COMPUTING DIFFERENTIAL PRIVACY GUARANTEES FOR HETEROGENEOUS COMPOSITIONS USING FFT

Antti Koskela & Antti Honkela
Helsinki Institute for Information Technology HIIT,
Department of Computer Science, University of Helsinki, Finland
{antti.h.koskela,antti.honkela}@helsinki.fi

ABSTRACT

The recently proposed Fast Fourier Transform (FFT)-based accountant for evaluating (ε, δ)-differential privacy guarantees using the privacy loss distribution formalism has been shown to give tighter bounds than commonly used methods such as Rényi accountants when applied to compositions of homogeneous mechanisms. This approach is also applicable to certain discrete mechanisms that cannot be analysed with Rényi accountants. We extend this approach to compositions of heterogeneous mechanisms. We carry out a full error analysis that allows choosing the parameters of the algorithm such that a desired accuracy is obtained.

1 INTRODUCTION

Differential privacy (DP) (Dwork et al., 2006) has become the standard approach for privacy-preserving machine learning. When using DP, one challenge is to accurately bound the compound privacy loss of the increasingly complex DP algorithms. This work extends the recent Fast Fourier Transform (FFT) accountant by Koskela et al. (2020a;b) to heterogeneous compositions of discrete mechanisms, using the privacy loss distribution (PLD) formalism introduced by Sommer et al. (2019). We also illustrate how to apply this accountant to continuous mechanisms and discrete-continuous hybrids. Experimental comparisons to the Tensorflow moments accountant show that the FFT-based method allows approximately 1.5 times as many compositions for equal privacy guarantees. We provide a rigorous error analysis for the proposed method in terms of the truncation and discretisation parameters L and n. This analysis both leads to strict upper (ε, δ)-bounds and gives means for automatically tuning these parameters. The analysis also gives a bound for the computational complexity in terms of the error which is analogous to the one given by Murtagh and Vadhan (2018). We also show how to speed up the evaluation using the Plancherel theorem, at the cost of increased pre-computation and memory usage.

More details on the results can be found in the supplementary material.

1.1 DIFFERENTIAL PRIVACY AND PRIVACY LOSS DISTRIBUTION

We first recall some basic definitions of DP (Dwork et al., 2006). An input data set 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 data sets X and Y are neighbours in remove/add relation if we get one by removing/adding an element from/to the other (denoted $\sim_R$). We say X and Y are neighbours in substitute relation if we get one by substituting one element in the other (denoted $\sim_S$).*

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 $E \subset \mathcal{R}$ we have*

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

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.

We consider discrete-valued one-dimensional mechanisms $\mathcal{M}$ which can be seen as mappings from $\mathcal{X}^N$ to the set of discrete-valued random variables. The *generalised probability density functions* of

$\mathcal{M}(X)$ and $\mathcal{M}(Y)$, denoted $f_X(t)$ and $f_Y(t)$, respectively, are given by

$$f_X(t) = \sum\nolimits_i a_{X,i} \cdot \delta_{t_{X,i}}(t), \quad f_Y(t) = \sum\nolimits_i a_{Y,i} \cdot \delta_{t_{Y,i}}(t), \tag{1.1}$$

where $\delta_t(\cdot)$, $t \in \mathbb{R}$, denotes the Dirac delta function centred at t, and $t_{X,i}, t_{Y,i} \in \mathbb{R}$ and $a_{X,i}, a_{Y,i} \geq 0$. The privacy loss distribution is defined as follows.

Definition 3. *Let $\mathcal{M} : \mathcal{X}^N \to \mathcal{R}$, $\mathcal{R} \subset \mathbb{R}$, be a discrete-valued randomised mechanism and let $f_X(t)$ and $f_Y(t)$ be probability density functions as defined by* (B.1)*. We define the generalised privacy loss distribution (PLD) $\omega_{X/Y}$ as*

$$\omega_{X/Y}(s) = \sum\nolimits_{t_{X,i}=t_{Y,j}} a_{X,i} \cdot \delta_{s_i}(s), \quad s_i = \log\left(\tfrac{a_{X,i}}{a_{Y,j}}\right). \tag{1.2}$$

