Computing Tight Differential Privacy Guarantees Using FFT

Koskela, Antti

AISTATS
2020-06

Koskela , A , Jälkö , J & Honkela , A 2020 , Computing Tight Differential Privacy Guarantees Using FFT . in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics . Proceedings of Machine Learning Research , vol. 108 , AISTATS , pp. 2560-2569 , AISTATS 2020: The 23rd International Conference on Artificial Intelligence and Statistics , Palermo , Italy , 03/06/2020 . < http://proceedings.mlr.press/v108/koskela20b.html >

http://hdl.handle.net/10138/318490

unspecified
acceptedVersion

Computing Tight Differential Privacy Guarantees Using FFT

Antti Koskela
University of Helsinki

Joonas Jälkö
Aalto University

Antti Honkela
University of Helsinki

Abstract

Differentially private (DP) machine learning has recently become popular. The privacy loss of DP algorithms is commonly reported using (ε, δ)-DP. In this paper, we propose a numerical accountant for evaluating the privacy loss for algorithms with continuous one dimensional output. This accountant can be applied to the subsampled multidimensional Gaussian mechanism which underlies the popular DP stochastic gradient descent. The proposed method is based on a numerical approximation of an integral formula which gives the exact (ε, δ)-values. The approximation is carried out by discretising the integral and by evaluating discrete convolutions using the fast Fourier transform algorithm. We give both theoretical error bounds and numerical error estimates for the approximation. Experimental comparisons with state-of-the-art techniques demonstrate significant improvements in bound tightness and/or computation time.

1 Introduction

Differential privacy (DP) (Dwork et al., 2006) has clearly been established as the dominant paradigm for privacy-preserving machine learning. Early work on DP machine learning focused on single shot perturbations for convex problems (Chaudhuri et al., 2011), while contemporary research has focused on iterative algorithms such as DP stochastic gradient descent (SGD) (Rajkumar and Agarwal, 2012; Song et al., 2013; Abadi et al., 2016b) .

Evaluating the privacy loss of an iterative algorithm is based on the composition theory of DP. The so-called advanced composition theorem of (Dwork et al., 2010) showed how to trade decreased ε with slightly increased δ in (ϵ, δ)-DP. This was further improved e.g. by Kairouz et al. (2017). The privacy amplification by subsampling (Chaudhuri and Mishra, 2006; Beimel et al., 2013; Bassily et al., 2014; Wang et al., 2015) is another component that has been studied to improve the privacy bounds.

A major breakthrough in obtaining tighter composition bounds came from using the entire privacy loss profile of DP algorithms instead of single (ε, δ)-values. This was first introduced by the moments accountant (Abadi et al., 2016b). The development of Rényi differential privacy (RDP) (Mironov, 2017) allowed tight bounds on the privacy cost of composition, and recently proposed amplification theorems for RDP (Balle et al., 2018; Wang et al., 2019) showed how subsampling affects the privacy cost of RDP. Zhu and Wang (2019) gave tight RDP bounds for the Poisson subsampling method.

Using the recently introduced privacy loss distribution (PLD) formalism (Sommer et al., 2019), we compute tight (ε, δ)-DP bounds on the composition of subsampled Gaussian mechanisms, using discrete Fourier transforms to evaluate the required convolutions. We show numerically that the achieved privacy bounds are tighter than those obtained by Rényi DP compositions and the moments accountant.

Within this computational framework, in addition to the commonly considered Poisson subsampling method, we are also able to compute tight privacy bounds for the subsampling with replacement and subsampling without replacement methods.

2 Differential Privacy

We first recall some basic definitions of differential privacy (Dwork and Roth, 2014). We use the following notation. An input dataset containing N data points is denoted as $X = (x_1, \ldots, x_N) \in \mathcal{X}^N$, where $x_i \in \mathcal{X}$, $1 \leq i \leq N$.

Definition 1. *We say two datasets X and Y are*

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

neighbours in remove/add relation if you get one by removing/adding an element from/to to other and denote it with $\sim_R$. We say X and Y are neighbours in substitute relation if you get one by substituting one element in the other. We denote this with $\sim_S$.

Definition 2. *Let $\varepsilon > 0$ and $\delta \in [0, 1]$. Let $\sim$ define a neighbouring relation. Mechanism $\mathcal{M} : \mathcal{X}^N \to \mathcal{R}$ is $(\varepsilon, \delta, \sim)$-DP if for every $X \sim Y$ and every measurable set $E \subset \mathcal{R}$ it holds that*

$$\Pr(\mathcal{M}(X) \in E) \leq \mathrm{e}^{\varepsilon} \Pr(\mathcal{M}(Y) \in E) + \delta.$$

When the relation is clear from context or irrelevant, we will abbreviate it as (ε, δ)-DP. We call $\mathcal{M}$ tightly $(\varepsilon, \delta, \sim)$-DP, if there does not exist $\delta' < \delta$ such that $\mathcal{M}$ is $(\varepsilon, \delta', \sim)$-DP.

3 Privacy loss distribution

We first introduce the basic tool for obtaining tight privacy bounds: the privacy loss distribution (PLD). The results in this section can be seen as continuous versions of their discrete counterparts given by Meiser and Mohammadi (2018) and Sommer et al. (2019). Detailed proofs are given in Appendix. The results apply for both neighbouring relations $\sim_S$ and $\sim_R$. We focus on mechanisms of the following form.

Definition 3. *Let $\mathcal{M} : \mathcal{X}^N \to \mathbb{R}$ be a randomised mechanism and let $X \sim Y$. Let $f_X(t)$ denote the density function of $\mathcal{M}(X)$ and $f_Y(t)$ the density function of $\mathcal{M}(Y)$. Assume $f_X(t) > 0$ and $f_Y(t) > 0$ for all $t \in \mathbb{R}$. We define the privacy loss function of f_X over f_Y as*

$$\mathcal{L}_{X/Y}(t) = \log \frac{f_X(t)}{f_Y(t)}.$$

The following gives the definition of the privacy loss distribution via its density function for differentiable privacy loss functions. We note that the assumptions hold especially for the subsampled Gaussian mechanism which is considered in Sec. 6.

