federated learning is that the clients can have very different amounts of data, which cannot usually be assumed to be i.i.d. between the clients. We test the robustness of our approaches to such differences in the data held by each client by using two unbalanced data splits besides the balanced split. In the balanced case, the data is split evenly between all the clients. In both unbalanced data cases half of the clients have a significantly smaller share of data than the larger clients. In the first alternative the small clients only have a few minority class examples, while in the second one they have a considerably larger fraction than the large clients. We defer the details of the data splitting to Appendix B.

In all experiments, we use sequential PVI when not assuming a trusted aggregator, and synchronous PVI otherwise. The number of communications is measured as the number of server-client message exchanges performed by all clients. The actual wall-clock times would depend on the method and implementation: with sequential PVI only one client can update at any one time but communications do not need encryption, while with synchronous PVI all clients can update at the same time but the trusted aggregator methods would also need to account for the time taken by the secure primitive in question. All privacy bounds are calculated numerically with the Fourier accountant (Koskela et al., 2020).[5] The reported privacy budgets include only the privacy cost of the actual learning, while we ignore the privacy leakage due to hyperparameter tuning. More details on the experiments can be found in Appendix B.

We use two baselines: i) DP Bayesian committee machines (BCM with same and split prior, Tresp 2000; Ashman et al. 2022), which are trained with DP-SGD and use a single global communication round to aggregate DP results from all clients, and ii) a centralised DP-VI, which can be trained in our setting with DP-SGD when we assume a trusted aggregator without any communication limits (global VI, trusted aggr., Jälkö et al. 2017).

To measure performance, we report prediction accuracy on held-out data, as well as model log-likelihood as we are interested in uncertainty quantification. Model likelihood effectively measures how well the model can fit unseen data and whether it knows where it is likely to be wrong.

Logistic regression The logistic regression model likelihood is given by

$$p(y|\theta, x) = \sigma(\tilde{x}^\top \theta)^y (1 - \sigma(\tilde{x}^\top \theta))^{1-y},$$

where $y \in \{0, 1\}, \sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, and $\tilde{x} = [1, x^\top]^\top$ is the augmented data vector with a bias term. In the experiments, we use a standard normal prior for the weights $p(\theta) = \mathcal{N}(\theta|0, I)$ and Monte Carlo (MC) approximate the posterior predictive distribution.

Figures 1 & 2 show the results for logistic regression on Adult and balanced MIMIC-III datasets, respectively, for the three different data splits for 10 clients.

Global VI (global VI, trusted aggr.) is a very strong model utility baseline, that is approximately the best we could hope for. However, as is evident from the results, achieving this baseline requires a trusted aggregator and it uses orders of magnitude more communications than the DP-PVI implementations.[6]

The minimal single-round communication baselines are given by the two BCM variants (BCM same, BCM split). While they are very communication-efficient, the utility varies markedly between the different

[5] Available from `https://github.com/DPBayes/PLD-Accountant/`.

[6] Note that the results for global VI do not depend on the data split and hence this baseline curve is the same for all the splits.

settings and they are outperformed by the DP-PVI methods (DP optimisation, local avg, virtual) in several experiments.

For DP-PVI methods, we report results separately without a trusted aggregator (DP optimisation, local avg, virtual) and with a trusted aggregator (local avg, trusted aggr.; virtual, trusted aggr.). In terms of communications, all DP-PVI methods are on par with each other: requiring around an order of magnitude more communications than the minimal communication BCM baselines, and two orders less than the strong utility global VI baseline that always assumes a trusted aggregator.

Our DP optimisation method performs overall well in terms of model utility, always performing slightly worse than the strong utility global VI baseline (global VI, trusted aggr.), being on par with DP-PVI using virtual clients (virtual), and regularly outperforming the BCM baselines (BCM same, BCM split) as well as DP-PVI with local averaging (local avg).

The performance of our local averaging method without access to a trusted aggregator (local avg) seems unstable: it lags behind our other two methods (DP optimisation, virtual) as well as the minimal communication BCM methods (especially BCM split) in several experiments. Given access to a trusted aggregator (local avg, trusted aggr.) the performance improves, as can be expected from the noise scaling in Theorem 11, in several settings significantly so. While it now outperforms our DP optimisation (which does not benefit from the trusted aggregator) and the BCM baselines in many settings, it still regularly lags behind our virtual PVI clients with a trusted aggregator (virtual, trusted aggr.).

Our virtual PVI clients with no trusted aggregator (virtual) performs consistently well in terms of model utility: it lags somewhat behind the strong utility baseline (global VI, trusted aggr.), is on par with our DP optimisation, and outperforms both our local averaging (local avg) and the BCM baselines (BCM same, BCM split) in several experiments. With a trusted aggregator (virtual, trusted aggr.) the model utility improves in line with the noise scaling in Theorem 14, sometimes by a clear margin and even reaching the strong utility baseline (global VI, trusted aggr.) in several settings.

While the results for our local averaging and virtual PVI clients in Figures 1& 2 improve given access to a trusted aggregator even with a very limited communication budget (compare local avg vs. local avg, trusted aggr. and virtual vs. virtual, trusted aggr.), the benefits of having the DP noise scale according to Theorems 11& 14 become more marked with less local data and more clients. Figure 3 shows the results for Adult data with 200 clients: our DP optimisation, which does not benefit from access to a trusted aggregator, now performs clearly worse than local averaging (local avg, trusted aggr.) or virtual PVI clients (virtual, trusted aggr.) which use a trusted aggregator. In contrast to the results in Figures 1& 2, in the more demanding setting in Figure 3, all of our DP-PVI methods clearly outperform the minimum communication BCM baselines (BCM same, BCM split) in model utility.

BNNs We expect local averaging and virtual PVI clients to work best when the model has a single, well-defined optimum, since then the local data partitioning should not have a major effect on the resulting model. To test the methods with a more complex model, we use a BNN with one fully connected hidden layer of 50 hidden units, and ReLU non-linearities. Writing $f_\theta(x)$ for the output from the network with parameters θ and input $x \in \mathbb{R}^d$, the likelihood for $y \in \{0,1\}$ is a Bernoulli distribution $p(y|\theta,x) = Bern(y|\pi)$, where the class probability $\pi = \sigma^{-1}(f_\theta(x))$, i.e., the logit is predicted by the network. We use a standard normal prior on the weights $p(\theta) = \mathcal{N}(\theta|0,I)$, and MC approximate the predictive distribution:

$$p(y^* = 1|x^*, x, y) \simeq \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} p(y^* = 1|\theta^{(i)}, x^*), \theta^{(i)} \sim q(\theta) \; \forall i.$$

The results are shown in Figure 4 for Adult data and 10 clients. The minimum communication baselines (BCM same, BCM split) are very poor in terms of model utility: to achieve good utility with the BNN model in this setting requires some communication between the clients (compare to BCM same and BCM split results in Figure 1 using a tighter privacy budget but simpler logistic regression model). Our DP optimisation works very well in terms of model utility, being almost on par with the high-utility baseline that uses a trusted aggregator (global VI, trusted aggr.). In contrast, both our local averaging (local avg) and virtual