2 Fourier Accountant for Heterogeneous Compositions

Similarly as in (Koskela et al., 2020b), we place the PLD on a grid $X_n = \{x_0, \ldots, x_{n-1}\}$, $n \in \mathbb{Z}^+$, where $x_i = -L + i\Delta x$, $\Delta x = 2L/n$. Suppose the distribution ω of the PLD is of the form

$$\omega(s) = \sum\nolimits_i a_i \cdot \delta_{s_i}(s), \tag{2.1}$$

where $a_i \geq 0$ and $-L \leq s_i \leq L - \Delta x$ for all i. The integral $\mathbb{E}_{s\sim\omega(s)}[(1 - \mathrm{e}^{\varepsilon - s})]$ then leads to the tight $\delta(\varepsilon)$-value (see the Supplements). We define the right grid approximation

$$\omega^{\mathrm{R}}(s) := \sum\nolimits_i a_i \cdot \delta_{s_i^{\mathrm{R}}}(s), \quad s_i^{\mathrm{R}} = \min\{x \in X_n \,:\, x \geq s_i\}. \tag{2.2}$$

We directly get the following result which holds for heterogeneous compositions:

Lemma 4. *Consider a composition with PLDs $\omega_1, \cdots, \omega_k$ (as in eq. C.3). Let $\delta^{\mathrm{R}}(\varepsilon)$ correspondingly be determined by $\omega_1^{\mathrm{R}}, \ldots, \omega_k^{\mathrm{R}}$ (as in eq. C.4). Then for all $\varepsilon > 0$: $\delta(\varepsilon) \leq \delta^{\mathrm{R}}(\varepsilon)$.*

We note that often a moderate L is sufficient for the condition $-L \leq s_i \leq L - \Delta x$ to hold for all i. We also provide analysis for the case where this assumption does not hold (Supplements). This is the case, for example, for the discrete Gaussian distribution (Canonne et al., 2020).

We compute strict $\delta(\varepsilon)$-upper bound as follows: using values $L > 0$ and $n \in \mathbb{Z}^+$, we form a grid X_n and place each PLD ω_i, $1 \leq i \leq k$, on X_n to obtain ω_i^{R}'s as defined in (C.4). For examples of ω_i's in case of the randomised response and a discretisation of the continuous Gaussian mechanism, see the experiments of Section 5.1. We then approximate $\delta^R(\varepsilon)$ using Algorithm 2. Our error analysis bounds the error incurred by Algorithm 2. By adding this error to the approximated $\delta(\varepsilon)$-values, we get strict upper bounds.

Algorithm 1 Fourier Accountant Algorithm for Heterogeneous Discrete-Valued Mechanisms

Input: distributions $\omega_1, \ldots, \omega_m$ of the form $\omega_j(s) = \sum_i a_i^j \cdot \delta_{s_i}(s)$, $1 \leq j \leq m$, such that $s_i = -L + i\Delta x$, where n is even and, $0 \leq i \leq n-1$, $\Delta x = 2L/n$. Numbers of compositions for each mechanism, $k_1, \ldots, k_m$.

Set

$$\boldsymbol{a}^j = \begin{bmatrix} a_0^j & \ldots & a_{n-1}^j \end{bmatrix}^{\mathrm{T}}, \quad 1 \leq j \leq m, \quad D = \begin{bmatrix} 0 & I_{n/2} \\ I_{n/2} & 0 \end{bmatrix}.$$

For each j, $1 \leq j \leq m$, evaluate the FFT: $\quad \widetilde{\boldsymbol{a}}^j = \mathcal{F}(D\boldsymbol{a}^j)$.
Compute the element-wise products and apply $\mathcal{F}^{-1}$:

$$\boldsymbol{b} = \left[D\,\mathcal{F}^{-1}\left((\widetilde{\boldsymbol{a}}^1)^{\odot k_1} \odot \cdots \odot (\widetilde{\boldsymbol{a}}^m)^{\odot k_m}\right)\right].$$

Approximate: $\delta(\varepsilon) \approx 1 - \prod_{\ell=1}^{m} (1 - \delta_{X/Y,\ell}(\infty))^{k_\ell} + \sum_{\{\ell\,:\,-L+\ell\Delta x > \varepsilon\}} \left(1 - \mathrm{e}^{\varepsilon - (-L+\ell\Delta x)}\right) b_\ell,$
where $\delta_{X/Y,\ell}(\infty)$ is the probability mass outside the shared support of $\mathcal{M}_\ell(X)$ and $\mathcal{M}_\ell(Y)$.