Definition 4. *Suppose $\mathcal{L}_{X/Y} : \mathbb{R} \to D$, $D \subset \mathbb{R}$ is a continuously differentiable bijective function. The privacy loss distribution (PLD) of $\mathcal{M}(X)$ over $\mathcal{M}(Y)$ is defined to be a random variable which has the density function*

$$\omega_{X/Y}(s) = \begin{cases} f_X\big(\mathcal{L}^{-1}_{X/Y}(s)\big) \frac{\mathrm{d}\mathcal{L}^{-1}_{X/Y}(s)}{\mathrm{d}s}, & s \in \mathcal{L}_{X/Y}(\mathbb{R}), \\ 0, & \text{else.} \end{cases}$$

For the discrete valued versions of the following result, see (Sommer et al., 2019, Lemmas 5 and 10).

Lemma 5. *Assume $(\varepsilon, \infty) \subset \mathcal{L}_{X/Y}(\mathbb{R})$. $\mathcal{M}$ is tightly (ε, δ)-DP for*

$$\delta(\varepsilon) = \max\{\delta_{X/Y}(\varepsilon), \delta_{Y/X}(\varepsilon)\},$$

where

$$\delta_{X/Y}(\varepsilon) = \int_{\varepsilon}^{\infty} (1 - \mathrm{e}^{\varepsilon - s})\, \omega_{X/Y}(s)\, \mathrm{d}s,$$

and similarly for $\delta_{Y/X}(\varepsilon)$.

The PLD formalism is essentially based on Lemma A.2 which states that the mechanism $\mathcal{M}$ is tightly (ε, δ)-DP with

$$\delta(\varepsilon) = \max_{X \sim Y} \Bigg\{ \int_{\mathbb{R}} \max\{f_X(t) - \mathrm{e}^{\varepsilon} f_Y(t), 0\}\, \mathrm{d}t, \int_{\mathbb{R}} \max\{f_Y(t) - \mathrm{e}^{\varepsilon} f_X(t), 0\}\, \mathrm{d}t \Bigg\}.$$

The integral representation of Lemma 5 is then obtained by change of variables. Denoting $s = \mathcal{L}_{X/Y}(t)$, it clearly holds that $f_Y(t) = \mathrm{e}^{-s} f_X(t)$ and

$$\max\{f_X(t) - \mathrm{e}^{\varepsilon} f_Y(t), 0\} = \begin{cases} (1 - \mathrm{e}^{\varepsilon - s}) f_X(t), & \text{if } s > \varepsilon, \\ 0, & \text{otherwise.} \end{cases}$$

By change of variables $t = \mathcal{L}^{-1}_{X/Y}(s)$, we obtain the representation of Lemma 5.

We get the tight privacy guarantee for compositions from a continuous counterpart of the results given by Sommer et al. (2019, Thm. 1).

Theorem 6. *Consider k consecutive applications of a mechanism $\mathcal{M}$. Let $\varepsilon > 0$. The composition is tightly (ε, δ)-DP for δ given by $\delta(\varepsilon) = \max\{\delta_{X/Y}(\varepsilon), \delta_{Y/X}(\varepsilon)\}$, where*

$$\delta_{X/Y}(\varepsilon) = \int_{\varepsilon}^{\infty} (1 - \mathrm{e}^{\varepsilon - s}) \left(\omega_{X/Y} *^k \omega_{X/Y}\right)(s)\, \mathrm{d}s,$$

where $\omega_{X/Y} *^k \omega_{X/Y}$ denotes the k-fold convolution of $\omega_{X/Y}$ (a similar formula holds for $\delta_{Y/X}(\varepsilon)$).

4 The discrete Fourier transform

The discrete Fourier transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ are linear operators $\mathbb{C}^n \to \mathbb{C}^n$ that decompose a complex vector into a Fourier series, or reconstruct it from its Fourier series. Suppose $x = (x_0, \ldots, x_{n-1}), w = (w_0, \ldots, w_{n-1}) \in \mathbb{R}^n$. Then, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are defined as (Stoer and Bulirsch, 2013)

$$(\mathcal{F}x)_k = \sum\nolimits_{j=0}^{n-1} x_j \mathrm{e}^{-i2\pi kj/n},$$
$$(\mathcal{F}^{-1}w)_k = \frac{1}{n} \sum\nolimits_{j=0}^{n-1} w_j \mathrm{e}^{i2\pi kj/n}.$$

Evaluating $\mathcal{F}x$ and $\mathcal{F}^{-1}w$ takes $O(n^2)$ operations, however evaluation via the Fast Fourier Transform

(FFT) (Cooley and Tukey, 1965) reduces the computational cost to $O(n \log n)$.

The convolution theorem (Stockham Jr, 1966) states that for periodic discrete convolutions it holds that

$$\sum_{i=0}^{n-1} v_i w_{k-i} = \mathcal{F}^{-1}(\mathcal{F}v \odot \mathcal{F}w), \tag{4.1}$$

where $\odot$ denotes the elementwise product of vectors and the summation indices are modulo n.

5 Description of the method

We next describe the numerical method for computing tight DP-guarantees for continuous one dimensional distributions.

