Bayesian Differential Privacy for Machine Learning

Aleksei Triastcyn and **Boi Faltings**
Artificial Intelligence Lab
EPFL
Lausanne, Switzerland
{aleksei.triastcyn, boi.faltings}@epfl.ch

Abstract

We propose *Bayesian differential privacy*, a relaxation of differential privacy that provides sharper privacy guarantees for similarly distributed data, especially in difficult scenarios, such as deep learning. We derive a general privacy accounting method for iterative learning algorithms under Bayesian differential privacy and show that it is a generalisation of the well-known moments accountant. Our experiments show significant improvements in privacy guarantees for typical cases in deep learning datasets, such as MNIST and CIFAR-10, in some cases bringing the privacy budget ε from 8 down to 0.5.

1 Introduction

Machine learning (ML) and data analytics present countless opportunities for companies, governments and individuals to benefit from the accumulated data. At the same time, their ability to capture fine levels of detail potentially compromises privacy of data providers. Recent research [18, 39, 23] suggests that even in a black-box setting it is possible to argue about the presence of individual records in the training set or recover certain features of these records.

To tackle this problem a number of solutions has been proposed. They vary in how privacy is achieved and to what extent data is protected. In this work, we consider a notion that is viewed by many researchers as the gold standard—*differential privacy* (DP) [17]. Initially, DP algorithms focused on sanitising simple statistics, such as mean, median, etc., using a technique known as output perturbation. In recent years, the field made a lot of progress towards the goal of privacy-preserving machine learning, through works on objective perturbation [12], stochastic gradient descent with DP updates [40], to more complex and practical techniques [1, 35, 36, 30]. For a more detailed overview of related work, we refer the reader to Appendix A.

Despite significant advances, differentially private machine learning still suffers from two major problems: (a) utility loss due to excessive amounts of noise added during training and (b) difficulty in interpreting the privacy parameters ε and δ. In many cases where the first problem appears to be solved, it is actually being hidden by the second. To illustrate it, we design a motivational example in Section 2 that shows how a seemingly strong privacy guarantee allows for the attacker accuracy to be as high as 99%. Even considering that this guarantee is very pessimistic and holds against a very powerful adversary with any auxiliary information, it can hardly be viewed as a reassurance to a user. Moreover, it provides only the worst-case bound, leaving users to wonder how far is the worst-case from a typical case.

In this paper, we focus on practicality of a privacy guarantee and propose a relaxation of differential privacy that provides more meaningful guarantees for *typical* scenarios on top of the global differential privacy guarantee. We name it *Bayesian differential privacy* (BDP).

The key to our relaxation is our definition of *typical* scenarios. At the core of it lies the observation that machine learning models are designed and tuned for a particular data distribution (for example, an MRI dataset is very unlikely to contain a picture of a car). Moreover, such prior distribution of data is often already available to the attacker. We consider a scenario *typical* when all sensitive data is drawn from the same distribution. While the traditional differential privacy treats all data as equally likely and hides differences by large amounts of noise, Bayesian differential privacy calibrates noise to the data distribution. Thus, for any two adjacent datasets drawn from the same distribution, and given the same privacy mechanism with the same amount of noise, BDP provides tighter expected guarantees.

The idea of using the data randomness in the context of DP is not new by itself [13, 6, 21, 5, 25], and our work could be viewed as a special case of some of these definitions. Nonetheless, to the best of our knowledge, the prior work does not provide efficient and tight privacy accounting in real-world scenarios.

As the data distribution is usually *unknown*, BDP estimates the necessary statistics from data as shown in the following sections. Furthermore, since typical scenarios are determined by data, the participants of the dataset are covered by the BDP guarantee with high probability.

To accompany the notion of Bayesian differential privacy (Section 3.1), we provide its theoretical analysis and the privacy accounting framework (Section 3.2). The latter considers the privacy loss random variable and employs principled

tools from probability theory to find concentration bounds on it. It provides a clean derivation of privacy accounting in general (Sections 3.2 and 3.3), as well as in the special case of subsampled Gaussian noise mechanism. Moreover, we show that it is a generalisation of the well-known moments accountant (MA) [1] (Section 3.4).

Since our privacy accounting relies on data distribution samples, a natural concern would be that the data not present in the dataset are not taken into account, and thus, are not protected. However, this is not the case, because our finite sample estimator is specifically designed to address this issue (see Section 3.3).

Our contributions in this paper are the following:

- we propose a relaxation of DP, called Bayesian differential privacy, that allows to provide more practical privacy guarantees in a wide range of scenarios;
- we derive a clean, principled method for privacy accounting in learning that generalises the moments accountant;
- we experimentally demonstrate advantages of our method (Section 4), including the state-of-the-art privacy bounds in deep learning applications (Section 4.2).

2 Motivation

Before we proceed, we find it important to motivate the research on alternative definitions of privacy. The primary reason for this is that the complexity of the concept of differential privacy often leads to misunderstanding or overestimation of the guarantees it provides.

Consider the state-of-the-art differentially private machine learning models [1, 36]. In order to come close to the non-private accuracy (say within 10% of it), all of the reported models stretch their privacy budget to $\varepsilon > 2$ (for a reasonably low δ), while in many cases it goes up to $\varepsilon > 5$. In real-world applications, it can even be larger than 10[1]. These numbers seem small, and thus, may often be overlooked. But let us present an alternative interpretation.

What we are interested in is the change in the posterior distribution of the attacker after they see the private model compared to prior [33, 10]. Let us consider the stronger, pure DP for simplicity and assume the following specific example. The datasets D, D' consist of income values for residents of a small town. There is one individual x' whose income is orders of magnitude higher than the rest, and whose residency in the town is what the attacker wishes to infer. The attacker observes the mean income w sanitised by a differentially private mechanism with $\varepsilon = \varepsilon_0$. It is easy to see, that if the individual is not present in the dataset, the probability of w being above a certain threshold is extremely small. On the contrary, if x' is present, this probability is higher (say it is equal to r). The attacker takes a Bayesian approach, computes the likelihood of the observed value under each of the two assumptions and the corresponding posteriors given a flat prior. The attacker then concludes that the individual is present in the dataset and is a resident.

[1]https://www.wired.com/story/apple-differential-privacy-shortcomings/

By the above expression, r can only be e^{ε_0} times larger than the corresponding probability without x'. However, if the $re^{-\varepsilon_0}$ is small enough, then the probability $P(A)$ of the attacker's guess being correct is as high as $\frac{r}{r+re^{-\varepsilon_0}}$ or, equivalently,

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

To put it in perspective, for a DP algorithm with $\varepsilon = 2$, the upper bound on the accuracy of this attack is as high as 88%. For $\varepsilon = 5$, it is 99.33%. For $\varepsilon = 10$, 99.995%. Remember that we used an uninformative flat prior, and for a more informed attacker these numbers could be even larger.