3 UPPER BOUND FOR THE COMPUTATIONAL COMPLEXITY

The results by Murtagh and Vadhan (2018) state that there is no algorithm for computing tight (ε, δ)-bounds that would have polynomial complexity in k. However, Theorem 1.7 by Murtagh and Vadhan (2018) states that allowing a small error in the output, the bounds can be evaluated efficiently. More precisely, given a non-adaptive composition of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$, each mechanism $\mathcal{M}_i$ being tightly $(\varepsilon_i, \delta_i)$-DP, the result states that there exists an algorithm that outputs $\widetilde{\varepsilon}(\delta)$ such that

$$\varepsilon(\delta) \leq \widetilde{\varepsilon}(\delta) \leq \varepsilon(\mathrm{e}^{-\eta^2/2} \cdot \delta) + \eta,$$

where $\varepsilon(\delta)$ gives a tight bound for the composition, and the algorithm runs in time

$$\mathcal{O}\left(\frac{k^3 \cdot \overline{\varepsilon} \cdot (1+\overline{\varepsilon})}{\eta} \log \frac{k^2 \cdot \overline{\varepsilon} \cdot (1+\overline{\varepsilon})}{\eta}\right), \qquad \text{where} \quad \overline{\varepsilon} = \tfrac{1}{k} \sum\nolimits_i \varepsilon_i. \tag{3.1}$$

Assuming there are $m < k$ distinct mechanisms in the composition, our error analysis (Supplements) leads to a slightly tighter complexity bound for the evaluation of tight δ as a function of ε:

Theorem 5. *Consider a non-adaptive composition of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ with corresponding worst-case pairs of distributions $f_{X,i}$ and $f_{Y,i}$, $1 \leq i \leq k$. Suppose the sequence $\mathcal{M}_1, \ldots, \mathcal{M}_k$ consists of m distinct mechanisms. Then, it is possible to have an approximation of $\delta(\varepsilon)$ with error less than η with number of operations*

$$\mathcal{O}\left(\frac{2m \cdot k^2 \cdot C_k}{\eta} \log \frac{k^2 \cdot C_k}{\eta}\right),$$

where

$$C_k = \max\{\tfrac{1}{k} \sum_i D_\infty(f_{X,i} || f_{Y,i}), \tfrac{1}{k} \sum_i D_\infty(f_{Y,i} || f_{X,i})\}, \quad D_\infty(f_X || f_Y) = \sup_{a_{Y,i} \neq 0} \log \frac{a_{X,i}}{a_{Y,i}}$$

and the additional factor in the leading $\mathcal{O}$-constant is the leading constant of the FFT algorithm.

4 FAST EVALUATION USING THE PLANCHEREL THEOREM

When using Algorithm 2 to approximate $\delta(\varepsilon)$, we need to evaluate an expression of the form

$$\boldsymbol{b}^k = D\,\mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m}\big) \tag{4.1}$$

and the sum

$$\widetilde{\delta}(\varepsilon) = \sum\nolimits_{-L+\ell\Delta x > \varepsilon} \big(1 - \mathrm{e}^{\varepsilon - (-L + \ell\Delta x)}\big)\, b_\ell^k. \tag{4.2}$$

When evaluating $\widetilde{\delta}(\varepsilon)$ for different numbers of compositions, the transform $\mathcal{F}^{-1}$ is the most expensive part if the vectors $\mathcal{F}(D\boldsymbol{a}^i)$ are precomputed. The following lemma shows that updates of $\widetilde{\delta}(\varepsilon)$ can actually be performed without using $\mathcal{F}^{-1}$, i.e., in linear time.