5.1 Truncation of convolutions

We first approximate the convolutions on a truncated interval $[-L, L]$ as

$$(\omega * \omega)(x) \approx \int_{-L}^{L} \omega(t)\omega(x-t)\,\mathrm{d}t =: (\omega \circledast \omega)(x).$$

To obtain periodic convolutions for the discrete Fourier transform we need to periodise ω. Let $\widetilde{\omega}$ be a $2L$-periodic extension of ω such that $\widetilde{\omega}(t + n2L) = \omega(t)$ for all $t \in [-L, L)$ and $n \in \mathbb{Z}$. We further approximate

$$\int_{-L}^{L} \omega(t)\omega(x-t)\,\mathrm{d}t \approx \int_{-L}^{L} \widetilde{\omega}(t)\widetilde{\omega}(x-t)\,\mathrm{d}t. \tag{5.1}$$

5.2 Discretisation of convolutions

Divide the interval $[-L, L]$ on n equidistant points $x_0, \ldots, x_{n-1}$ such that

$$x_i = -L + i\Delta x, \text{ where } \Delta x = 2L/n.$$

Consider the vectors

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_0 \\ \vdots \\ \omega_{n-1} \end{bmatrix} \quad \text{and} \quad \widetilde{\boldsymbol{\omega}} = \begin{bmatrix} \widetilde{\omega}_0 \\ \vdots \\ \widetilde{\omega}_{n-1} \end{bmatrix},$$

where

$$\omega_i = \omega(-L + i\Delta x) \quad \text{and} \quad \widetilde{\omega}_i = \widetilde{\omega}(i\Delta x).$$

Assuming n is even, from the periodicity it follows that

$$\widetilde{\boldsymbol{\omega}} = D\boldsymbol{\omega}, \quad \text{where} \quad D = \begin{bmatrix} 0 & I_{n/2} \\ I_{n/2} & 0 \end{bmatrix}.$$

We approximate (5.1) using a Riemann sum and the convolution theorem (4.1) as

$$\begin{aligned}(\widetilde{\omega} \circledast \widetilde{\omega})(i\Delta x) &= \int_{-L}^{L} \widetilde{\omega}(t)\widetilde{\omega}(i\Delta x - t)\,\mathrm{d}t \\ &\approx \Delta x \sum_{\ell=0}^{n-1} \widetilde{\omega}_\ell\, \widetilde{\omega}_{i-\ell} \text{ (indices modulo } n) \\ &= \Delta x \left[\mathcal{F}^{-1}\big(\mathcal{F}(\widetilde{\boldsymbol{\omega}}) \odot \mathcal{F}(\widetilde{\boldsymbol{\omega}})\big)\right]_i.\end{aligned}$$

Discretisation of k-fold truncated convolutions leads to k-fold discrete convolutions and to the approximation

$$\begin{aligned}&(\widetilde{\omega} \circledast^k \widetilde{\omega})(-L + i\Delta x) \\ &\quad \approx (\Delta x)^{k-1} \left[D\,\mathcal{F}^{-1}\big(\mathcal{F}(\widetilde{\boldsymbol{\omega}})^{\odot k}\big)\right]_i \\ &\quad = (\Delta x)^{-1} \left[D\,\mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{\omega}\Delta x)^{\odot k}\big)\right]_i,\end{aligned}$$

where $^{\odot k}$ denotes kth elementwise power of vectors.

5.3 Approximation of the $\delta(\varepsilon)$-integral

Finally, using the discretised convolutions we approximate the integral formula for the exact δ-value. Denote the discrete convolution vector

$$C^k = (\Delta x)^{-1} \left[D\,\mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{\omega}\Delta x)^{\odot k}\big)\right]$$

and the starting point of the discrete sum

$$\ell_\varepsilon = \min\{\ell \in \mathbb{Z} : -L + \ell\Delta x > \varepsilon\}.$$

Using the vector $C^k = \begin{bmatrix} C_0^k & \ldots & C_{n-1}^k \end{bmatrix}^T$, we approximate the integral formula given in Thm. 6 as a Riemann sum:

$$\begin{aligned}\delta(\varepsilon) &= \int_{\varepsilon}^{\infty} (1 - \mathrm{e}^{\varepsilon - s})(\omega *^k \omega)(s)\,\mathrm{d}s \\ &\approx \Delta x \sum_{\ell=\ell_\varepsilon}^{n-1} \big(1 - \mathrm{e}^{\varepsilon - (-L + \ell\Delta x)}\big) C_\ell^k.\end{aligned} \tag{5.2}$$

We call this method the Fourier Accountant (FA) and describe it in the pseudocode of Alg. 1. The computational cost of the method is dominated by applying FFT and its inverse. Thus Alg. 1 has running time complexity $O(n \log n)$. We give in Sec. 7 estimates to determine the parameters L and n such that the error caused by approximations is below a desired level.

5.4 Computing $\varepsilon(\delta)$ using Newton's method

In order to get the function $\varepsilon(\delta)$, we compute the inverse of $\delta(\varepsilon)$ using Newton's method. From (5.2) it follows that (see Lemma D.1 of Appendix)

$$\delta'(\varepsilon) = -\int_{\varepsilon}^{\infty} \mathrm{e}^{\varepsilon - s}(\omega *^k \omega)(s)\,\mathrm{d}s. \tag{5.3}$$

Thus, in order to find ε such that $\delta(\varepsilon) = \bar{\delta}$, we apply Newton's method (Stoer and Bulirsch, 2013) to the function $\delta(\varepsilon) - \bar{\delta}$ which gives the iteration

$$\varepsilon_{\ell+1} = \varepsilon_\ell - \frac{\delta(\varepsilon_\ell) - \bar{\delta}}{\delta'(\varepsilon_\ell)}.$$

Evaluating $\delta'(\varepsilon)$ for different values of ε is cheap using the formula (5.3) and an approximation analogous to (5.2). As is common practice, we use as a stopping criterion $\left|\delta(\varepsilon_\ell) - \bar{\delta}\right| \leq \tau$ for some prescribed tolerance parameter τ. The iteration was found to converge in all experiments with an initial value $\varepsilon_0 = 0$.

Algorithm 1 Fourier Accountant algorithm

Input: privacy loss distribution ω, number of compositions k, truncation parameter L, number of discretisation points n.