Such a guarantee is hardly better than no guarantee at all, because in a more realistic scenario, even without any privacy protection, this high accuracy is not likely to be achieved by the attacker. Considering the generality of the DP formulation, it is not surprising, and to obtain more meaningful guarantees one needs more stringent assumptions. On the other hand, it is beneficial to maintain a looser general guarantee in case assumptions do not hold.

In the next section, we present a relaxation of DP that uses the same privacy mechanism and augments the general DP guarantee with a tighter guarantee for the expected case. In the view of our example: while it is hard to hide the presence of a wealthy individual, the privacy guarantee for the rest of the town residents is likely to be significantly stronger.

3 Bayesian Differential Privacy

In this section, we define *Bayesian differential privacy (BDP)*. We then derive a practical privacy loss accounting method, and discuss its relation to the moments accountant.

3.1 Definition

Let us define Bayesian differential privacy (Definition 1) and *weak* Bayesian differential privacy (Definition 2). The first definition provides a better intuition, connection to concentration inequalities, and is being used for privacy accounting. Unfortunately, it may not be closed under post-processing, and therefore, the actual guarantee provided by BDP is stated in Definition 2 and mimics the (ε, δ)-differential privacy [16]. It is similar to how the moments accountant bounds tails of the privacy loss random variable and converts it to the (ε, δ)-DP guarantee in [1].

Definition 1 (Bayesian Differential Privacy)**.** *A randomised function $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$, range $\mathcal{R}$, and outcome $w = \mathcal{A}(\cdot)$, satisfies $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$,* differing in a single data point $x' \sim \mu(x)$*, the following holds:*

$$\Pr[L_{\mathcal{A}}(w, D, D') \geq \varepsilon_\mu] \leq \delta_\mu, \tag{2}$$

where probability is taken over the randomness of the outcome w and the additional example x'.

Here, $L_{\mathcal{A}}(w, D, D')$ is the privacy loss defined as

$$L_{\mathcal{A}}(w, D, D') = \log \frac{p(w|D)}{p(w|D')}, \tag{3}$$

where $p(w|D)$, $p(w|D')$ are private outcome distributions for corresponding datasets. For brevity, we often omit parameters and denote the privacy loss simply by L.

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

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

Definition 2 (Weak Bayesian Differential Privacy)**.** *A randomised function $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\varepsilon_\mu, \delta_\mu)$-weak Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$, differing in a single data point $x' \sim \mu(x)$, and for any set of outcomes $\mathcal{S}$ the following holds:*

$$\Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \leq e^{\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta_\mu. \tag{4}$$

Proposition 1. *$(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy implies $(\varepsilon_\mu, \delta_\mu)$-weak Bayesian differential privacy.*

Bayesian DP mirrors some basic properties of the classic DP, such as composition, resilience to post-processing and group privacy. We state and proof these properties, as well as the above proposition, in Appendix B.

Remark. While Definition 1 does not specify the distribution of any point in the dataset other than the additional example x', it is natural to assume that all examples in the dataset are drawn from the same distribution $\mu(x)$. This holds in many real-world applications, including applications evaluated in this paper, and it allows using dataset samples instead of requiring knowing the true distribution.

Remark. We also assume that all data points are exchangeable [3], i.e. any permutation of data points has the same joint probability. It enables tighter accounting for iterative applications of the privacy mechanism (see Section 3.2), and is naturally satisfied in the considered scenarios.

3.2 Privacy Accounting

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

Observe that Eq. 2 is a typical concentration bound inequality, which are well studied in probability theory. One of the most common examples of such bounds is Markov's inequality. In its extended form, it states the following:

$$\Pr[|L| \geq \varepsilon_\mu] \leq \frac{\mathbb{E}[\varphi(|L|)]}{\varphi(\varepsilon_\mu)}, \tag{5}$$

where $\varphi(\cdot)$ is a monotonically increasing non-negative function. It is immediately evident that it provides a relation between ε_μ and δ_μ (i.e. $\delta_\mu = \mathbb{E}[\varphi(|L|)]/\varphi(\varepsilon_\mu)$), and in order to determine them we need to choose φ and compute the expectation $\mathbb{E}[\varphi(|L(w, D, D')|)]$. Note that $L(w, D, D') = -L(w, D', D)$, and since the inequality has to hold for any pair of D, D', we can use L instead of $|L|$.

We use the Chernoff bound that can be obtained by choosing $\varphi(L) = e^{\lambda L}$. It is widely known because of its tightness, and although not explicitly stated, it is also used by Abadi et al. [1]. The inequality in this case transforms to

$$\Pr[L \geq \varepsilon_\mu] \leq \frac{\mathbb{E}[e^{\lambda L}]}{e^{\lambda \varepsilon_\mu}}. \tag{6}$$

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

$$\begin{aligned}\mathbb{E}[e^{\lambda L}] &= \mathbb{E}\left[e^{\lambda \log \frac{p(w|D)}{p(w|D')}}\right] \\ &= \mathbb{E}\left[\left(\frac{p(w|D)}{p(w|D')}\right)^{\lambda}\right].\end{aligned} \tag{7}$$

If the expectation is taken only over the outcome randomness, this expression is the function of Rényi divergence between $p(w|D)$ and $p(w|D')$, and following this path yields re-derivation of Rényi differential privacy [33]. However, by also taking the expectation over additional examples $x' \sim \mu(x)$, we can further tighten this bound.

By the law of total expectation,

$$\mathbb{E}\left[\left(\frac{p(w|D)}{p(w|D')}\right)^{\lambda}\right] = \mathbb{E}_x\left[\mathbb{E}_w\left[\left(\frac{p(w|D)}{p(w|D')}\right)^{\lambda} \middle| x'\right]\right], \tag{8}$$

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

Combining Eq. 7 and 8 and plugging it in Eq. 6, we get

$$\Pr[L \geq \varepsilon_\mu] \leq \mathbb{E}_x\left[e^{\lambda \mathcal{D}_{\lambda+1}[p(w|D)\|p(w|D')] - \lambda \varepsilon_\mu}\right]. \tag{9}$$

This expression determines how to compute ε_μ for a fixed δ_μ (or vice versa) for one invocation of the privacy mechanism. However, to accommodate the iterative nature of learning, we need to deal with the composition of multiple applications of the mechanism. We already determined that our privacy notion is naively composable, but in order to achieve better bounds we need a tighter composition theorem. Note also that due to computing expectation over data in Eq. 8 and Eq. 9, we assume exchangeability [3] to obtain results in practice. This assumption is weaker than independence and is natural in many applications.

Theorem 1 (Advanced Composition)**.** *Let a learning algorithm run for T iterations. Denote by $w^{(1)} \dots w^{(T)}$ a sequence of private learning outcomes at iterations $1, \dots, T$, and $L^{(1:T)}$ the corresponding total privacy loss. Then,*

$$\mathbb{E}\left[e^{\lambda L^{(1:T)}}\right] = \prod_{t=1}^{T} \mathbb{E}_x\left[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)}\right],$$

where $p_t = p(w^{(t)}|w^{(t-1)}, D)$, $q_t = p(w^{(t)}|w^{(t-1)}, D')$.