Lemma 6. *Denote $\boldsymbol{w}_\varepsilon \in \mathbb{R}^n$ such that $(\boldsymbol{w}_\varepsilon)_\ell = \max\{1 - \mathrm{e}^{\varepsilon - (-L + \ell\Delta x)}, 0\}$. Then, we have that*

$$\widetilde{\delta}(\varepsilon) = \frac{1}{n} \langle \mathcal{F}(D\boldsymbol{w}_\varepsilon), \mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m} \rangle. \tag{4.3}$$

We instantly see that if the vectors $\mathcal{F}(D\boldsymbol{w}_\varepsilon)$ and $\mathcal{F}(D\boldsymbol{a}^i)$, $1 \leq i \leq m$, are precomputed, $\widetilde{\delta}(\varepsilon)$ can be updated in $\mathcal{O}(n)$ time. We believe this approach can be used for designing efficient online (ε, δ)-accountants that also give tight guarantees.

Experimental Illustration. Consider computing $\delta(\varepsilon)$-bound for the subsampled Gaussian mechanism (see Sec. 5.2), for $q = 0.02$ and $\sigma = 2.0$. Evaluate $\delta(\varepsilon)$ after $k = 100, 200, \ldots, 500$ compositions at $\varepsilon = 1.0$. Table 4 illustrates the compute time for each update of $\delta(\varepsilon)$, using a) a pre-computed vector $\mathcal{F}(D\boldsymbol{a})$, the transform $\mathcal{F}^{-1}$ and the summation (D.13) and b) pre-computed vectors $\mathcal{F}(D\boldsymbol{a})^{\odot 100}$ and $\mathcal{F}(D\boldsymbol{w}_\varepsilon)$ and the inner product (D.14).

n	t (ms), eq. D.13	t (ms), eq. D.14	$\delta(\varepsilon)$
$5 \cdot 10^4$	5.8	0.18	$2.900925 \cdot 10^{-6}$
$1 \cdot 10^5$	12	0.36	$2.851835 \cdot 10^{-6}$
$1 \cdot 10^6$	140	5.1	$2.846942 \cdot 10^{-6}$
$5 \cdot 10^6$	750	30	$2.846941 \cdot 10^{-6}$

Table 1: Compute times (in milliseconds) for an update of $\delta(\varepsilon)$-bound using the summation of (D.13) and the inner product of (D.14) and the $\delta(\varepsilon)$-upper bound after $k = 500$ compositions. We see that using Lemma D.8 an accurate update of $\delta(\varepsilon)$ is possible in less than one millisecond.

5 Experiments

5.1 Experiment 1: Compositions of Discrete and Continuous Mechanisms

We consider a non-adaptive composition of k mechanisms of the form $\mathcal{M}(X) = \big(\mathcal{M}_1(X), \widetilde{\mathcal{M}}_2(X), \ldots, \mathcal{M}_{k-1}(X), \widetilde{\mathcal{M}}_k(X)\big)$, where each $\mathcal{M}_i$ is a Gaussian mechanism with sensitivity 1, and each $\widetilde{\mathcal{M}}_i$ is a randomised response mechanism with probability of a correct answer p, $\frac{1}{2} < p < 1$. We know that for the randomised response the PLD leading to the worst-case bound is given by $\omega_{\mathrm{R}}(s) = p \cdot \delta_{c_p}(s) + (1-p) \cdot \delta_{-c_p}(s)$, where $c_p = \log \frac{p}{1-p}$ (Koskela et al., 2020b). Also, for the PLD ω_{G} of the Gaussian mechanism we know that (Sommer et al., 2019) $\omega_{\mathrm{G}} \sim \mathcal{N}\left(\frac{1}{2\sigma^2}, \frac{1}{\sigma^2}\right)$. Let the Δx-grid be defined as above, i.e., let $L > 0$, $n \in \mathbb{Z}^+$, $\Delta x = 2L/n$ and $s_i = -L + i\Delta x$ for all $i \in \mathbb{Z}$. Define

$$\omega_{\mathrm{G,max}}(s) = \sum\nolimits_{i=0}^{n-1} a_i^+ \cdot \delta_{s_i}(s), \quad a_i^+ = \Delta x \cdot \max\nolimits_{s \in [s_{i-1}, s_i]} \omega_{\mathrm{G}}(s). \tag{5.1}$$