Evaluate the discrete distribution values

$$\omega_i = \omega(-L + i\Delta x), \quad i = 0, \ldots, n-1,\ \Delta x = \tfrac{2L}{n}.$$

Set

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_0 \\ \vdots \\ \omega_{n-1} \end{bmatrix}$$

Evaluate

$$C^k = (\Delta x)^{-1} \left[D\, \mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{\omega}\Delta x)^{\odot k}\big)\right],$$
$$\ell_\varepsilon = \min\{\ell \in \mathbb{Z} : -L + \ell\Delta x > \varepsilon\}.$$

Evaluate the approximation

$$\delta(\varepsilon) \approx \Delta x \sum\nolimits_{\ell=\ell_\varepsilon}^{n-1} \big(1 - \mathrm{e}^{\varepsilon - (-L + \ell\Delta x)}\big) C_\ell^k.$$

5.5 Approximation for varying mechanisms

Our approach also allows computing privacy cost of a composite mechanism $\mathcal{M}_1 \circ \ldots \circ \mathcal{M}_k$, where the PLDs of the mechanisms $\mathcal{M}_i$ vary. This is needed for example when accounting the privacy loss of Stochastic Gradient Langevin Dynamics iterations (Wang et al., 2015), where decreasing the step size increases σ.

In this case the function $\delta(\varepsilon)$ is given by Thm. A.7 of Appendix by an integral formula of the form

$$\delta(\varepsilon) = \int_\varepsilon^\infty (1 - \mathrm{e}^{\varepsilon - s})(\omega_1 * \ldots * \omega_k)(s)\, \mathrm{d}s,$$

where ω_i's are PLD distributions determined by the mechanisms $\mathcal{M}_i$, $1 \le i \le k$.

Denoting $C = (\Delta x)^{-1} \left[D\, \mathcal{F}^{-1}\big(F_1 \odot \ldots \odot F_k\big)\right]$, where $F_i = \mathcal{F}(D\boldsymbol{\omega_i}\Delta x)$ and $\boldsymbol{\omega_i}$'s are obtained from discretisations of ω_i's (as in Sec. 5.2), then $\delta(\varepsilon)$ can be approximated as in (5.2).

6 Subsampled Gaussian mechanism

The main motivation for this work comes from privacy accounting of the subsampled Gaussian mechanism which gives privacy bounds for DP-SGD (Abadi et al., 2016b). In Appendix, we show that the worst case privacy analysis of DP-SGD can be carried out by analysis of one dimensional probability distributions. We derive the privacy loss distributions for three different subsampling methods: Poisson subsampling with both $\sim_R$- and $\sim_S$-neighbouring relations, sampling without replacement with $\sim_S$-neighbouring relation and sampling with replacement with $\sim_S$-neighbouring relation. We note the following related works. Balle et al. (2018) consider RDP bounds for these three subsampling methods, Wang et al. (2019) give improved RDP bounds for the case of sampling without replacement and Zhu and Wang (2019) give tight RDP bounds for the case of Poisson subsampling.

6.1 Poisson subsampling for $(\varepsilon, \delta, \sim_R)$-DP

We start with the Poisson subsampling method, where each member of the dataset is included in the stochastic gradient minibatch with probability q. This method is also used in the moments accountant (Abadi et al., 2016b), and also considered by Meiser and Mohammadi (2018) and Wang et al. (2019). As we show in Appendix, the $(\varepsilon, \delta, \sim_R)$-DP analysis of the Poisson subsampling is equivalent to considering the following one dimensional distributions:

$$f_X(t) = q\, \tfrac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-(t-1)^2}{2\sigma^2}} + (1-q)\, \tfrac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-t^2}{2\sigma^2}},$$
$$f_Y(t) = \tfrac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-t^2}{2\sigma^2}}.$$

Here σ^2 denotes the variance of the additive Gaussian noise. Using Definition 3, the privacy loss function is given by

$$\mathcal{L}_{X/Y}(t) = \log \frac{q\, \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-(t-1)^2}{2\sigma^2}} + (1-q)\, \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-t^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{\frac{-t^2}{2\sigma^2}}}$$
$$= \log\left(q\, \mathrm{e}^{\frac{2t-1}{2\sigma^2}} + (1-q)\right).$$

Now $\mathcal{L}_{X/Y}(\mathbb{R}) = (\log(1-q), \infty)$ and $\mathcal{L}_{X/Y}$ is again a strictly increasing continuously differentiable bijective function in the whole $\mathbb{R}$. Straightforward calculation shows that

$$\mathcal{L}_{X/Y}^{-1}(s) = \sigma^2 \log \frac{\mathrm{e}^s - (1-q)}{q} + \frac{1}{2}.$$

Moreover,

$$\frac{\mathrm{d}}{\mathrm{d}\, s} \mathcal{L}_{X/Y}^{-1}(s) = \frac{\sigma^2 \mathrm{e}^s}{\mathrm{e}^s - (1-q)}.$$

The privacy loss distribution $\omega_{X/Y}$ is determined by the density function given in Def. 4. Lemma A.9 and its corollary explain the observation that generally $\delta_{X/Y} > \delta_{Y/X}$.

6.2 Sampling without replacement for $(\varepsilon, \delta, \sim_S)$-DP

We next consider the $\sim_S$-neighbouring relation and sampling without replacement. In this case the batch

size m is fixed and each member of the dataset contributes at most once for each minibatch. Here $q = m/n$, where n denotes the total number of data samples. Without loss of generality we consider here the density functions

$$f_X(t) = q\,\tfrac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\frac{-(t-1)^2}{2\sigma^2}} + (1-q)\tfrac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\frac{-t^2}{2\sigma^2}},$$
$$f_Y(t) = q\,\tfrac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\frac{-(t+1)^2}{2\sigma^2}} + (1-q)\tfrac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\frac{-t^2}{2\sigma^2}}.$$

The privacy loss function is now given by

$$\mathcal{L}_{X/Y}(t) = \log\left(\frac{q\,\mathrm{e}^{\frac{2t-1}{2\sigma^2}} + (1-q)}{q\,\mathrm{e}^{\frac{-2t-1}{2\sigma^2}} + (1-q)}\right).$$

We see that $\mathcal{L}_{X/Y}(\mathbb{R}) = \mathbb{R}$ and again that $\mathcal{L}_{X/Y}$ is a strictly increasing continuously differentiable function. With a straightforward calculation we find that

$$\mathcal{L}^{-1}_{X/Y}(s) = \sigma^2\log\Big(\frac{1}{2c}\big(-(1-q)(1-\mathrm{e}^s) + \sqrt{(1-q)^2(1-\mathrm{e}^s)^2 + 4c^2\mathrm{e}^s}\big)\Big),$$

where $c = q\,\mathrm{e}^{-\frac{1}{2\sigma^2}}$.