Proof. See Appendix C. □

We denote the logarithm of the quantity inside the product in Theorem 1 as $c_t(\lambda)$ and call it the *privacy cost* of the iteration t:

$$c_t(\lambda) = \log \mathbb{E}_x \left[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)} \right] \tag{10}$$

The privacy cost of the whole learning process is then a sum of the costs of each iteration. We can now relate ε and δ parameters of BDP through the privacy cost.

Theorem 2. *Let the algorithm produce a sequence of private learning outcomes $w^{(1)} \dots w^{(T)}$ using a known probability distribution $p(w^{(t)}|w^{(t-1)}, D)$. Then, for a fixed ε_μ:*

$$\log \delta_\mu \leq \sum_{t=1}^{T} c_t(\lambda) - \lambda \varepsilon_\mu.$$

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

$$\varepsilon_\mu \leq \frac{1}{\lambda} \sum_{t=1}^{T} c_t(\lambda) - \frac{1}{\lambda} \log \delta_\mu.$$

Theorems 1, 2 and Corollary 1 immediately provide us with an efficient privacy accounting algorithm. During training, we compute the privacy cost $c_t(\lambda)$ for each iteration t, accumulate it, and then use to compute $\varepsilon_\mu, \delta_\mu$ pair. This process is ideologically close to that of the moment accountant, but accumulates a different quantity (note the change from the privacy loss random variable to Rényi divergence). We further explore this connection in Section 3.4.

The link to Rényi divergence is a great advantage for applicability of this framework: as long as the outcome distribution $p(w|D)$ has a known analytic expression for Rényi divergence [20, 42], it can be used within a privacy mechanism, and our accountant can track its privacy loss.

For the popular subsampled Gaussian mechanism [1], we can demonstrate the following.

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

$$c_t(\lambda) = \max\{c_t^L(\lambda), c_t^R(\lambda)\},$$

where

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

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

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

Proof. See Appendix E. □

3.3 Privacy Cost Estimator

Computing $c_t(\lambda)$ precisely requires access to the data distribution $\mu(x)$, which is unrealistic. Therefore, we need an estimator for $\mathbb{E}[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)}]$.

Typically, having access to the distribution samples, one would use the law of large numbers and approximate the expectation with the sample mean. This estimator is unbiased and converges with the growing number of samples. However, these are not the properties we are looking for. The most important property of the estimator in our context is that it *does not underestimate* $\mathbb{E}[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)}]$, because the bound (Eq. 6) would not hold for this estimate otherwise.

We employ the Bayesian view of the parameter estimation problem [34] and design an estimator with this single property: given a fixed probability γ, it returns the value that overestimates the true expectation with probability $1-\gamma$. We then incorporate the estimator uncertainty γ in δ_μ.

Binary Case Let us demonstrate the process of constructing the expectation estimator with the aforementioned property on a simple binary example. This technique is based on [34] and it translates directly to other classes of distributions with minor adjustments. We also address a natural concern about taking into account the data not present in the dataset by providing a specific example.

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

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

$$f(x_i|\rho) = \rho^{x_i}(1-\rho)^{1-x_i}, \tag{11}$$

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

$$f(x_1, \dots, x_N|\rho) = \rho^{N_1}(1-\rho)^{N_0}, \tag{12}$$

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

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

$$f(\rho|x_1, \dots, x_N) = \frac{\Gamma(N_0 + N_1 + 2)}{\Gamma(N_0 + 1)\Gamma(N_1 + 1)} \rho^{N_1}(1-\rho)^{N_0}, \tag{13}$$

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

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

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

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

To demonstrate the resilience of this estimator to unseen data, consider the following simple example. Let the true expectation be 0.01, and let the data consist of 100 zeros, and no ones. A typical "frequentist" mean estimator would confidently output 0. However, our estimator would never output 0, unless the confidence is set to 0. When the confidence is set to 1 ($\gamma = 0$), the output is 1, which is the most pessimistic estimate. Finally, the output $\hat{\rho} = \rho = 0.01$ will be assigned the failure probability $\gamma = 0.99^{101} \approx 0.36$, which is the probability of not drawing a single 1 in 101 draws.

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

Continuous Case For applications evaluated in this paper, we are primarily concerned with continuous case. Thus, let us define the following m-sample estimator of $c_t(\lambda)$ for continuous distributions with existing mean and variance:

$$\hat{c}_t(\lambda) = \log\left[M(t) + \frac{F^{-1}(1-\gamma, m-1)}{\sqrt{m-1}} S(t)\right], \quad (14)$$

where $M(t)$ and $S(t)$ are the sample mean and the sample standard deviation of $e^{\lambda \hat{D}^{(t)}_{\lambda+1}}$, $F^{-1}(1-\gamma, m-1)$ is the inverse of the Student's t-distribution CDF at $1-\gamma$ with $m-1$ degrees of freedom, and

$$\begin{aligned}
&\hat{D}^{(t)}_{\lambda+1} = \max\left\{D_{\lambda+1}(\hat{p}_t \| \hat{q}_t),\ D_{\lambda+1}(\hat{q}_t \| \hat{p}_t)\right\},\\
&\hat{p}_t = p(w^{(t)} \mid w^{(t-1)}, B^{(t)}),\\
&\hat{q}_t = p(w^{(t)} \mid w^{(t-1)}, B^{(t)} \setminus \{x_i\}).
\end{aligned}$$

Since in many cases learning is performed on mini-batches, we can similarly compute Rényi divergence on batches $B^{(t)}$.

Theorem 4. *Estimator $\hat{c}_t(\lambda)$ overestimates $c_t(\lambda)$ with probability $1-\gamma$. That is,*

$$\Pr\left[\hat{c}_t(\lambda) < c_t(\lambda)\right] \le \gamma.$$

Proof. The proof is similar to the above binary example. See more details in Appendix D. □

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

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

$$\begin{aligned}
&\Pr[L_{\mathcal{A}}(w^{(t)}, D, D') \ge \varepsilon_\mu]\\
&\quad = \Pr\left[L_{\mathcal{A}}(w^{(t)}, D, D') \ge \varepsilon_\mu, \hat{c}_t(\lambda) \ge c_t(\lambda)\right]\\
&\qquad + \Pr\left[L_{\mathcal{A}}(w^{(t)}, D, D') \ge \varepsilon_\mu, \hat{c}_t(\lambda) < c_t(\lambda)\right].
\end{aligned}$$

When $\hat{c}_t(\lambda) \ge c_t(\lambda)$, using the Chernoff inequality, the first summand is bounded by $\beta = \exp(\sum_{t=1}^{T} \hat{c}_t(\lambda) - \lambda\varepsilon_\mu)$. Whenever $\hat{c}_t(\lambda) < c_t(\lambda)$,

$$\begin{aligned}
&\Pr[L_{\mathcal{A}}(w^{(t)}, D, D') \ge \varepsilon_\mu, \hat{c}_t(\lambda) < c_t(\lambda)]\\
&\quad \le \Pr[\hat{c}_t(\lambda) < c_t(\lambda)]\\
&\quad \le \gamma.
\end{aligned}$$

Therefore, the true δ_μ is bounded by $\beta + \gamma$, and despite the incomplete data, we can claim that the mechanism is $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private, where $\delta_\mu = \beta + \gamma$.