Using a bound for the moment generating function of the infinitely extending counterpart of $\omega_{\max}$ and by using Algorithm 2 we obtain a numerical value $\delta_{\max}(\varepsilon)$ (depending on n and L) for which we have that $\delta(\varepsilon) \leq \delta_{\max}(\varepsilon)$, where $\delta(\varepsilon)$ gives a tight bound for the composition $\mathcal{M}(X)$ (see the Supplements). As a comparison, in Figure 1 we also show the guarantees given by Tensorflow moments accountant. We know that for $\alpha > 1$, the α-RDP of the randomised response is given by $\frac{1}{\alpha-1} \log\left(p^\alpha (1-p)^{1-\alpha} + (1-p)^\alpha p^{1-\alpha}\right)$ and correspondingly for the Gaussian mechanism by $\frac{\alpha}{2\sigma^2}$ (Mironov, 2017). As is commonly done, we evaluate RDPs for integer values and sum them up along the compositions. Then, using the moments accountant method the corresponding (ε, δ)-bounds are obtained (Abadi et al., 2016).

5.2 Heterogeneous Subsampled Gaussian Mechanism

We next show how to compute upper bounds for heterogeneous compositions of the subsampled Gaussian mechanism. We consider the Poisson subsampling and $\sim_R$-neighbouring relation. For a subsampling ratio q and noise level σ, the continuous PLD is given by (Koskela et al., 2020a)

$$\omega(s) = \begin{cases} f(g(s))g'(s), & \text{if } s > \log(1-q), \\ 0, & \text{otherwise,} \end{cases}$$

where

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \left[q \mathrm{e}^{\frac{-(t-1)^2}{2\sigma^2}} + (1-q)\mathrm{e}^{-\frac{t^2}{2\sigma^2}}\right], \quad g(s) = \sigma^2 \log\left(\frac{\mathrm{e}^s - (1-q)}{q}\right) + \frac{1}{2}.$$

We obtain an upper-bound PLD $\omega_{\max}$ such that at each Δx-interval of the grid X_n, we place the maximal value of ω multiplied by Δx to the right end-point as a discrete mass (Koskela et al., 2020b, Supplements). Then, using Alg. 2 we obtain a numerical value $\delta_{\max}(\varepsilon)$ such that after k compositions, $\delta(\varepsilon) \leq \delta_{\max}(\varepsilon)$, where $\delta(\varepsilon)$ gives a tight bound for the heterogeneous composition of subsampled Gaussian mechanisms. Figure 2 illustrates the upper bound $\delta_{\max}(\varepsilon)$ as k grows and σ decreases, when $L = 10$ and $n = 10^6$. For comparison, we also show the numerical values given by Tensorflow moments accountant (Abadi et al., 2016).

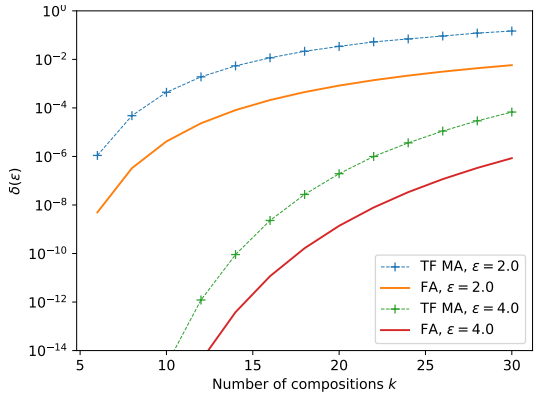

Figure 1: Bounds for $\delta(\varepsilon)$ computed using Algorithm 2 (FA) and Tensorflow moments accountant (TF MA), when $\sigma = 5.0$ and $p = 0.52$, for $\varepsilon = 2.0, 4.0$. We see that when $\delta \in [10^{-6}, 10^{-4}]$, FA allows approximately 1.5 times as many compositions as TF MA for the same ε. We use here $L = 10$ and $n = 10^5$.