Using Lemma A.9 and the property $f_Y(-t) = f_X(t)$, we see that $\delta = \delta_{Y/X} = \delta_{X/Y}$.

We remark that in $(\varepsilon, \delta, \sim_S)$-DP, the Poisson subsampling with the sampling parameter γ is equivalent to the case of the sampling without replacement with $q = \gamma$, as in both cases the differing element is included in the minibatch with probability γ.

6.3 Sampling with replacement

Consider next the sampling with replacement and the $\sim_S$-neighbouring relation. Again the batch size is fixed, however this time each element of the minibatch is drawn from the dataset with probability q. Thus the number of contributions of each member of the dataset is not limited. Then ℓ, the number of times the differing sample x' is in the batch, is binomially distributed, i.e., $\ell \sim \mathrm{Binomial}(1/n, m)$, where m denotes the batch size and n the total number of data samples.

Without loss of generality, we consider here the density functions

$$f_X(t) = \tfrac{1}{\sqrt{2\pi\sigma^2}}\sum_{\ell=0}^{m}\left(\frac{1}{n}\right)^{\ell}\left(1-\frac{1}{n}\right)^{m-\ell}\binom{m}{\ell}\mathrm{e}^{\frac{-(t-\ell)^2}{2\sigma^2}},$$
$$f_Y(t) = \tfrac{1}{\sqrt{2\pi\sigma^2}}\sum_{\ell=0}^{m}\left(\frac{1}{n}\right)^{\ell}\left(1-\frac{1}{n}\right)^{m-\ell}\binom{m}{\ell}\mathrm{e}^{\frac{-(t+\ell)^2}{2\sigma^2}}.$$

The privacy loss function is then given by

$$\mathcal{L}_{X/Y}(t) = \log\left(\frac{\sum_{\ell=0}^{m} c_\ell x^\ell}{\sum_{\ell=0}^{m} c_\ell x^{-\ell}}\right),$$

where

$$c_\ell = \left(\frac{1}{n}\right)^{\ell}\left(1-\frac{1}{n}\right)^{m-\ell}\binom{m}{\ell}\mathrm{e}^{\frac{-\ell^2}{2\sigma^2}}, \quad x = \mathrm{e}^{\frac{t}{\sigma^2}}.$$

Since $c_\ell > 0$ for all $\ell = 1, \ldots, m$, clearly $\sum_{\ell=0}^{m} c_\ell x^\ell$ is strictly increasing as a function of t and $\sum_{\ell=0}^{m} c_\ell x^{-\ell}$ is strictly decreasing. Moreover, we see that

$$\frac{\sum_{\ell=0}^{m} c_\ell x^\ell}{\sum_{\ell=0}^{m} c_\ell x^{-\ell}} \to 0 \quad \text{as } t \to -\infty$$

and

$$\frac{\sum_{\ell=0}^{m} c_\ell x^\ell}{\sum_{\ell=0}^{m} c_\ell x^{-\ell}} \to \infty \quad \text{as } t \to \infty.$$

Thus, $\mathcal{L}_{X/Y}(\mathbb{R}) = \mathbb{R}$ and $\mathcal{L}_{X/Y}(t)$ is a strictly increasing continuously differentiable function in its domain. To find $\mathcal{L}^{-1}_{X/Y}(s)$ one needs to solve $\mathcal{L}_{X/Y}(t) = s$, i.e., one needs to find the single positive real root of a polynomial of order $2m$. As in the case of subsampling without replacement, here $\delta = \delta_{Y/X} = \delta_{X/Y}$.

7 Error estimates

We give error estimates for the Poisson subsampling method with the neighbouring relation $\sim_R$. Thus, in this section ω denotes the PLD density function defined in Sec. 6.1. The estimates are determined by the parameters L and n, the truncation interval radius and the number of discretisation points, respectively.

The total error consists of (see Thm. C.1 in Appendix)

1. The errors arising from the truncation of the convolution integrals and periodisation.
2. The error from neglecting the tail integral
$$\int_L^\infty (1-\mathrm{e}^{\varepsilon-s})(\omega *^k \omega)(s)\,\mathrm{d}s. \tag{7.1}$$
3. The numerical errors in the approximation of the convolutions $(\omega *^k \omega)$ and in the Riemann sum approximation (5.2).

We obtain bounds for the first two sources of error, i.e., for the tail integral (7.1) and the periodisation error, using the Chernoff bound (Wainwright, 2019)

$$\mathbb{P}[X \geq t] = \mathbb{P}[\mathrm{e}^{\lambda X} \geq \mathrm{e}^{\lambda t}] \leq \frac{\mathbb{E}[\mathrm{e}^{\lambda X}]}{\mathrm{e}^{\lambda t}},$$

which holds for any random variable X and all $\lambda > 0$. Denoting also the PLD random variable by ω, the moment generating function $\mathbb{E}[\mathrm{e}^{\lambda\omega}]$ is related to the log of the moment generating function of the privacy loss

function $\mathcal{L} = \mathcal{L}_{X/Y}$ as follows. Define (see also (Abadi et al., 2016b))

$$\alpha(\lambda) := \log \mathop{\mathbb{E}}_{t\sim f_X(t)}[\mathrm{e}^{\lambda\mathcal{L}(t)}].$$

By the change of variable $s = \mathcal{L}(t)$ we have

$$\begin{aligned}\mathbb{E}[\mathrm{e}^{\lambda\omega}] &= \int_{-\infty}^{\infty} \mathrm{e}^{\lambda s}\omega(s)\,\mathrm{d}s \\ &= \int_{\log(1-q)}^{\infty} \mathrm{e}^{\lambda s} f_X(\mathcal{L}^{-1}(s))\frac{\mathrm{d}\mathcal{L}^{-1}(s)}{\mathrm{d}s}\,\mathrm{d}s \quad (7.2)\\ &= \int_{-\infty}^{\infty} \mathrm{e}^{\lambda\mathcal{L}(t)} f_X(t)\,\mathrm{d}t = \mathrm{e}^{\alpha(\lambda)}.\end{aligned}$$

Using existing bounds for $\alpha(\lambda)$ given by Abadi et al. (2016b) and Mironov et al. (2019), we bound $\mathbb{E}[\mathrm{e}^{\lambda\omega}]$ and obtain the required tail bounds.