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

3.4 Discussion

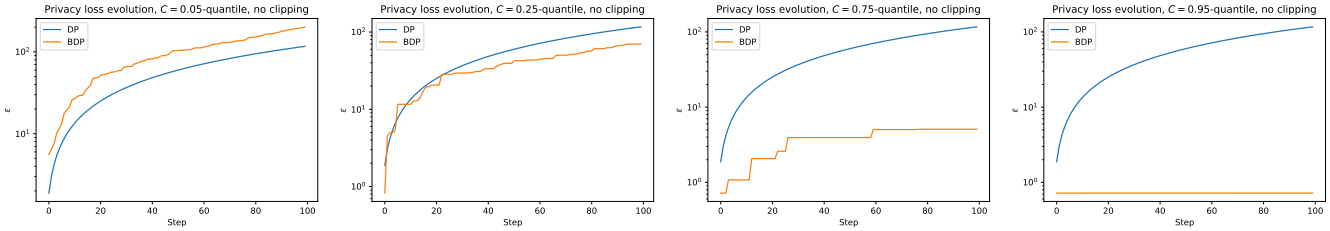
Relation to Moments Accountant and RDP As mentioned in Section 3.2, removing the distribution requirement on D, D' and further simplifying Eq. 9, we can recover the relation between Rényi differential privacy and (ε, δ)-DP.

At the same time, our accounting technique closely resembles the moments accountant. In fact, we can show that the moments accountant is a special case of Theorem 3. Ignoring the data distribution information and substituting expectation by $\max_{D,D'}$ yields the substitution of $\|g_t - g'_t\|$ for C in Theorem 3, where C is the sensitivity (or clipping threshold), which turns out to be the exact moments accountant bound. In addition, there are some extra benefits, such as avoiding numerical integration when using $\lambda \in \mathbb{N}$ due to connection to Binomial distribution, which improves numerical stability and computational efficiency.

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

Privacy of $\hat{c}_t(\lambda)$ Due to computing $\hat{c}_t(\lambda)$ from data our privacy guarantee ε becomes data-dependent and may theoretically leak sensitive information by itself. There are multiple ways to approach this problem.

One way would be to observe that the privacy leakage is tied to the error of the estimator: an adversary who has access to the prior data distribution would be able to compute the true $c_t(\lambda)$ with arbitrary precision, and thus, the only leaking information about the actual data is the error between $\hat{c}_t(\lambda)$ and $c_t(\lambda)$. On the other hand, it may be possible to express the distribution of the sample mean and variance



(a) 0.05-quantile of $\|\nabla f\|$. (b) 0.25-quantile of $\|\nabla f\|$. (c) 0.75-quantile of $\|\nabla f\|$. (d) 0.95-quantile of $\|\nabla f\|$.

Figure 1: Evolution of ε and ε_μ over multiple steps of the Gaussian noise mechanism with $\sigma = C$ for DP (with clipping) and BDP (without clipping). Sub-captions indicate the noise variance relative to the gradient norms distribution.

of the privacy loss through the true mean and bound the density ratios for two neighbouring datasets.

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

Finally, one should consider the fact that the information from many high-dimensional vectors gets first compressed down to their pairwise distances, which are not as informative in high-dimensional spaces (i.e. the curse of dimensionality), and then down to one number. We believe that at this rate of compression very little knowledge can be gained by an attacker in practice. In Section 4.2, we examine pairwise gradient distances of the points within the training set and outside, and do not find any evidence of privacy leakage. However, obtaining strict theoretical bounds, potentially in one of the ways discussed above, is more desirable and is an important future research direction.

4 Evaluation

This experimental section comprises two parts. First, we examine how well Bayesian DP composes over multiple steps. We use the Bayesian accountant and compare to the state-of-the-art DP results obtained by the moments accountant [1]. Second, we consider the context of machine learning. In particular, we use the differentially private stochastic gradient descent (DP-SGD), a well known privacy-preserving learning technique broadly used in combination with the moments accountant, to train neural networks on classic image classification tasks MNIST [28] and CIFAR10 [27]. We then compare the accuracy and privacy guarantees obtained under BDP and under DP. We also perform experiments with variational inference on Abalone [45] and Adult [26] datasets.

As stated above, DP and BDP can use the same privacy mechanism and be accounted in parallel to ensure the DP guarantees hold if BDP assumptions fail. Thus, all comparisons in this section should be viewed in the following way: the reported BDP guarantee would apply to *typical* data (i.e. data drawn from the same distribution as the dataset); the reported DP guarantee would apply to all other data; their difference is the advantage for typical data we gain by using Bayesian DP. In some experiments we use smaller noise variance for BDP in order to speed up training, meaning that the reported BDP guarantees will further improve if noise variance is increased to DP levels. Finally, it is worth reiterating that the interpretation of δ_μ of BDP is different from δ of DP, as discussed in Sections 3.1 and 3.3.

Table 1: Estimated privacy bounds $\varepsilon, \varepsilon_\mu$ for $\delta = \delta_\mu = 10^{-5}$ for MNIST, CIFAR10, Abalone and Adult datasets.

Dataset	**Accuracy** Baseline	Private	ε DP	ε_μ BDP	$P(A)$ DP	BDP
MNIST	99%	96%	2.18	**0.62**	89.8%	**65.0%**
CIFAR10	86%	73%	8.0	**0.51**	99.9%	**62.5%**
Abalone	77%	76%	7.6	**0.5**	99.9%	**62.3%**
Adult	81%	81%	0.5	**0.16**	62.3%	**54.0%**

4.1 Composition

First, we study the growth rate of the privacy loss over a number of mechanism invocations. This experiment is carried out using synthetic gradients drawn from the Weibull distribution with the shape parameter < 1 to imitate a more difficult case of heavy-tailed gradient distributions. We do not clip gradients for BDP in order to show the raw effect of the signal-to-noise ratio on the privacy loss behaviour.

In Figure 1, we plot ε and ε_μ as a function of steps for different levels of noise. Naturally, as the noise standard deviation gets closer to the expected gradients norm, the growth rate of the privacy loss decreases dramatically. Even when the noise is at the 0.25-quantile, the Bayesian accountant matches the moments accountant. It is worth noting, that DP behaves the same in all these experiments because the gradients get clipped at the noise level C. Introducing clipping for BDP yields the behaviour of Figure 1d, as we demonstrate in the next section on real data.