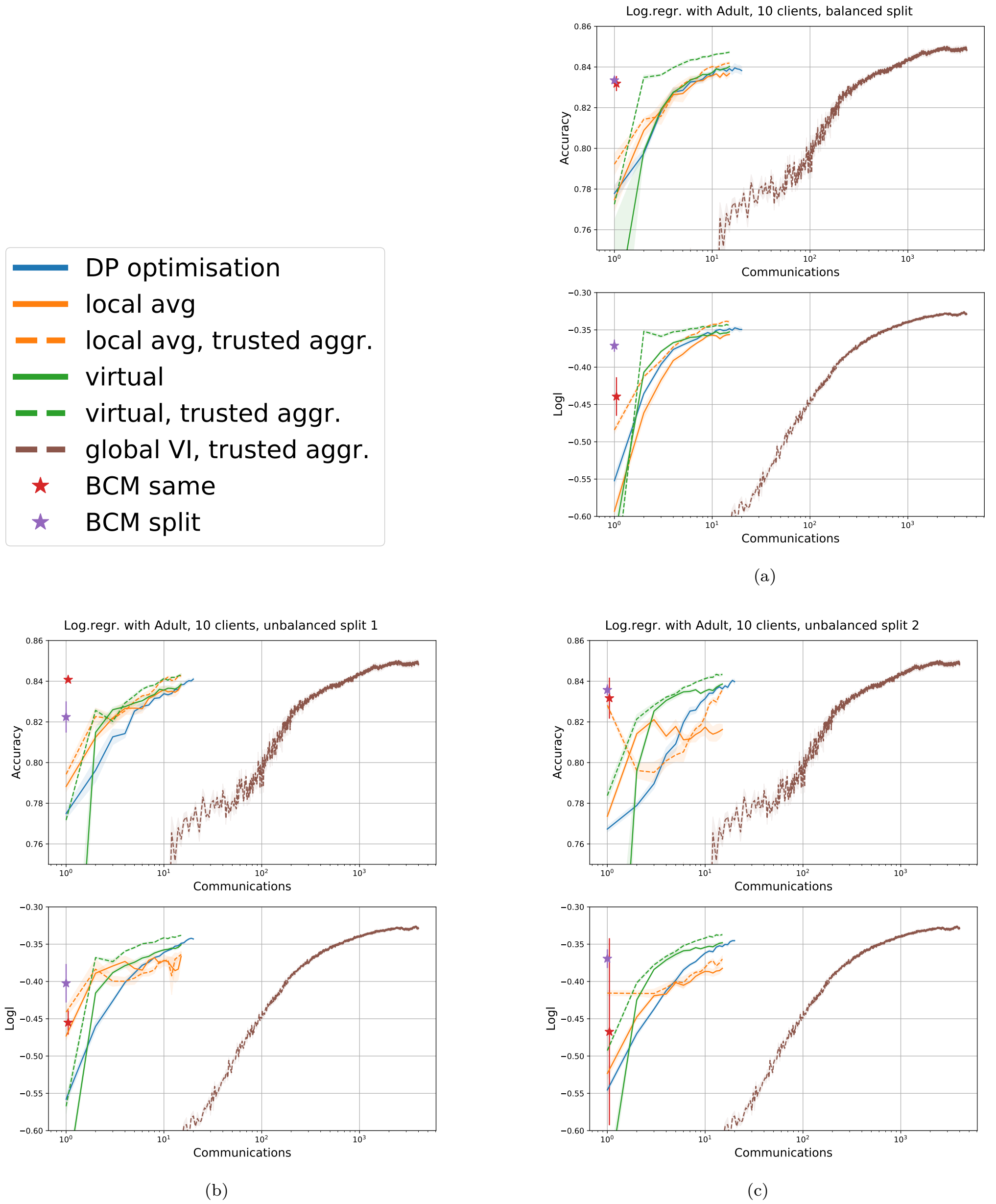
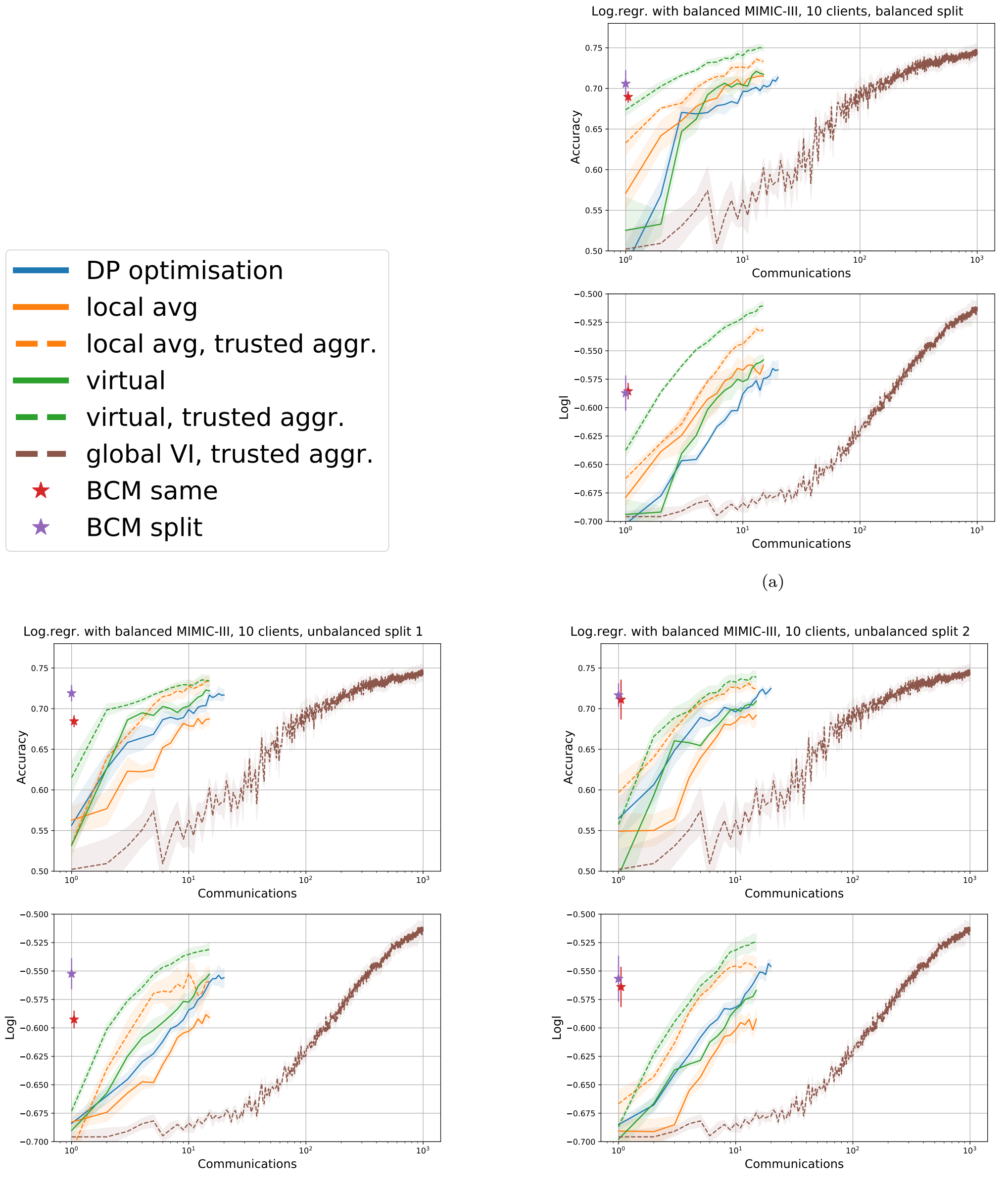


Figure 1: $(1, 10^{-5})$-DP logistic regression, Adult data with 10 clients: mean over 5 seeds with SEM. a) balanced split, b) unbalanced split 1, c) unbalanced split 2.