7.1 Periodisation and truncation of convolutions

We have the following bound for the error arising from the periodisation and the truncation of the convolution integrals. The proof is given in Appendix, Lemma C.6.

Lemma 7. *Let $0 < q < \frac{1}{2}$. Let ω be defined as in Sec. 6.1, and let $L \geq 1$. Then, for all $x \in \mathbb{R}$,*

$$\begin{aligned}\left|\int_{\varepsilon}^{L} (\omega *^k \omega - \widetilde{\omega} \circledast^k \widetilde{\omega})(x)\,\mathrm{d}x\right| &\leq Lk\sigma \mathrm{e}^{-\frac{(\sigma^2 L + C)^2}{2\sigma^2}} \\ &+ \mathrm{e}^{\alpha(L/2)}\mathrm{e}^{-\frac{L^2}{2}} + 2\sum\nolimits_{n=1}^{\infty} \mathrm{e}^{k\alpha(nL)}\mathrm{e}^{-2(nL)^2},\end{aligned}$$

where $C = \sigma^2 \log(\frac{1}{2q}) - \frac{1}{2}$.

For example, setting σ, q as in the example of Figure 2, and $k = 2\cdot 10^4$, the first term is $O(10^{-16})$ already for $L = 4.0$. The second term dominates the rest of the bound of Lemma 7 and it is much smaller than the tail bound (7.3) ($\mathrm{e}^{\alpha(L/2)}$ vs. $\mathrm{e}^{k\alpha(L/2)}$). Therefore, this error is much smaller than estimates for the tail integral (7.1) and it is neglected in the estimates.

7.2 Convolution tail bound

Let ω denote the PLD density function. Now, the tail of the integral representation for δ (Thm. 6), with $L > \varepsilon$, can be bounded as

$$\int_L^{\infty} (1 - \mathrm{e}^{\varepsilon - s})(\omega *^k \omega)(s)\,\mathrm{d}s < \int_L^{\infty} (\omega *^k \omega)(s)\,\mathrm{d}s.$$

We consider both upper bounds and estimates for the tail integral of convolutions.

7.2.1 Analytic tail bound

Using the Chernoff bound we derive an analytic bound for the tail integral of convolutions. In a certain sense this is equivalent to finding bounds for the RDP parameters, since an RDP bound gives a bound also for the moment generating function $\mathbb{E}[\mathrm{e}^{\lambda\omega}]$ needed in the Chernoff bound. The following result is derived from recent RDP results (Mironov et al., 2019). The proof and an illustration of the result are given in Appendix.

Theorem 8. *Suppose $q \leq \frac{1}{5}$ and $\sigma \geq 4$. Let L be chosen such that $\lambda = L/2$ satisfies*

$$\begin{aligned}1 < \lambda &\leq \frac{1}{2}\sigma^2 c - 2\log\sigma, \\ \lambda &\leq \frac{\frac{1}{2}\sigma^2 c - \log 5 - 2\log\sigma}{c + \log(q\lambda) + 1/(2\sigma^2)},\end{aligned}$$

where $c = \log\left(1 + \frac{1}{q(\lambda-1)}\right)$. Then, we have

$$\int_L^{\infty} (\omega *^k \omega)(s)\,\mathrm{d}s \leq \left(1 + \frac{2q^2(\frac{L}{2}+1)\frac{L}{2}}{\sigma^2}\right)^k \mathrm{e}^{-\frac{L^2}{2}}.$$

In order to avoid the restriction on σ in Thm. 8, we consider an approximative bound.

7.2.2 Tail bound estimate

We next derive an approximative tail bound using the $\alpha(\lambda)$-bound given by Abadi et al. (2016b). Denote $S_k := \sum_{i=1}^k \omega^i$, where ω^i denotes the PLD random variable of the ith mechanism. Since ω^i's are independent, $\mathbb{E}[\mathrm{e}^{\lambda S_k}] = \prod_{i=1}^k \mathbb{E}[\mathrm{e}^{\lambda\omega^i}]$ and the Chernoff bound shows that

$$\int_L^{\infty} (\omega *^k \omega)(s)\,\mathrm{d}s = \mathbb{P}[S_k \geq L] \leq \mathrm{e}^{k\alpha(\lambda)}\mathrm{e}^{-\lambda L}$$

for any $\lambda > 0$. We recall the result by Abadi et al. (2016b, Lemma 3) which holds for the Poisson subsampling method.

Lemma 9. *Let $\sigma \geq 1$ and $q < \frac{1}{16\sigma}$, then for any positive integer $\lambda \leq \sigma^2 \ln\frac{1}{q\sigma}$,*

$$\alpha(\lambda) \leq \frac{q^2\lambda(\lambda+1)}{(1-q)\sigma^2} + \mathcal{O}(q^3\lambda^3/\sigma^3).$$

Suppose the conditions of Lemma 9 hold for $\lambda = L/2$. Substituting the bound of Lemma 9 to the Chernoff bound and neglecting the $\mathcal{O}(q^3\lambda^3/\sigma^3)$-term gives the approximative upper bound

$$\int_L^{\infty} (\omega *^k \omega)(s)\,\mathrm{d}s \lessapprox \exp\left(k\frac{q^2(\frac{L}{2}+1)\frac{L}{2}}{(1-q)\sigma^2}\right)\mathrm{e}^{-\frac{L^2}{2}}. \quad (7.3)$$

For example, when $q = 0.01$ and $\sigma = 2.0$, the conditions of Lemma 9 hold for λ up to ≈ 9.5 (i.e. (7.3) holds for L up to ≈ 19). Figure 2 of Appendix shows the convergence of the bound (7.3) in this case.