4.2 Learning

We now consider the application to privacy-preserving deep learning. Our setting closely mimics that of [1] to enable a direct comparison with the moments accountant and DP. We use a version of DP-SGD [1] that has been extensively applied to build differentially private machine learning models. The idea of DP-SGD is to clip the gradient norm to

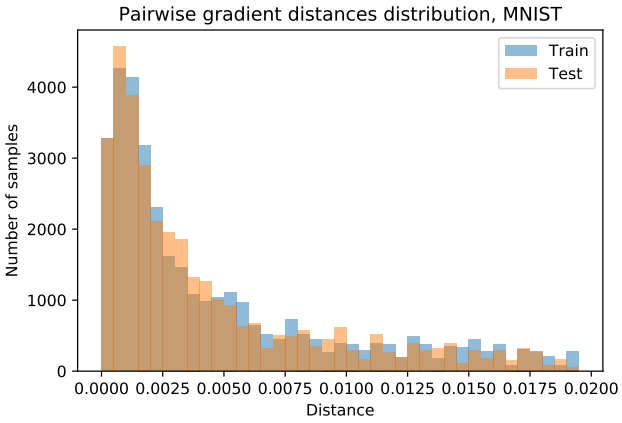


Figure 2: Pairwise gradient distances for MNIST.

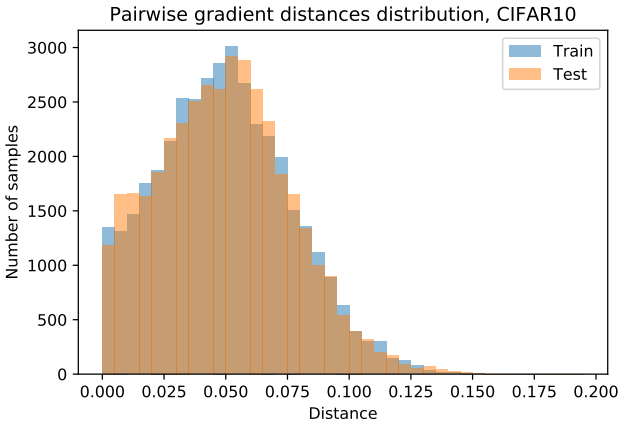


Figure 3: Pairwise gradient distances for CIFAR10.

some constant C (ensuring bounded sensitivity) and then add Gaussian noise with variance $C^2\sigma^2$ at every iteration of SGD. For Abalone and Adult datasets, we use variational inference and a setting similar to [24]. See more setting details in Appendix F.

Using the information about gradient distribution allows the BDP models to reach the same accuracy at ε_μ much lower than ε. On MNIST, we manage to reduce it from 2.18 to 0.62. For CIFAR10, from 8.0 to 0.51. See details in Table 1. Alternatively, Bayesian differential privacy allows to add less noise to achieve ε_μ comparable to ε. Because of this, the models reach the same test accuracy much faster. For example, our model reaches 96% accuracy within 20 epochs for MNIST, while DP model requires hundreds of epochs to avoid ε blowing up. These results also confirm our assumption that the actual disagreement between gradient directions is much smaller than their norms, and therefore, requires less noise to hide. To make our results more transparent, we include in Table 1 the potential attack success probability $P(A)$ computed using Eq. 1. In this interpretation, the benefits of using Bayesian differential privacy become even more apparent.

An important aspect of BDP, discussed in Section 3.4, is the potential privacy leakage of the privacy cost estimator. Since at the moment we do not have a rigorous bound on the amount of information it leaks, we conduct the following experiment. After training the model (to ensure it contains as much information about data as possible), we compute the gradient pairwise distances over train and test sets. We then plot the histograms of these distances to inspect any divergence that would distinguish the data that was used in training. Note that this is more information than what is available to an adversary, who only observes ε_μ.

As it turns out, these distributions are nearly identical (see Figures 2 and 3), and we do not observe any correlation with the fact of the presence of data in the training set. For example, the sample mean of the test set can be both somewhat higher or lower than that of the train set. We also run the t-test for equality of means and Levene's test for equality of variances, obtaining p-values well over the 0.05 threshold, suggesting that the difference of the means and the variances of these distributions is not statistically significant and the equality hypothesis cannot be rejected.

5 Conclusion

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

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

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

References

[1] 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*, 308–318. ACM.

[2] Abowd, J. M.; Schneider, M. J.; and Vilhuber, L. 2013. Differential privacy applications to bayesian and linear mixed model estimation. *Journal of Privacy and Confidentiality* 5(1):4.

[3] Aldous, D. J. 1985. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*. Springer. 1–198.

[4] Bassily, R., and Freund, Y. 2016. Typical stability. *arXiv preprint arXiv:1604.03336*.

[5] Bassily, R.; Groce, A.; Katz, J.; and Smith, A. 2013. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 439–448. IEEE.

[6] Bhaskar, R.; Bhowmick, A.; Goyal, V.; Laxman, S.; and Thakurta, A. 2011. Noiseless database privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*, 215–232. Springer.

[7] Blum, A.; Ligett, K.; and Roth, A. 2013. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)* 60(2):12.

[8] Bun, M., and Steinke, T. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, 635–658. Springer.

[9] Bun, M.; Dwork, C.; Rothblum, G. N.; and Steinke, T. 2018. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 74–86. ACM.

[10] Bun, M. 2017. A teaser for differential privacy.

[11] Charest, A.-S., and Hou, Y. 2017. On the meaning and limits of empirical differential privacy. *Journal of Privacy and Confidentiality* 7(3):3.

[12] Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12(Mar):1069–1109.

[13] Duan, Y. 2009. Privacy without noise. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 1517–1520. ACM.

[14] Dwork, C., and Rothblum, G. N. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

[15] Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

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

[17] Dwork, C. 2006. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, 1–12. Venice, Italy: Springer Verlag.

[18] Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333. ACM.

[19] Geumlek, J.; Song, S.; and Chaudhuri, K. 2017. Renyi differential privacy mechanisms for posterior sampling. In *Advances in Neural Information Processing Systems*, 5289–5298.

[20] Gil, M.; Alajaji, F.; and Linder, T. 2013. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences* 249:124–131.

[21] Hall, R.; Rinaldo, A.; and Wasserman, L. 2011. Random differential privacy. *arXiv preprint arXiv:1112.2680*.

[22] He, X.; Machanavajjhala, A.; and Ding, B. 2014. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 1447–1458. ACM.

[23] Hitaj, B.; Ateniese, G.; and Pérez-Cruz, F. 2017. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 603–618. ACM.

[24] Jälkö, J.; Dikmen, O.; and Honkela, A. 2016. Differentially private variational inference for non-conjugate models. *arXiv preprint arXiv:1610.08749*.

[25] Kifer, D., and Machanavajjhala, A. 2014. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)* 39(1):3.