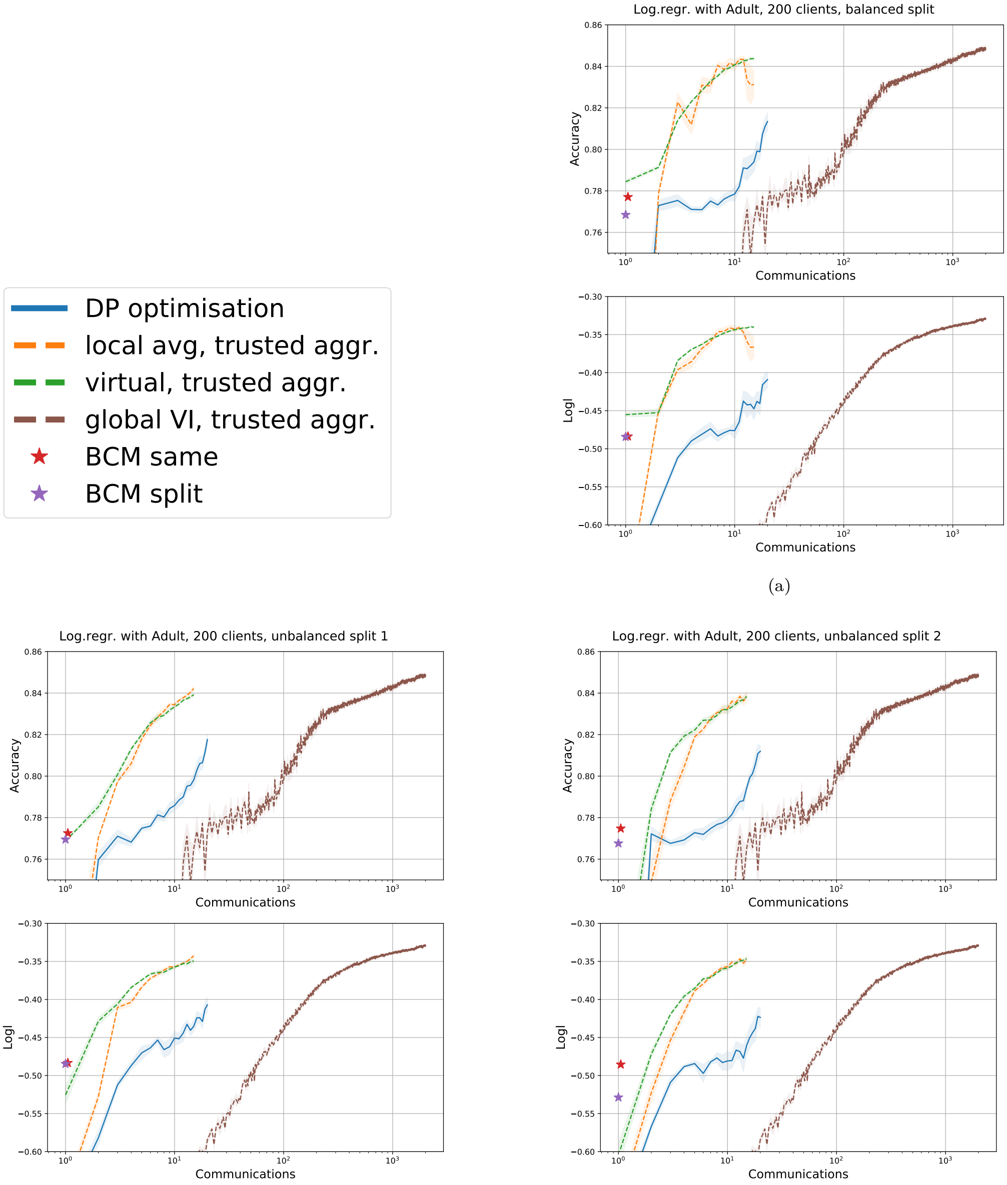


(a)

(b)

(c)

Figure 2: $(1, 10^{-5})$-DP logistic regression, balanced MIMIC-III data with 10 clients: mean over 5 seeds with SEM. a) balanced split, b) unbalanced split 1, c) unbalanced split 2.



(a)

(b)

(c)

Figure 3: $(\frac{1}{2}, 10^{-5})$-DP logistic regression with Adult data and 200 clients: mean over 5 seeds with SEM. a) balanced split, b) unbalanced split 1, c) unbalanced split 2.

PVI clients (virtual) perform clearly worse. They are also much more likely to diverge and seem to require generally more careful tuning of the hyperparameters to reach any reasonable performance.

6 Discussion

We have proposed three different implementations of DP-PVI, one based on perturbing the local optimisation (DP optimisation), and two based on perturbing the model updates (local averaging and virtual PVI clients). As is clear from the empirical results in Section 5, no single method dominates the others in all settings. Instead, the method needs to be chosen according to the problem at hand. However, we can derive guidelines for choosing the most suitable method based on the theoretical results as well as on the empirical experiments.

DP optimisation is a good candidate method with simple or complex models regardless of the number of clients as long as the clients have enough local data. However, with little local data and more clients, since it cannot easily benefit from secure primitives, the performance can be sub-optimal.

In contrast, local averaging and virtual PVI clients work best with relatively simple models, while the performance with more complex models can easily lag behind DP optimisation results. However, given access to a trusted aggregator both methods can leverage other clients to reduce the amount of DP noise required. Based on our experiments, from the two methods local averaging is harder to tune properly and can be unstable whereas using virtual PVI clients is a more robust alternative.

Acknowledgments

The authors would like to thank Mrinank Sharma, Michael Hutchinson, Stratis Markou and Antti Koskela for useful discussions. The authors acknowledge CSC – IT Center for Science, Finland, and the Finnish Computing Competence Infrastructure (FCCI) for computational and data storage resources.

Mikko Heikkilä was at the University of Helsinki during this work.

Matthew Ashman is supported by the George and Lilian Schiff Foundation.

Siddharth Swaroop was at the University of Cambridge during this work, where he was supported by an EPSRC DTP studentship and a Microsoft Research EMEA PhD Award.

Richard E. Turner is supported by Google, Amazon, ARM, Improbable and EPSRC grant EP/T005386/1.

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI; and grant 325573), the Strategic Research Council at the Academy of Finland (Grant 336032) as well as the European Union (Project 101070617). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

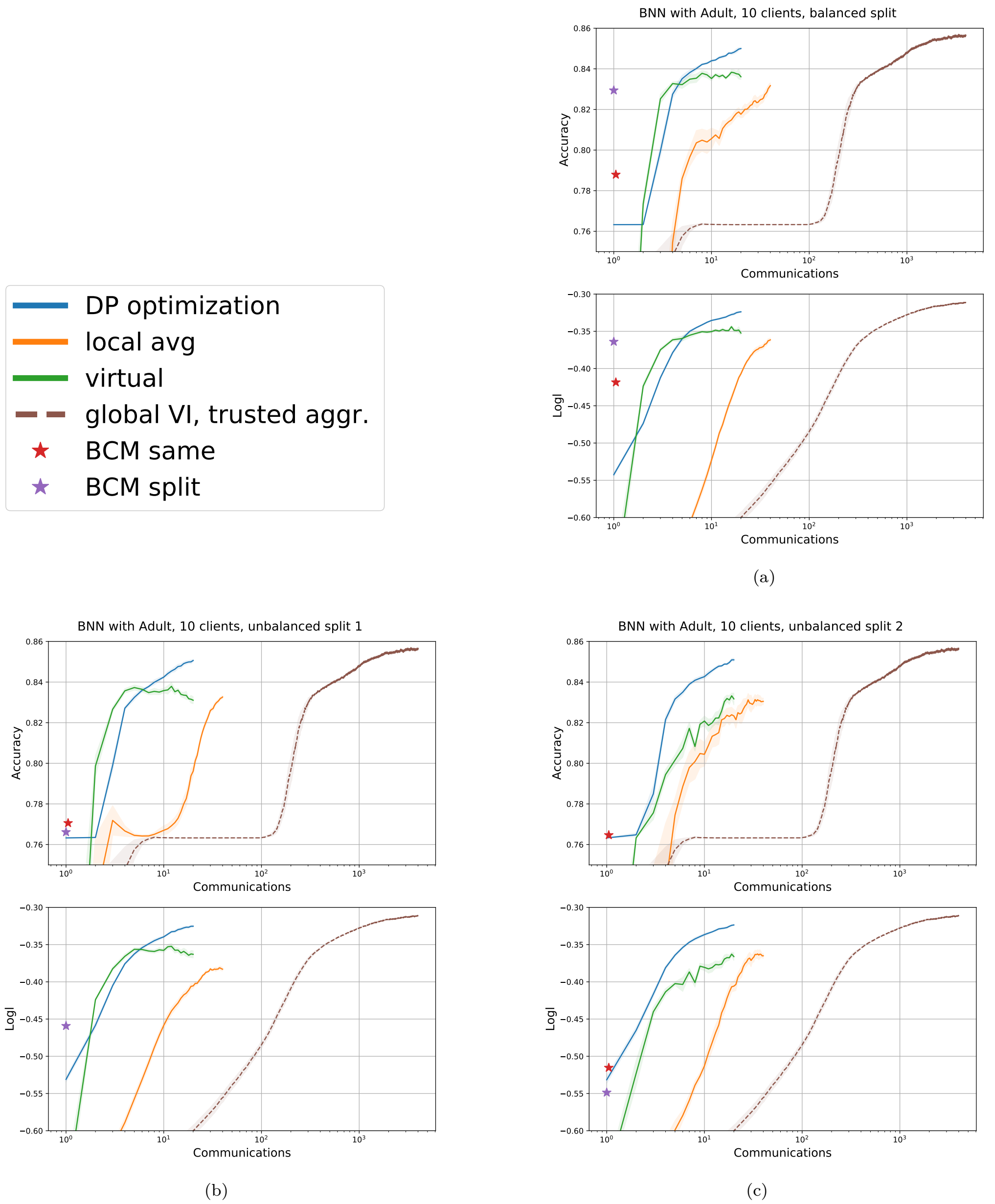
References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Naman Agarwal, Peter Kairouz, and Ziyu Liu. The Skellam mechanism for differentially private federated learning. October 2021.