7.3 Discretisation errors

Derivation of discretisation error bounds can be carried out using the so called Euler–Maclaurin formula (Sec. C.3 in Appendix). This requires bounds for higher order derivatives of ω. As an illustrating example, consider the bound (recall $\Delta x = 2L/n$)

$$\begin{aligned}&\left|\int_{-L}^{L} \omega(s)\,\mathrm{d}s - \Delta x \sum\nolimits_{\ell=0}^{n-1} \omega(-L+\ell\Delta x)\right| \\ &\quad\le \Delta x\,\omega(L) + \frac{(\Delta x)^2}{12}\max_{t\in[-L,L]}|\omega''(t)| \\ &\quad\le \Delta x\,\sigma \mathrm{e}^{-\frac{-(\sigma^2 L+C)^2}{2\sigma^2}} + \frac{(\Delta x)^2}{12}\max_{t\in[-L,L]}|\omega''(t)|\,,\end{aligned}$$

where $C = \sigma^2\log(\frac{1}{2q}) - \frac{1}{2}$. By Lemma D.4, $\max_t|\omega''(t)|$ has an upper bound $O(\sigma^3/q^3)$. With bounds for higher order derivatives, tighter error bound could be obtained. In a similar fashion, bounds for the errors for the approximation (5.2) could be derived. However, we resort to numerical estimates.

7.3.1 Estimate for the discretisation error

Consider the error arising from the Riemann sum

$$I_n := \Delta x \sum\nolimits_{\ell=\ell_\varepsilon}^{n-1} \big(1-\mathrm{e}^{\varepsilon-(-L+\ell\Delta x)}\big)C_\ell^k.$$

As we show in Sec. C.3 of Appendix, it holds

$$\begin{aligned}E_n :=& \int_{\varepsilon}^{L} (1-\mathrm{e}^{\varepsilon-s})(\widetilde{\omega}\circledast^k\widetilde{\omega})(s)\,\mathrm{d}s - I_n \\ =& K\Delta x + O((\Delta x)^2) = K\frac{2L}{n} + O\Big(\Big(\frac{2L}{n}\Big)^2\Big)\end{aligned}$$

for some constant K independent of n. Therefore,

$$2(I_n - I_{2n}) = E_n + O((\Delta x)^2)$$

which leads us to use as an estimate

$$\mathrm{err}(L,n) := 2\,|I_n - I_{2n}| \tag{7.4}$$

for the numerical error E_n.

8 Experiments

In all experiments, we consider the Poisson subsampling with $(\varepsilon,\delta,\sim_R)$-DP (Sec. 6.1).

We first illustrate the numerical convergence of FA for $\delta(\varepsilon)$ and the estimates (7.3) and (7.4), when $k = 10^4$, $q = 0.01$, $\sigma = 1.5$ and $\varepsilon = 1.0$ (Tables 1 and 2). We emphasise that the error estimates (7.3) and (7.4) represent the distance to the tight $\delta(\varepsilon)$-value. Full numerical tables are given in Appendix, Sec. D.2.

n	FA	$\mathrm{err}(L,n)$
$5\cdot10^4$	0.0491228786423	$2.01\cdot10^{-2}$
$2\cdot10^5$	0.0496013846114	$1.06\cdot10^{-6}$
$8\cdot10^5$	0.0496014103252	$2.66\cdot10^{-11}$
$3.2\cdot10^6$	0.0496014103163	$2.22\cdot10^{-12}$

Table 1: Convergence of $\delta(\varepsilon)$-approximation with respect to n (when $L = 12$) and the estimate (7.4). The tail bound estimate (7.3) is $O(10^{-24})$.

L	FA	estimate (7.3)
2.0	0.0422160172923	$3.32\cdot10^{-1}$
6.0	0.0496014103158	$3.32\cdot10^{-6}$
10.0	0.0496014103134	$1.36\cdot10^{-16}$
12.0	0.0496014103163	$8.30\cdot10^{-24}$

Table 2: Convergence of the $\delta(\varepsilon)$-approximation with respect to L (when $n = 3.2\cdot10^6$) and the error estimate (7.3). The estimate $\mathrm{err}(L,n) = O(10^{-12})$.

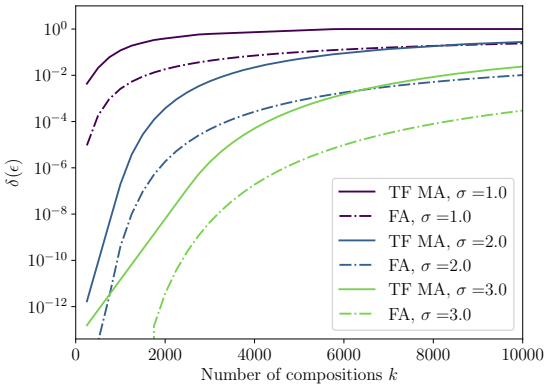
We first compare the Fourier accountant method to the privacy accountant method included in the Tensorflow library (Abadi et al., 2016a) which is the moments accountant method (Abadi et al., 2016b) (Figure 1). We use $q = 0.01$ and $\sigma \in \{1.0, 2.0, 3.0\}$, for number of compositions k up to 10^4. We set the parameters $L = 12$ and $n = 5\cdot10^6$ for the approximation of the exact integral. Then, for $\sigma = 1.0$, the tail integral error estimate (7.3) is at most $O(10^{-13})$ and the estimate $\mathrm{err}(L,n)$ is at most $O(10^{-10})$. For $\sigma = 2.0, 3.0$ the error estimates are smaller.

We next compare FA to the RDP accountant method described by Zhu and Wang (2019) (Figure 2). Although the RDP accountant gives tight RDP-bounds, there is a small gap to the tight $(\varepsilon,\delta,\sim_R)$-DP.