[26] Kohavi, R. 1996. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. Citeseer.

[27] Krizhevsky, A. 2009. Learning multiple layers of features from tiny images.

[28] LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

[29] Leung, S., and Lui, E. 2012. Bayesian mechanism design with efficiency, privacy, and approximate truthfulness. In *International Workshop on Internet and Network Economics*, 58–71. Springer.

[30] McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning differentially private recurrent language models.

[31] Mir, D. J. 2012. Information-theoretic foundations of differential privacy. In *International Symposium on Foundations and Practice of Security*, 374–381. Springer.

[32] Mironov, I.; Pandey, O.; Reingold, O.; and Vadhan, S. 2009. Computational differential privacy. In *Annual International Cryptology Conference*, 126–142. Springer.

[33] Mironov, I. 2017. Renyi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, 263–275. IEEE.

[34] Oliphant, T. E. 2006. A bayesian perspective on estimating mean, variance, and standard-deviation from data.

[35] Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

[36] Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, Ú. 2018. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*.

[37] Schneider, M. J., and Abowd, J. M. 2015. A new method for protecting interrelated time series with bayesian prior distributions and synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 178(4):963–975.

[38] Shokri, R., and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1310–1321. ACM.

[39] Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 3–18. IEEE.

[40] Song, S.; Chaudhuri, K.; and Sarwate, A. D. 2013. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, 245–248. IEEE.

[41] Triastcyn, A., and Faltings, B. 2019. Generating artificial data for private deep learning. In *Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies, AAAI Spring Symposium Series*, 33–40.

[42] Van Erven, T., and Harremos, P. 2014. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory* 60(7):3797–3820.

[43] Wang, Y.-X.; Lei, J.; and Fienberg, S. E. 2016. On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms. In *International Conference on Privacy in Statistical Databases*, 121–134. Springer.

[44] Wang, W.; Ying, L.; and Zhang, J. 2016. On the relation between identifiability, differential privacy, and mutual-information privacy. *IEEE Transactions on Information Theory* 62(9):5018–5029.

[45] Waugh, S. G. 1995. *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. Ph.D. Dissertation, University of Tasmania.

[46] Yang, B.; Sato, I.; and Nakagawa, H. 2015. Bayesian differential privacy on correlated data. In *Proceedings of the 2015 ACM SIGMOD international conference on Management of Data*, 747–762. ACM.

A Related Work

As machine learning applications become more and more common, various vulnerabilities and attacks on ML models get discovered, based on both passive (for example, model inversion [18] and membership inference [39]) and active adversaries (e.g. [23]), raising the need for developing matching defences.

Differential privacy [17, 15] is one of the strongest privacy standards that can be employed to protect ML models from these and other attacks. Since pure ε-DP is hard to achieve in many realistic learning settings, a notion of approximate (ε, δ)-DP is used across-the-board in machine learning. It is often achieved as a result of applying the Gaussian noise mechanism [16]. Several alternative notions and relaxations of DP have also been proposed, such as computational DP [32], mutual-information privacy [31, 44], different versions of concentrated DP (CDP [14], zCDP [8], tCDP [9]), and Rényi DP (RDP) [33]. Some other relaxations [2, 37, 11, 43, 41] tip the balance even further in favour of applicability at the cost of weaker guarantees, considering the average-case instead of the worst-case or limiting the guarantee to a given dataset. Unlike these relaxations, our notion is not limited to a particular dataset, but rather a particular distribution of data (e.g. emails, MRI images, etc.), which is a much weaker assumption.

For a long time, approximate DP remained unachievable in more popular deep learning scenarios. Some earlier attempts [38] led to prohibitively high bounds on ε [1, 35] that were later shown to be ineffective against attacks [23]. A major step in the direction of bringing privacy loss values down to more practical magnitudes was done by Abadi et al. [1] with the introduction of the *moments accountant*, currently a state-of-the-art method for keeping track of the privacy loss during training. Followed by improvements in differentially private training techniques [35, 36], it allowed to achieve single-digit DP guarantees ($\varepsilon < 10$) for classic supervised learning benchmarks, such as MNIST, SVHN, and CIFAR.

In general, an important aspect of a privacy notion is composability, accountability, and interpretability. Apart from sharp bounds, the moments accountant is attractive because it operates within the classic notion of (ε, δ)-DP. Some of the alternative notions of DP, such as [33, 9], also provide tight composition theorems, along with some other advantages, but to the best of our knowledge, they are not broadly used in practice compared to traditional DP (although there are some examples [19]). One of the possible reasons for that is interpretability: parameters of (α, ε)-RDP or (μ, τ)-CDP

are hard to interpret. While it may be difficult to quantify the actual guarantee provided by specific values of ε, δ of the traditional DP, it is still advantageous that they have a clearer probabilistic interpretation.

Our privacy notion can be related to some of the past work on DP relaxations. In Section 3.4, we discuss its connection to RDP [33] and the moments accountant [1]. Similarly, there is a link to concentrated DP definitions.

A number of previous relaxations considered a similar idea of limiting the scope of protected data or using the data generating distribution, either through imposing a set of data evolution scenarios [25], policies [22], distributions [7, 6], or families of distributions [5, 4]. Some of these definitions (e.g. [7]) may require more noise, because they are stronger than DP in the sense that datasets can differ in more than one data point. This is not the case with our definition: like DP, it considers adjacent datasets *differing in a single data point*. The major problem of such definitions, however, is that in real-world scenarios it is not feasible to exactly define distributions or families of distributions that generate data. And even if this problem is solved by restricting the query functions to enable the usage of the central limit theorem (e.g. [6, 13]), these guarantees would only hold asymptotically and may require prohibitively large batch sizes. While Bayesian DP can be seen as a special case of some of the above definitions, the crucial difference with the prior work is that our additional assumptions allow the Bayesian accounting (Sections 3.2, 3.3) to provide guarantees w.h.p. with finite number of samples from data distributions, and hence, allow a broad range of real-world applications.

Finally, there are other approaches that use the data distribution information in one way or another, and coincidentally share the same [46] or similar [29] names. Yet, similarly to the methods discussed above, their assumptions (e.g. the bound on the minimum probability of a datapoint) and implementation requirements (e.g. potentially constructing correlation matrices for millions of data samples) make practical applications difficult. Perhaps the most similar to our approach is the random differential privacy [21], however, the authors only propose a basic composition theorem, which is not tight enough, and computing the probabilities over all dataset examples would not be practical in many realistic machine learning scenarios.

B Proof of Propositions

This appendix contains the basic properties of Bayesian differential privacy and related proofs. Let us begin with restating and proving Proposition 1.