Cedric Archambeau and Beyza Ermis. Incremental variational inference for latent Dirichlet allocation. *arXiv preprint arXiv:1507.05016*, 2015.

Matthew Ashman, Thang D. Bui, Cuong V. Nguyen, Stratis Markou, Adrian Weller, Siddharth Swaroop, and Richard E. Turner. Partitioned variational inference: A framework for probabilistic federated learning. 2022. doi: 10.48550/ARXIV.2202.12275. URL `https://arxiv.org/abs/2202.12275`.



(a)

(b)

(c)

Figure 4: $(2, 10^{-5})$-DP 1-layer BNN with Adult data and 10 clients: mean over 5 seeds with SEM. a) balanced split, b) unbalanced split 1, c) unbalanced split 2.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pp. 464–473, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-6517-5. doi: 10.1109/FOCS.2014.56. URL http://dx.doi.org/10.1109/FOCS.2014.56.

José M Bernardo and Adrian F M Smith. *Bayesian Theory.* John Wiley & Sons, Inc., 1994.

Garrett Bernstein and Daniel Sheldon. Differentially private bayesian inference for exponential families. In S. Bengio and H. Wallach and H. Larochelle and K. Grauman and N. Cesa-Bianchi and R. Garnett (ed.), *Advances in Neural Information Processing Systems*, volume 31, 2018.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient learning of deep networks from decentralized data. February 2016.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, 2013.

Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. November 2018.

David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24 (2):84–90, February 1981.

Wei-Ning Chen, Christopher A Choquette-Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. March 2022.

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

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Robust and private Bayesian inference. In *Proc. ALT 2014*, pp. 291–305. 2014.

Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Differential privacy for Bayesian inference through posterior sampling. *Journal of Machine Learning Research*, 18(11):1–39, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

Cynthia Dwork and Adam Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2), Apr. 2010. doi: 10.29012/jpc.v1i2.570. URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/570.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. TCC 2006*, pp. 265–284. 2006b. ISBN 978-3-540-32732-5. doi: 10.1007/11681878_14. URL http://dx.doi.org/10.1007/11681878_14.

James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proc. 32nd Conf. on Uncertainty in Artificial Intelligence (UAI 2016)*, 2016.

James R. Foulds, Mijung Park, Kamalika Chaudhuri, and Max Welling. Variational Bayes in private settings (VIPS) (extended abstract). In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 5050–5054. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/705. URL `https://doi.org/10.24963/ijcai.2020/705`. Journal track.

Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. Rényi differential privacy mechanisms for posterior sampling. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5295–5304, USA, 2017. Curran Associates Inc.

Zoubin Ghahramani and H. Attias. Online variational Bayesian learning. In *NIPS Workshop on Online Learning*, 2000.

A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J.E. Mietus, G.B. Moody, C.K Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23), 2000.

Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(96), June 2019.

Mikko Heikkilä, Eemil Lagerspetz, Samuel Kaski, Kana Shimizu, Sasu Tarkoma, and Antti Honkela. Differentially private bayesian learning on distributed data. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/dfce06801e1a85d6d06f1fdd4475dacd-Paper.pdf`.

Mikko Heikkilä, Joonas Jälkö, Onur Dikmen, and Antti Honkela. Differentially private markov chain monte carlo. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/074177d3eb6371e32c16c55a3b8f706b-Paper.pdf`.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Conference on Computational Learning Theory*, pp. 5–13, 1993.

Antti Honkela, Mrinal Das, Arttu Nieminen, Onur Dikmen, and Samuel Kaski. Efficient differentially private learning improves drug sensitivity prediction. *Biology Direct*, 13(1):1, 2018.

Joonas Jälkö, Antti Honkela, and Onur Dikmen. Differentially private variational inference for non-conjugate models. In *Proc. UAI 2017*, 2017. URL `http://auai.org/uai2017/proceedings/papers/152.pdf`.

Alistair Johnson, Tom Pollard, and Roger Mark. MIMIC-III clinical database (version 1.4). PhysioNet, 2016a. URL `https://doi.org/10.13026/C2XW26`.

Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(160035), May 2016b.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn

Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. 2019. doi: 10.48550/ARXIV.1912.04977. URL https://arxiv.org/abs/1912.04977.

Mohammad Emtiyaz Khan and Wu Lin. Conjugate-computation variational inference : Converting variational inference in non-conjugate models to inferences in conjugate models. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

David A. Knowles and Tom Minka. Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, pp. 1701–1709, 2011.

Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 202–207. AAAI Press, 1996.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *International Conference on Artificial Intelligence and Statistics*, pp. 2560–2569. PMLR, 2020.

Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient mcmc and differential privacy. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 557–566. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/li19a.html.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f3bd5ad57c8389a8a1a541a76be463bf-Paper.pdf.

Tom Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, January 2005. URL https://www.microsoft.com/en-us/research/publication/divergence-measures-and-message-passing/.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*, pp. 75, 2007.

Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pp. 735–746, New York, NY, USA, 2010. ACM.

Ossi Räisä, Antti Koskela, and Antti Honkela. Differentially private Hamiltonian Monte Carlo, 2021. URL https://arxiv.org/abs/2106.09376.

Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

Mrinank Sharma, Michael Hutchinson, Siddharth Swaroop, Antti Honkela, and Richard E. Turner. Differentially private federated variational inference, 2019. URL https://arxiv.org/abs/1911.10563.

Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *Proc. GlobalSIP 2013*, pp. 245–248, 2013. doi: 10.1109/GlobalSIP.2013.6736861. URL https://doi.org/10.1109/GlobalSIP.2013.6736861.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Margarita Vinaroz and Mijung Park. Differentially private stochastic expectation propagation (DP-SEP), 2021. URL https://arxiv.org/abs/2111.13219.

J Wainwright, M I Jordan, Martin J Wainwright, and Michael I Jordan. Graphical models, exponential families, and variational inference. *Mach. Learn.*, 1:1–2, 2008.

Matt P. Wand. Fully simplified multivariate Normal updates in non-conjugate variational message passing. *Journal of Machine Learning Research*, 15:1351–1369, 2014.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2493–2502, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/wangg15.html.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony Q S Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.*, 15:3454–3469, 2020.