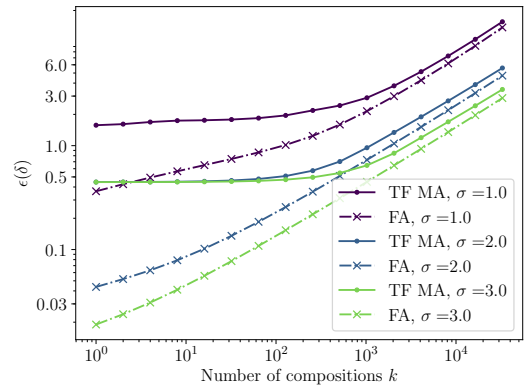
As we see from Figures 1b and 2, the moments accountant and the RDP bound (Zhu and Wang, 2019) do not capture the true ε-bound for small number of compositions k, whereas FA gives tight bounds.

Figure 3 shows a comparison of FA to the Berry–Esseen theorem based bound given by Sommer et al. (2019, Thm. 6). The Berry–Esseen bound suffers from the error term which converges as $O(k^{-\frac{1}{2}})$.

Lastly, we compare FA to the Privacy Buckets (PB) algorithm (Meiser and Mohammadi, 2018) (Figure 4). The additional ratio parameter of PB was tuned for the experiments. The algorithm seems to suffer from some instabilities which is also mentioned by Meiser and Mohammadi (2018). For larger σ and smaller q PB gave bounds closer to that of FA, however the com-



(a) $\delta(\varepsilon)$ as a function of k for $\varepsilon = 1.0$.



(b) $\varepsilon(\delta)$ as a function of k for $\delta = 10^{-6}$.

Figure 1: Comparison of the Tensorflow moments accountant and the Fourier accountant. Here $q = 0.01$.

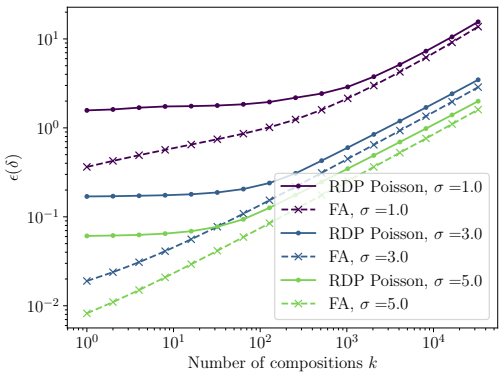


Figure 2: Comparison of the RDP bound for the Poisson subsampling (Zhu and Wang, 2019) and FA. Here $\delta = 10^{-6}$, $q = 0.01$.

pute times were always much bigger, as in experiments of Figure 4.

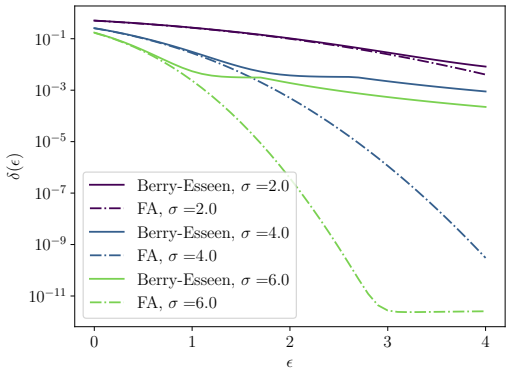


Figure 3: Comparison of the Berry–Esseen bound and FA for $(\varepsilon, \delta, \sim_R)$-DP. Here $k = 5 \cdot 10^4$, $q = 0.01$.

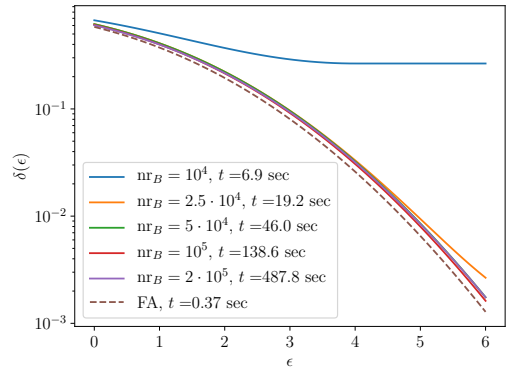


Figure 4: Comparison of the Privacy Buckets algorithm (nr_B = number of buckets) and FA. Legend contains compute times. Here $k = 2^{12}$, $\sigma = 1.0$, $q = 0.02$.

9 Conclusions

We have presented a novel approach for computing tight privacy bounds for DP. Although we have focused on the subsampled Gaussian mechanism (with various subsampling strategies), our method is applicable also to other mechanisms. We remark that the assumptions of Def. 4 would not hold for example for the Laplace mechanism: then the PLD distribution becomes a discrete/continuous mixture distribution. However, using Lemma A.2 the integral formula of Thm. 6 can be shown to hold also in this case and the FA algorithm can also be applied to this case. This is left for future work. As future work, it would also be interesting to carry out a full error analysis for the discretisation error. Moreover, evaluating the privacy parameters for compositions involving both continuous and discrete valued mechanisms is an interesting objective.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016a). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

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

Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6277–6287.

Bassily, R., Smith, A., and Thakurta, A. (2014). 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, pages 464–473, Washington, DC, USA. IEEE Computer Society.

Beimel, A., Nissim, K., and Stemmer, U. (2013). Characterizing the sample complexity of private learners. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 97–110, New York, NY, USA. ACM.

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

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

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Proc. TCC 2006*, pages 265–284. Springer Berlin Heidelberg.

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

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). 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.

Kairouz, P., Oh, S., and Viswanath, P. (2017). The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049.

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.

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

Mironov, I., Talwar, K., and Zhang, L. (2019). Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*.

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

Sommer, D. M., Meiser, S., and Mohammadi, E. (2019). Privacy loss classes: The central limit theorem in differential privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(2):245–269.

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

Stockham Jr, T. G. (1966). High-speed convolution and correlation. In *Proceedings of the April 26-28, 1966, Spring joint computer conference*, pages 229–233. ACM.

Stoer, J. and Bulirsch, R. (2013). *Introduction to numerical analysis*, volume 12. Springer Science & Business Media.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

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

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. (2019). Subsampled Rényi differential privacy and analytical moments accountant. In *Proc. AISTATS 2019*.

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