Proposition 1. *$(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy implies $(\varepsilon_\mu, \delta_\mu)$-weak Bayesian differential privacy.*

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

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

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] = \int \Pr[\mathcal{A}(D) \in \mathcal{S}, x'] \, dx' \tag{15}$$

$$= \int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x'), x'] \tag{16}$$

$$+ \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{17}$$

$$= \int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}^c(x') | x'] \mu(x') \tag{18}$$

$$+ \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{19}$$

$$\leq \int e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S} \cap \mathcal{F}^c(x') | x'] \mu(x') \tag{20}$$

$$+ \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{21}$$

$$\leq \int e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S}, x'] \tag{22}$$

$$+ \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{23}$$

$$\leq e^\varepsilon \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta_\mu, \tag{24}$$

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

$$\int \Pr[\mathcal{A}(D) \in \mathcal{S} \cap \mathcal{F}(x'), x'] \, dx' \tag{25}$$

$$\leq \int \Pr[\mathcal{A}(D) \in \mathcal{F}(x'), x'] \, dx' \tag{26}$$

$$= \int \mu(x') \Pr[\mathcal{A}(D) \in \mathcal{F}(x') \mid x'] \, dx' \tag{27}$$

$$= \int \mu(x') \int_{w \in \mathcal{F}(x')} p_{\mathcal{A}}(w|D, x') \, dw \, dx' \tag{28}$$

$$= \mathbb{E}_{x'} \left[\mathbb{E}_w \left[\mathbb{1}\{L > \varepsilon\} \right] \right] \tag{29}$$

$$\leq \delta_\mu \tag{30}$$

□

Proposition 2 (Post-processing)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for any arbitrary randomised data-independent mapping $f : \mathcal{R} \to \mathcal{R}'$, $f(\mathcal{A}(D))$ is $(\varepsilon_\mu, \delta_\mu)$-weak Bayesian differentially private.*

Proof. By Proposition 1, $(\varepsilon_\mu, \delta_\mu)$-BDP implies

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

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

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

$$\Pr\left[f(\mathcal{A}(D)) \in \mathcal{T}\right] = \Pr\left[\mathcal{A}(D) \in \mathcal{S}\right] \tag{32}$$

$$\leq e^{\varepsilon_\mu} \Pr\left[\mathcal{A}(D') \in \mathcal{S}\right] + \delta_\mu, \tag{33}$$

$$= e^{\varepsilon_\mu} \Pr\left[f(\mathcal{A}(D')) \in \mathcal{T}\right] + \delta_\mu \tag{34}$$

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

Proposition 3 (Basic composition)**.** *Let $\mathcal{A}_i : \mathcal{D} \to \mathcal{R}_i$, $\forall i = 1..k$, be a sequence of $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithms. Then their combination, defined as $\mathcal{A}_{1:k} : \mathcal{D} \to \mathcal{R}_1 \times \ldots \times \mathcal{R}_k$, is $(k\varepsilon_\mu, k\delta_\mu)$-Bayesian differentially private.*

Proof. Let us denote $L = \log \frac{p(w_1,\ldots,w_k|D)}{p(w_1,\ldots,w_k|D')}$.

Also, let $L_i = \log \frac{p(w_i|D,w_{i-1},\ldots,w_1)}{p(w_i|D',w_{i-1},\ldots,w_1)}$. Then,

$$\Pr\left[L \geq k\varepsilon_\mu\right] = \Pr\left[\sum_{i=1}^{k} L_i \geq k\varepsilon_\mu\right] \tag{35}$$

$$\leq \sum_{i=1}^{k} \Pr[L_i \geq \varepsilon_\mu] \tag{36}$$

$$\leq \sum_{i=1}^{k} \delta_\mu \tag{37}$$

$$\leq k\delta_\mu \tag{38}$$

□

Proposition 4 (Group privacy)**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be a $(\varepsilon_\mu, \delta_\mu)$-Bayesian differentially private algorithm. Then for all pairs of datasets $D, D' \in \mathcal{D}$, differing in k data points $x_1, \ldots, x_k$ s.t. $x_i \sim \mu(x)$ for $i = 1..k$, $\mathcal{A}(D)$ is $(k\varepsilon_\mu, k\delta_\mu)$-Bayesian differentially private.*

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

$$\frac{p(w|D)}{p(w|D')} = \frac{p(w|D^0)p(w|D^1)\ldots p(w|D^{k-1})}{p(w|D^1)p(w|D^2)\ldots p(w|D^k)} \tag{39}$$

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

Finally, applying the definition of $(\varepsilon_\mu, \delta_\mu)$-Bayesian differential privacy,

$$\Pr\left[L \geq k\varepsilon_\mu\right] = \Pr\left[\sum_{i=1}^{k} L_i \geq k\varepsilon_\mu\right] \tag{40}$$

$$\leq \sum_{i=1}^{k} \Pr[L_i \geq \varepsilon_\mu] \tag{41}$$

$$\leq k\delta_\mu \tag{42}$$

□

C Proof of Theorem 1

Let us restate the theorem:

Theorem 1 (Advanced Composition)**.** *Let a learning algorithm run for T iterations. Denote by $w^{(1)} \ldots w^{(T)}$ a sequence of private learning outcomes at iterations $1, \ldots, T$, and $L^{(1:T)}$ the corresponding total privacy loss. Then,*

$$\mathbb{E}\left[e^{\lambda L^{(1:T)}}\right] = \prod_{t=1}^{T} \mathbb{E}_x\left[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)}\right],$$

where $p_t = p(w^{(t)}|w^{(t-1)}, D)$, $q_t = p(w^{(t)}|w^{(t-1)}, D')$.

Proof. The proof closely follows [1].

First, we can write

$$L^{(1:T)} = \log \frac{p(w^{(1)} \ldots w^{(T)} \mid D)}{p(w^{(1)} \ldots w^{(T)} \mid D')} \tag{43}$$

$$= \log \prod_{t=1}^{T} \frac{p(w^{(t)} \mid w^{(t-1)} \ldots p(w^{(1)}, D)}{p(w^{(t)} \mid w^{(t-1)} \ldots p(w^{(1)}, D')} \tag{44}$$

$$= \log \prod_{t=1}^{T} \frac{p(w^{(t)} \mid w^{(t-1)}, D)}{p(w^{(t)} \mid w^{(t-1)}, D')} \tag{45}$$

$$= \sum_{t=1}^{T} L^{(t)} \tag{46}$$

Then,

$$\mathbb{E}\left[e^{\lambda L^{(1:T)}}\right] = \mathbb{E}\left[e^{\lambda \sum_{t=1}^{T} L^{(t)}}\right] \tag{47}$$

$$= \mathbb{E}\left[e^{\lambda \sum_{t=1}^{T} \log \frac{p(w^{(t)} \mid w^{(t-1)}, D)}{p(w^{(t)} \mid w^{(t-1)}, D')}}\right] \tag{48}$$

$$= \mathbb{E}\left[\prod_{t=1}^{T} e^{\lambda \log \frac{p(w^{(t)} \mid w^{(t-1)}, D)}{p(w^{(t)} \mid w^{(t-1)}, D')}}\right] \tag{49}$$

$$= \prod_{t=1}^{T} \mathbb{E}\left[e^{\lambda \log \frac{p(w^{(t)} \mid w^{(t-1)}, D)}{p(w^{(t)} \mid w^{(t-1)}, D')}}\right] \tag{50}$$

$$= \prod_{t=1}^{T} \mathbb{E}\left[e^{\lambda \mathcal{D}_{\lambda+1}(p_t \| q_t)}\right], \tag{51}$$

where for (50), we additionally assume samples within D (as well as D') are exchangeable, because of taking expectation over data. This assumption is natural in the applications we consider: the order of data points should not matter and the joint probability of any permutation of points should be the same. Finally, (51) is by Eq. 8. □