John Winn, Christopher M. Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

Sinan Yıldırım and Beyza Ermiş. Exact MCMC with differentially private moves. *Statistics and Computing*, 29(5):947–963, sep 2019. ISSN 0960-3174. doi: 10.1007/s11222-018-9847-x. URL https://doi.org/10.1007/s11222-018-9847-x.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, 2019. doi: 10.1109/TPAMI.2018.2889774.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *J. Mach. Learn. Res.*, 14:3321–3363, 2013.

Zuhe Zhang, Benjamin Rubinstein, and Christos Dimitrakakis. On the differential privacy of Bayesian inference. In *Proc. Conf. AAAI Artif. Intell. 2016*, 2016.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf.

A Appendix: theorems and proofs

This Appendix contains all proofs and some additional theorems omitted from the main text. For easy of reading, we state all the theorems before the proofs.

Privacy via local optimisation: DP optimisation

Theorem A.1. *Running DP-SGD for client-level optimisation in Algorithm 1, using subsampling fraction* $q_{sample} \in (0,1]$ *on the local data level for* T *local optimisation steps in total, with* S *global updates interleaved with the local steps, the resulting model is* (ε,δ)*-DP, with* $\delta \in (0,1)$ *s.t.* $\varepsilon = \mathbb{O}(\delta, q_{sample}, T, \mathcal{G}_\sigma)$.

Proof. Standard DP-SGD theory (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016) ensures that the local optimised approximation is DP after a given number of local optimisation steps by a given client m, when on every local step we clip each per-example gradient to enforce a known ℓ_2-norm bound C, add iid Gaussian noise with standard deviation σ scaled by C to each dimension, and the privacy amplification by subsampling factor q_{sample} for the sampling without replacement function (see Definition 4) is calculated from the fraction of local data utilised on each step. Since the global update does not access the sensitive data, the global model is DP w.r.t. data held by client m after the global update due to post-processing guarantees. Hence, when accounting for DP for client m, the total number of compositions is T, regardless of the number of global updates. The total privacy is therefore (ε, δ), when δ is such that $\varepsilon = \mathbb{O}(\delta, q_{sample}, T, \mathcal{G}_\sigma)$. □

With Theorem A.1, DP is guaranteed independently by each client w.r.t. their own data, and hence the global model will have DP guarantees w.r.t. any clients' data via parallel composition, i.e., the global model is $(\epsilon_{max}, \delta_{max})$-DP w.r.t. any single training data sample with $\epsilon_{max} = \max\{\epsilon_1, \ldots, \epsilon_M\}, \delta_{max} = \max\{\delta_1, \ldots, \delta_M\}$, where ϵ_m, δ_m are the parameters used by client m. In all the experiments in this paper, we use a common (ϵ, δ) budget shared by all the clients, and sampling without replacement on the local data level as the subsampling method.

Properties of non-DP local averaging The following properties are the local averaging counterparts to the regular PVI properties shown by Ashman et al. (2022). We write $n_{m,k}$ for the number of samples in shard k at client m after the initial partitioning, so $\sum_{k=1}^{N_m} n_{m,k} = n_m$.

***Property* A.2** (cf. Property 2.1 of Ashman et al. 2022)**.** *Maximizing the local ELBO*

$$\mathcal{L}_{m,k}^{(s)}(q(\theta)) := \int d\theta q(\theta) \log \frac{[p(x_{m,k}|\theta)]^{N_m} q^{(s-1)}(\theta)}{q(\theta) t_m^{(s-1)}(\theta)}$$

is equivalent to the KL optimization

$$q^{(s)}(\theta) = \arg\min_q D_{\mathrm{KL}}(q(\theta) \| \hat{p}_{m,k}^{(s)}(\theta)),$$

where $\hat{p}_{m,k}^{(s)}(\theta) = \frac{1}{\hat{Z}_{m,k}^{(s)}} p(\theta) \prod_{j \neq m} t_j^{(s-1)}(\theta) \cdot [p(x_{m,k}|\theta)]^{N_m}$ *is the tempered tilted distribution before global update* s *for local shard* k *at client* m*.*

Proof. The proof is identical to the one in (Ashman et al., 2022, A.1) when we replace the full local likelihood $p(x_m|\theta)$ by the tempered likelihood $[p(x_{m,k}|\theta)]^{N_m}$ for shard k. □

***Property* A.3** (cf. Property 2.2 of Ashman et al. 2022)**.** *Let* $q^*(\theta) = p(\theta) \prod_{j=1}^M t_j^*(\theta)$ *be a fixed point for local averaging,* $\mathcal{L}_{m,k}^*(q(\theta)) = \int d\theta q(\theta) \log \frac{q^*(\theta)[p(x_{m,k}|\theta)]^{N_m}}{q(\theta) t_m^*(\theta)}$ *local ELBO at the fixed point w.r.t. shard* k *at client* m*, and* $\mathcal{L}(q(\theta)) = \int d\theta q(\theta) \log \frac{p(\theta)p(x|\theta)}{q(\theta)}$ *global ELBO. Then*

1. $\sum_{j=1}^M \frac{1}{N_j} \sum_{k=1}^{N_j} \mathcal{L}_{j,k}^*(q^*(\theta)) = \mathcal{L}(q^*(\theta)) - \log Z_{q^*}$.

2. *If* $q^*(\theta) = \arg\max_q \mathcal{L}_{j,k}(q(\theta))$ *for all* j, k*, then* $q^*(\theta) = \arg\max_q \mathcal{L}(q(\theta))$*.*

Proof. 1. Directly from the definition we have

$$\begin{aligned}
\mathcal{L}(q^*(\theta)) - \log Z_{q^*} &= \int d\theta q^*(\theta) \log \frac{p(\theta)p(x|\theta)}{q^*(\theta)Z_{q^*}} && \text{(A.1)}\\
&= \int d\theta q^*(\theta)[\log p(x|\theta) - \log \frac{p(\theta)\prod_{j=1}^{M} t_j^*(\theta)}{p(\theta)}] && \text{(A.2)}\\
&= \sum_{j=1}^{M} \int d\theta q^*(\theta)[\log \prod_{k=1}^{N_j} p(x_{j,k}|\theta) - \log t_j^*(\theta)] && \text{(A.3)}\\
&= \sum_{j=1}^{M} \int d\theta q^*(\theta)[\frac{1}{N_j} N_j \sum_{k=1}^{N_j} \log p(x_{j,k}|\theta) - \log \frac{q^*(\theta)t_j^*(\theta)}{q^*(\theta)}] && \text{(A.4)}\\
&= \sum_{j=1}^{M} \frac{1}{N_j} \sum_{k=1}^{N_j} \int d\theta q^*(\theta)[\log \frac{q^*(\theta)[p(x_{j,k}|\theta)]^{N_j}}{q^*(\theta)t_j^*(\theta)}] && \text{(A.5)}\\
&= \sum_{j=1}^{M} \frac{1}{N_j} \sum_{k=1}^{N_j} \mathcal{L}_{j,k}^*(q^*(\theta)). && \text{(A.6)}
\end{aligned}$$

Assume $q^*(\theta) = \arg\max_q \mathcal{L}_{j,k}(q(\theta))$ for all j, k. Then $q^*(\theta) = \arg\max_q \mathcal{L}^*_{j,k}(q(\theta))$ for all j, k implying that $\frac{d}{d\lambda_q}\mathcal{L}^*_{j,k}(q^*(\theta)) = 0$ for all j, k. Furthermore, since $q^*(\theta)$ is a maximizer, the Hessian $\frac{d^2}{d\lambda_q d\lambda_q^T}\mathcal{L}^*_{j,k}(q^*(\theta))$ is negative definite for all j, k. Looking at the first part of the proof we can write

$$\frac{d}{d\lambda_q}\mathcal{L}(q^*(\theta)) = \sum_{j=1}^{M} \frac{1}{N_j} \sum_{k=1}^{N_j} \frac{d}{d\lambda_q}\mathcal{L}^*_{j,k}(q^*(\theta)) = 0, \text{ and} \tag{A.7}$$

$$\frac{d^2}{d\lambda_q d\lambda_q^T}\mathcal{L}(q^*(\theta)) = \sum_{j=1}^{M} \frac{1}{N_j} \sum_{k=1}^{N_j} \frac{d^2}{d\lambda_q d\lambda_q^T}\mathcal{L}^*_{j,k}(q^*(\theta)). \tag{A.8}$$

From Equation A.7 we see that $q^*(\theta)$ is a fixed point of $\mathcal{L}(q(\theta))$, and since the Hessian in Equation A.8 can be expressed by summing negative definite matrices and multiplying them by positive numbers, the resulting Hessian is also negative definite, and hence $q^*(\theta)$ maximizes the global ELBO $\mathcal{L}(q(\theta))$.

□

***Property* A.4** (cf. Property 3.2 of Ashman et al. 2022)**.** *Assume the prior and approximate likelihood factors are in the unnormalized exponential family $t_m(\theta) = t_m(\theta; \lambda_m) = \exp(\lambda_m^T T(\theta))$, so the variational distribution is in the normalized exponential family $q(\theta) = \exp(\lambda_q^T T(\theta) - A(\lambda_q))$. Then a stationary point of the local ELBO at global update s for the kth local model at client m, $\frac{d\mathcal{L}^{(s)}_{m,k}(q_k(\theta))}{d\lambda_q} = 0$, implies*

$$\lambda^{(s)}_{m,k} = N_m \frac{d}{d\mu_{q^{(s-1)}}}\mathbb{E}_{q^{(s-1)}}[\log p(x_{m,k}|\theta)].$$

In addition, a stationary point for all N_m local models' ELBO implies

$$\lambda^{(s)}_m = \frac{1}{N_m}\sum_{k=1}^{N_m} \lambda^{(s)}_{m,k} = \frac{d}{d\mu_{q^{(s-1)}}}\mathbb{E}_{q^{(s-1)}}[\log p(x_m|\theta)],$$

which matches the regular PVI fixed point equation.

Proof. Writing $\mathcal{L}^{(s)}_{m,k}(q_k(\theta)) = \int d\theta q_k(\theta) \log \frac{[p(x_{m,k}|\theta)]^{N_m} q^{(s-1)}(\theta)}{q_k(\theta) t_m^{(s-1)}(\theta)}$ and noting that all N_m local models are started in parallel from the same point (so $\mu_{q_k^{(s-1)}} = \mu_{q^{(s-1)}}, q_k^{(s-1)} = q^{(s-1)}\ \forall k$), then following the proof in

(Ashman et al., 2022, Supplement A.4) with minor changes establishes the first claim:

$$\lambda_{m,k}^{(s)} = N_m \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_{m,k}|\theta)].$$

Looking now at the average of local parameters we have

$$\lambda_m^{(s)} = \frac{1}{N_m} \sum_{k=1}^{N_m} \lambda_{m,k}^{(s)} \tag{A.9}$$

$$= \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\sum_{k=1}^{N_m} \log p(x_{m,k}|\theta)] \tag{A.10}$$

$$= \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_m|\theta)], \tag{A.11}$$

where the last equality assumes that the data are conditionally independent given the model parameters. □

***Property* A.5** (cf. Property 5 of Bui et al. 2018)**.** *Under the assumptions of Prop. A.4, using local averaging with parallel global updates result in identical dynamics for $q(\theta)$, given by the following equation, regardless of the partition of the data employed:*

$$\lambda_q^{(s)} = \lambda_0 + \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x|\theta)] = \lambda_0 + \sum_{i=1}^{\sum_j n_j} \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_i|\theta)],$$

where x_i is the ith data point, and n_j is the number of local samples on client j.

Proof. With M clients doing a parallel update, from Property A.4 we have

$$\lambda_q^{(s)} = \lambda_0 + \sum_{j=1}^{M} \lambda_j^{(s)} \tag{A.12}$$

$$= \lambda_0 + \sum_{j=1}^{M} \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_j|\theta)] \tag{A.13}$$

$$= \lambda_0 + \sum_{j=1}^{M} \sum_{k=1}^{n_j} \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_{m,k}|\theta)], \tag{A.14}$$

where Equation A.14 follows due to data being conditionally independent given the model parameters.

On the other hand, with $M = 1$ Property A.4 reads

$$\lambda_q^{(s)} = \lambda_0 + \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x|\theta)] \tag{A.15}$$

$$= \lambda_0 + \sum_{i=1}^{n} \frac{d}{d\mu_{q^{(s-1)}}} \mathbb{E}_{q^{(s-1)}}[\log p(x_i|\theta)], \tag{A.16}$$

which matches *Equation A*.14, since $n = \sum_j^M n_j$ for any M.

□

DP with local averaging

Theorem A.6. *Assume the change in the model parameters* $\|\lambda^*_{m_k} - \lambda^{(s-1)}\|_2 \le C, k = 1, \dots, N_m$ *for some known constant* C*, where* $\lambda^*_{m_k}$ *is a proposed solution to Equation 4.1, and* $\lambda^{(s-1)}$ *is the vector of common initial values. Then releasing* $\Delta\hat{\lambda}^*_m$ *is* (ε, δ)*-DP, with* $\delta \in (0,1)$ *s.t.* $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, 1, \mathcal{G}_\sigma)$*, when*

$$\Delta\hat{\lambda}^*_m = \frac{1}{N_m}\big[\sum_{k=1}^{N_m}\big(\lambda^*_{m_k} - \lambda^{(s-1)}\big) + \xi\big], \tag{A.17}$$

where $\xi \sim \mathcal{N}(0, \sigma^2 \cdot I)$*.*

Proof. Considering neighbouring datasets as in the DP definition 2, denoted by m, m', only one of the local models is affected by the differing element, w.l.o.g. assume it is $\lambda^*_{m_1}$. For the difference in the sum query between neighbouring datasets we therefore immediately have

$$\begin{aligned} \|\sum_{k=1}^{N_m}\big(\lambda^*_{m_k} - \lambda^{(s-1)}\big) - \sum_{k=1}^{N_m}\big(\lambda^*_{m'_k} - \lambda^{(s-1)}\big)\|_2 &= \|\lambda^*_{m_1} - \lambda^{(s-1)} - \lambda^*_{m'_1} + \lambda^{(s-1)}\|_2 && \text{(A.18)}\\ &\le \|\lambda^*_{m_1} - \lambda^{(s-1)}\|_2 + \|\lambda^*_{m'_1} - \lambda^{(s-1)}\|_2 && \text{(A.19)}\\ &\le 2C. && \text{(A.20)} \end{aligned}$$

The sum query therefore corresponds to a single call to the Gaussian mechanism with sensitivity $2C$ and noise standard deviation σ, and since DP guarantees are not affected by post-processing such as taking average, the claim follows. □

Corollary A.7. *A composition of* S *global updates with local averaging using a norm bound* C *for clipping is* (ε, δ)*-DP, with* $\delta \in (0,1)$ *s.t.* $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, S, \mathcal{G}_\sigma)$*.*

Proof. Since each global update is DP by Theorem A.6 when we enforce the norm bound by clipping, the result follows immediately by composing over the global updates. □

Theorem A.8. *With local averaging, The DP noise standard deviation can be scaled as* $\mathcal{O}(\frac{1}{N_m})$*, where* N_m *is the number of local partitions. Therefore, the effect of DP noise will vanish on the local factor level when the local dataset size and the number of local partitions grow.*

Proof. Rewriting Equation A.17 as

$$\Delta\hat{\lambda}^*_m = \frac{1}{N_m}\big(\sum_{k=1}^{N_m}\lambda^*_{m_k} - \lambda^{(s-1)}\big) + \frac{\xi}{N_m}, \tag{A.21}$$

where $\xi \sim \mathcal{N}(0, \sigma^2 \cdot I)$.