D Proof of Theorem 4

Let us restate the theorem:

Theorem 4. *Estimator $\hat{c}_t(\lambda)$ overestimates $c_t(\lambda)$ with probability $1 - \gamma$. That is,*

$$\Pr\left[\hat{c}_t(\lambda) < c_t(\lambda)\right] \leq \gamma.$$

Proof. First of all, we can drop the logarithm from our consideration because of its monotonicity.

Now, assuming that samples $e^{\lambda \hat{D}^{(t)}_{\lambda+1}}$ have a common mean and a common variance, and applying the maximum entropy principle in combination with an uninformative (flat) prior, one can show that the quantity $\frac{M(t) - \mathbb{E}\left[e^{\lambda \hat{D}^{(t)}_{\lambda+1}}\right]}{S(t)} \sqrt{m-1}$ follows the Student's t-distribution with $m-1$ degrees of freedom [34].

Finally, we use the inverse of the Student's t CDF to find the value that this random variable would only exceed with probability γ. The result follows by simple arithmetical operations. □

E Proof of Theorem 3

Let us restate the theorem:

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

$$c_t(\lambda) = \max\{c_t^L(\lambda), c_t^R(\lambda)\},$$

where

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

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

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

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

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

$$\mathbb{E}\left[\left(\frac{p(w|D')}{p(w|D)}\right)^{\lambda+1}\right]$$

$$= \mathbb{E}\left[\left(\frac{(1-q)\mathcal{N}(0,\sigma^2) + q\mathcal{N}(d_t,\sigma^2)}{\mathcal{N}(0,\sigma^2)}\right)^{\lambda+1}\right] \quad (52)$$

$$= \mathbb{E}\left[\left((1-q) + q\frac{\mathcal{N}(d_t,\sigma^2)}{\mathcal{N}(0,\sigma^2)}\right)^{\lambda+1}\right] \quad (53)$$

$$= \mathbb{E}\left[\left((1-q) + qe^{\frac{(w-d_t)^2-w^2}{2\sigma^2}}\right)^{\lambda+1}\right] \quad (54)$$

$$= \mathbb{E}\left[\left((1-q) + qe^{\frac{2dw-d_t^2}{2\sigma^2}}\right)^{\lambda+1}\right] \quad (55)$$

$$= \mathbb{E}\left[\sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} e^{\frac{2d_t kw - kd_t^2}{2\sigma^2}}\right] \quad (56)$$

$$= \sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} \mathbb{E}\left[e^{\frac{2d_t kw - kd_t^2}{2\sigma^2}}\right] \quad (57)$$

$$= \sum_{k=0}^{\lambda+1}\binom{\lambda+1}{k} q^k (1-q)^{\lambda+1-k} e^{\frac{k^2-k}{2\sigma^2}d_t^2} \quad (58)$$

$$= \mathbb{E}_{k\sim B(\lambda+1,q)}\left[e^{\frac{k^2-k}{2\sigma^2}\|g_t - g_t'\|^2}\right], \quad (59)$$

Here, in (56) we used the binomial expansion, in (57) the fact that the factors in front of the exponent do not depend on w, and in (58) the property $\mathbb{E}_w\left[\exp(2aw/(2\sigma^2))\right] = \exp(a^2/(2\sigma^2))$ for $w \sim \mathcal{N}(0,\sigma^2)$. Plugging the above in Eq. 10, we get the expression for $c_t^L(\lambda)$.

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

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

$$= \mathbb{E}\left[\left(\frac{\mathcal{N}(0,\sigma^2)}{(1-q)\mathcal{N}(0,\sigma^2) + q\mathcal{N}(d_t,\sigma^2)}\right)^{\lambda}\right] \quad (60)$$

$$\leq \mathbb{E}\left[\left((1-q) + qe^{\frac{d_t^2-2dw}{2\sigma^2}}\right)^{\lambda}\right] \quad (61)$$

$$= \mathbb{E}_{k\sim B(\lambda,q)}\left[e^{\frac{k^2+k}{2\sigma^2}\|g_t - g_t'\|^2}\right] \quad (62)$$

In practice, we haven't found any instance of $\mathcal{D}_{\lambda+1}(p(w|D')\|p(w|D)) < \mathcal{D}_{\lambda+1}(p(w|D)\|p(w|D'))$ when the latter was computed using numerical integration, although it may happen when using this theoretical upper bound. □

F Experimental setting

We train a classifier represented by a neural network (unlike [1], without PCA) on MNIST [28] and on CIFAR10 [27] using DP-SGD. The first dataset contains 60,000 training examples and 10,000 testing images. We use large batch sizes of 1024, clip gradient norms to $C = 1$, and $\sigma = 0.1$. The second dataset consists of 50,000 training images and 10,000 testing images of objects split in 10 classes. For this dataset, we use the batch size of 512, $C = 1$, and $\sigma = 0.7$. We fix $\delta = \delta_\mu = 10^{-5}$ in all experiments. In case of CIFAR10, in order for our results to be comparable to [1], we pre-train convolutional layers of the model on a different dataset and retrain a fully-connected layer in a privacy-preserving way.

Privacy accounting with DP-SGD works in the following way. The non-private learning outcome at each iteration t is the gradient g_t of the loss function w.r.t. the model parameters, the outcome distribution is the Gaussian $\mathcal{N}(g_t, \sigma^2 C^2)$. Before adding noise, the norm of the gradients is clipped to C. For the moments accountant, the privacy loss is calculated using this C and σ. For the Bayesian accountant, either pairs of examples x_i, x_j or pairs of batches are sampled from the dataset at each iteration, and used to compute $\hat{c}_t(\lambda)$. Although clipping gradients is no longer necessary with the Bayesian accountant, it is highly beneficial for incurring lower privacy loss at each iteration and obtaining tighter composition. Moreover, it ensures the classic DP bounds on top of BDP bounds.

We also run evaluation on two binary classification tasks taken from UCI database: Abalone [45] (predicting the age of abalone from physical measurements) and Adult [26] (predicting income based on a person's attributes). In this setting, we compare differentially private variational inference (DPVI-MA [24]) to the variational inference with BDP. The datasets have 4,177 and 48,842 examples with 8 and 14 attributes accordingly. We use the same pre-processing and models as [24].