Letting the number of local partitions grow, we immediately have

$$\lim_{N_m \to \infty} \Delta\hat{\lambda}^*_m = \lim_{N_m \to \infty} \Delta\lambda^*_m, \tag{A.22}$$

where $\Delta\lambda^*_m$ is the corresponding non-DP average. □

Theorem A.9. *Assume the effective prior* $p_{\setminus j}(\eta)$*, and the likelihood* $p(x_j|\eta), j \in \{1, \dots, M\}$ *are in a conjugate exponential family, where* η *are the natural parameters. Then the number of partitions used in local averaging does not affect the non-DP posterior.*

Proof. To avoid notational clutter, we drop the client index j in the rest of this proof, and simply write, e.g., x for the local data x_j and $p(\eta)$ for the effective prior $p_{\setminus j}(\eta)$.

Due to the conjugacy we can write the effective prior and the likelihood as

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - A(\eta)), \tag{A.23}$$

$$p(\eta|\tau_0, n_0) = H(\tau_0, n_0)\exp(\tau_0^T \eta - n_0 A(\eta)), \tag{A.24}$$

where T are the sufficient statistics, A is the log-partition function, τ_0, n_0 are the effective prior parameters, and h, H are some suitable functions determined by the exponential family.

The local posterior given a vector x of n iid observations is in the same exponential family:

$$p(\eta|x, \tau_0, n_0) \propto \exp\big((\tau_0 + \sum_{i=1}^{n} T(x_i))^T \eta - (n_0 + n)A(\eta)\big),$$

so the changes in parameters when updating from prior to posterior are

$$\tau_0 \to \tau_0 + \sum_{i=1}^{n} T(x_i) \tag{A.25}$$

$$n_0 \to n_0 + n. \tag{A.26}$$

Now partitioning the data into N shards, each with $n_k = \frac{n}{N}$ samples, together with the tempered (cold) likelihood $p(x|\eta)^N$ results in a posterior

$$p_{shard}(\eta|x_k, \tau_0, n_0) \propto \exp((\tau_0 + \sum_{k_i} NT(x_{k_i}))^T \eta - (n_0 + Nn_k)A(\eta)),$$

so the updates for shard k are

$$\tau_0 \to \tau_0 + \sum_{k_i} NT(x_{k_i}) \tag{A.27}$$

$$n_0 \to n_0 + Nn_k. \tag{A.28}$$

Averaging over the posterior parameters corresponding to the N local data shards we have updates

$$\tau_0 \to \frac{1}{N}\sum_{k=1}^{N}\big(\tau_0 + \sum_{k_i} NT(x_{k_i})\big) \tag{A.29}$$

$$= \tau_0 + \sum_{k=1}^{N}\sum_{k_i} T(x_{k_i}) \tag{A.30}$$

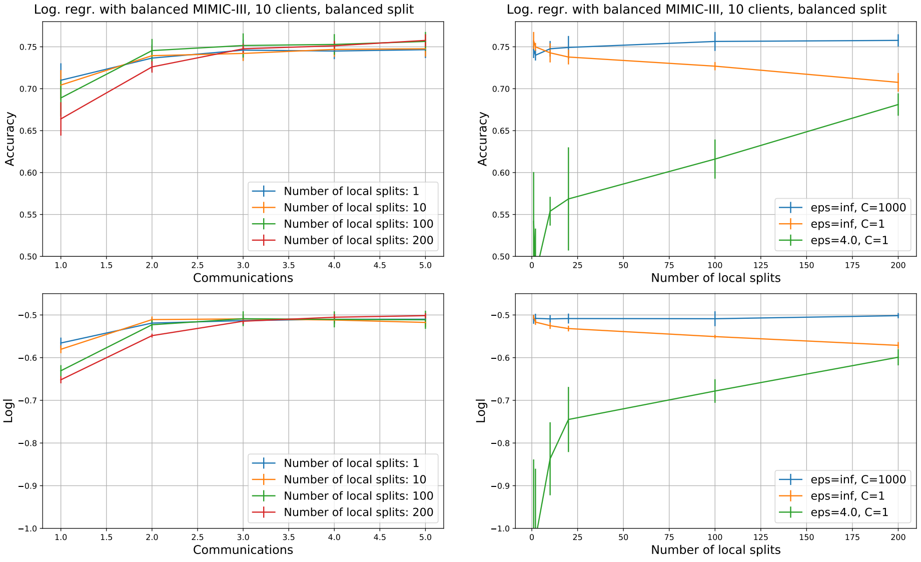
$$= \tau_0 + \sum_{i=1}^{n} T(x_i), \text{ and} \tag{A.31}$$

$$n_0 \to \frac{1}{N}\sum_{k=1}^{N}\big(n_0 + Nn_k\big) \tag{A.32}$$

$$= n_0 + n, \tag{A.33}$$

which match the expressions for the regular local posterior using full local data given in Equation A.25 and Equation A.26. □

Figure 5 shows the effects of changing the number of local data shards using a logistic regression model with and without DP. As discussed in Section 4.2.1, when the assumptions of Theorem A.9 are not satisfied, we would expect that increasing the number of local data partitions can lead to slower convergence due to increased estimator variance. This can be seen in Figure 5 a). In contrast, under privacy constraints increasing the number of local partitions can lead to improved performance, since adding local partitioning



(a)

(b)

Figure 5: Logistic regression, balanced MIMIC-III data with 10 clients: mean over 5 seeds with SEM, balanced split. a) Without DP, increasing the number of local partitions can lead to slower convergence, b) non-DP with clipping norm C, and $(4, 10^{-5})$-DP: wihout privacy increasing the number of local partitions does not help (non-DP with clipping $C = 1000$), or even hurts performance (non-DP with clipping $C = 1$), while with DP, increasing the number of local partitions mitigates the effect of DP noise.

can mitigate the DP noise effect, as seem in Figure 5 b). Note that increasing the number of local partitions can also increase the bias due to clipping, especially with tight clipping bound. In this experiment, we use same fixed hyperparameters in all runs: number of global updates or communication rounds = 5, number of local steps = 50, learning rate = 10^{-2}, damping = .4.

Theorem A.10. *Using local averaging with M clients and a shared number of local partitions $N_j = N \ \forall j$ assume the clients have access to a trusted aggregator. Then for any given privacy parameters ε, δ, the noise standard deviation added by a single client can be scaled as $\mathcal{O}(\frac{1}{\sqrt{M}})$ while guaranteeing the same privacy level.*

Proof. Let $\eta \sim \mathcal{N}(0, \sigma^2 \cdot I)$, and denote by σ_0 the noise standard deviation that locally guarantees the required DP level for every client with some known norm bound C (possibly due to clipping), and assign equal noise shares over clients. The message for a synchronous global update s is

$$\prod_{j=1}^{M} \Delta t_m^{(s)} = \sum_{j=1}^{M} \Big(\frac{1}{N} [\sum_{k=1}^{N} (\lambda_{j_k}^* - \lambda^{(s-1)}) + \eta] \Big) \tag{A.34}$$

$$= \frac{1}{N} \Big(\sum_{j=1}^{M} \sum_{k=1}^{N} (\lambda_{j_k}^* - \lambda^{(s-1)}) + \sum_{j=1}^{M} \eta \Big). \tag{A.35}$$

To match the target local noise standard deviation with the aggregated noise standard deviation we need

$$\sum_{j=1}^{M} \sigma^2 \geq \sigma_0^2 \tag{A.36}$$

$$\Leftrightarrow \sigma \geq \frac{\sigma_0}{\sqrt{M}}. \tag{A.37}$$

Setting σ to match the lower bound, we see that the total noise magnitude on the global approximation level in Equation A.35 does not change with M. □

Looking at Theorem A.10, when the global noise level is constant, adding a client to the protocol will reduce the relative effect of the noise in Equation A.35 if it increases the non-noise part in the sum. On the other hand, on global convergence we would have

$$\sum_{j=1}^{M} \big(\frac{1}{N} \sum_{k=1}^{N} \lambda_{j_k}^{*} - \lambda^{(s-1)} \big) = 0,$$

so an update near a global optimum will be mostly noise.

When applying Theorem A.10, DP is guaranteed jointly on the global model level, while the local approximations have less noise than required for the stated privacy level (although they might still have some valid DP guarantees). In contrast, when each client guarantees DP independently via Theorem A.6, the global model will be $(\epsilon_{max}, \delta_{max})$-DP w.r.t. any single training data sample by parallel composition with $\epsilon_{max} = \max\{\epsilon_1, \dots, \epsilon_M\}, \delta_{max} = \max\{\delta_1, \dots, \delta_M\}$, where ϵ_m, δ_m are the parameters used by client m. In all the experiments in this paper, we use a common (ϵ, δ) budget shared by all the clients.

DP with virtual PVI clients

Theorem A.11. *Assume the change in the model parameters $\|\lambda_{m_k}^{*} - \lambda^{(s-1)}\|_2 \leq C, k = 1, \dots, N_m$ for some known constant C, where $\lambda_{m_k}^{*}$ is a proposed solution to Equation 4.8, and $\lambda^{(s-1)}$ is the vector of common initial values. Then releasing $\Delta\tilde{\lambda}_m^{*}$ is (ε, δ)-DP, with $\delta \in (0,1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, 1, \mathcal{G}_\sigma)$, when*

$$\Delta\tilde{\lambda}_m^{*} = \sum_{k=1}^{N_m} \big(\lambda_{m_k}^{*} - \lambda^{(s-1)}\big) + \eta, \tag{A.38}$$

where $\eta \sim \mathcal{N}(0, \sigma^2 \cdot I)$.

Proof. Almost the same as the proof of Theorem A.6. □

Corollary A.12. *A composition of S global updates with virtual PVI clients using a norm bound C for clipping is (ε, δ)-DP, with $\delta \in (0,1)$ s.t. $\varepsilon = \mathbb{O}(\delta, q_{sample} = 1, S, \mathcal{G}_\sigma)$.*

Proof. Since each global update is DP by Theorem A.11 when we enforce the norm bound by clipping, the result follows immediately by composing over the global updates. □

Theorem A.13. *Assume there are M real clients adding virtual clients, and access to a trusted aggregator. Then for any given privacy parameters ε, δ, the noise standard deviation added by a single client can be scaled as $\mathcal{O}(\frac{1}{\sqrt{M}})$ while guaranteeing the same privacy level.*

Proof. Similar to the proof for Theorem A.10 with obvious modifications. □

As with local averaging, Theorem A.13 gives joint DP guarantees on the global model level. In contrast, when each client guarantees DP independently with Theorem A.11, the global model will be $(\epsilon_{max}, \delta_{max})$-DP w.r.t. any single training data sample by parallel composition with $\epsilon_{max} = \max\{\epsilon_1, \dots, \epsilon_M\}, \delta_{max} = \max\{\delta_1, \dots, \delta_M\}$, where ϵ_m, δ_m are the parameters used by client m. And again, in all the experiments we use a common (ϵ, δ) budget shared by all the clients.

B Appendix: experimental details

This appendix contains details of the experimental settings omitted from Section 5.

With Adult data, we first combine the training and test sets, and then randomly split the whole data with 80% for training and 20% for validation. With MIMIC-III data, we first preprocessing the data for the in-hospital mortality prediction task as detailed by Harutyunyan et al. (2019).[7]. Since the preprocessed data is very unbalanced and leaves little room for showing the differences between the methods (a constant prediction can reach close to 90% accuracy while a non-DP prediction can do some percentage points better), we first re-balance the data by keeping only as many majority label samples as there are in the minority class. This leaves 5594 samples, which are then randomly split into training and validation sets, giving a total of 4475 samples of training data to be divided over all the clients.

We divide the data between M clients using the following scheme[8]: half of the clients are small and the other half large, with data sizes given by

$$n_{small} = \lfloor \frac{n}{M}(1-\rho) \rfloor, \quad n_{large} = \lfloor \frac{n}{M}(1+\rho) \rfloor,$$

with $\rho \in [0,1]$. $\rho = 0$ gives equal data sizes for everyone while $\rho = 1$ means that the small clients have no data. For creating unbalanced data distributions, denote the fraction of majority class samples by λ. Then the target fraction of majority class samples for the small clients is parameterized by κ:

$$\lambda_{small}^{target} = \lambda + (1-\lambda) \cdot \kappa,$$

where having $\kappa = 1$ means small clients only have majority class labels, and $\kappa = -\frac{\lambda}{1-\lambda}$ implies small clients have only minority class labels. For large clients the labels are divided randomly.

We use the following splits in the experiments:

	ρ	κ	n_{small}	λ_{small}	λ_{large}
balanced	0	0	2442	.76	$\simeq .76$
unbalanced 1	.75	.95	610	.99	$\simeq .73$
unbalanced 2	.7	-3	732	.03	$\simeq .89$

Table 2: Adult data, 10 clients data split.

	ρ	κ	n_{small}	λ_{small}	λ_{large}
balanced	0	0	122	.75	.7-.8
unbalanced 1	.75	.95	30	.97	.67-.78
unbalanced 2	.7	-3	36	.03	.85-.93

Table 3: Adult data, 200 clients data split.

	ρ	κ	n_{small}	λ_{small}	λ_{large}
balanced	0	0	447	.5	$\simeq .5$
unbalanced 1	.75	.95	111	.97	.41-.44
unbalanced 2	.7	-.5	134	.25	.51-.58

Table 4: Balanced MIMIC-III data, 10 clients data split.

We use Adam (Kingma & Ba, 2014) to optimise all objective functions. In general, depending e.g. on the update schedule, even the non-DP PVI can diverge (see Ashman et al. 2022). We found that DP-PVI using

[7] The code for preprocessing is available from `https://github.com/YerevaNN/mimic3-benchmarks`

[8] The data splitting scheme was originally introduced by Sharma et al. (2019) in a workshop paper that combines DP with PVI. The current paper is otherwise completely novel and not based on the earlier workshop version.

any of our approaches is more prone to diverge than non-DP PVI, while DP optimisation is more stable than local averaging or virtual PVI clients. To improve model stability, we use some damping in all the experiments. When damping with a factor $\rho \in (0, 1]$, at global update s the model parameters $\lambda^{(s)}$ are set to

$$(1 - \rho) \cdot \lambda^{(s-1)} + \rho \cdot \lambda^{(s)}.$$

We use grid search to optimise all hyperparameters in terms of predictive accuracy and model log-likelihood using 1 random seed, and then run 5 independent random seeds using the best hyperparameters from the 1 seed runs. The reported results with 5 random seeds are the best results in terms of log-likelihood for each model. With BNNs using local averaging or virtual PVI clients some seeds diverged when using the hyperparameters optimised using a single seed. These seeds were rerun with the same hyperparameter settings to produce 5 full runs. This might give the methods some extra advantage in the comparison, but since they still do not work too well, we can surmise that the methods are not well suited for the task.

To approximate the posterior predictive distributions we use a Monte Carlo estimate with 100 samples:

$$p(y_*|x_*, y, x) \simeq \frac{1}{100} \sum_{i_{MC}=1}^{100} p(y_*|x_*, \theta_{i_{MC}}), \quad \theta_{i_{MC}} \sim q(\theta).$$