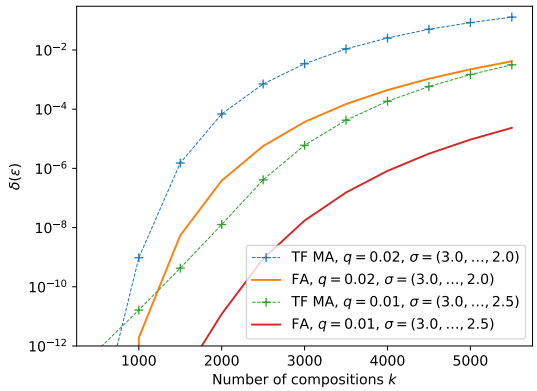

Figure 2: Bounds for $\delta(\varepsilon)$ computed using Algorithm 2 (FA) and Tensorflow moments accountant (TF MA). In the first option $\varepsilon = 1.0$, $q = 0.02$ and σ decreases linearly from 3.0 to 2.0. In the second option $\varepsilon = 1.5$, $q = 0.01$ and σ decreases linearly from 3.0 to 2.5. For each value of σ, 500 compositions are evaluated. We see that when $\delta \in [10^{-6}, 10^{-4}]$, FA allows approximately 1.5 times as many compositions as TF MA for the same guarantees.

6 Conclusions

We have extended the Fast Fourier Transform-based approach for computing tight privacy bounds to heterogeneous compositions. Using the derived error bounds it is possible to determine appropriate values for all the parameters of the algorithm. The error analysis also led to a bound for the computational complexity of the algorithm that is slightly better than the existing theoretical complexity bound for obtaining (ε, δ)-bounds within a given error tolerance. Using the Plancherel theorem, we have shown how to further speed up the evaluation of the privacy guarantees. We believe this gives tools to implement tight privacy accountants to services that require minimal delays. We emphasise that due to the construction of the algorithm and to the rigorous error analysis, the reported (ε, δ)-bounds are strict upper privacy bounds.

Acknowledgements

This work has been supported by the Academy of Finland [Finnish Center for Artificial Intelligence FCAI and grant 325573] and by the Strategic Research Council at the Academy of Finland [grant 336032].

REFERENCES

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., and McMahan, B. (2018). cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems*, pages 7564–7575.

Canonne, C., Kamath, G., and Steinke, T. (2020). The discrete gaussian for differential privacy. In *Advances in Neural Information Processing Systems*.

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.

Koskela, A. and Honkela, A. (2021). Computing differential privacy guarantees for heterogeneous compositions using fft. *arXiv preprint arXiv:2102.12412*.

Koskela, A., Jälkö, J., and Honkela, A. (2020a). Computing tight differential privacy guarantees using FFT. In *The 23rd International Conference on Artificial Intelligence and Statistics*.

Koskela, A., Jälkö, J., Prediger, L., and Honkela, A. (2020b). Tight approximate differential privacy for discrete-valued mechanisms using FFT. *arXiv preprint arXiv:2006.07134*.

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.

Murtagh, J. and Vadhan, S. (2018). The complexity of computing the optimal composition of differential privacy. *Theory of Computing*, 14(8):1–35.

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.

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.

A Details for the Experiments of Section 5.1

For the PLD ω_{G} of the Gaussian mechanism we know that (Sommer et al., 2019)

$$\omega_{\mathrm{G}} \sim \mathcal{N}\left(\frac{1}{2\sigma^2}, \frac{1}{\sigma^2}\right).$$

In order to carry out an error analysis for the approximations given in Section 5.1, we define the infinite extending grid approximation of $\omega_{\max}$. Let $L > 0$, $n \in \mathbb{Z}^+$, $\Delta x = 2L/n$ and let the grid X_n be defined as in (C.2). Define

$$\omega_{\max}(s) = \sum_{i=0}^{n-1} a_i^+ \cdot \delta_{s_i}(s),$$

where $s_i = i\Delta x$ and

$$a_i^+ = \Delta x \cdot \max_{s \in [s_{i-1}, s_i]} \omega_{\mathrm{G}}(s), \tag{A.1}$$

and define

$$\omega_{\max}^{\infty}(s) = \sum_{i \in \mathbb{Z}} a_i^+ \cdot \delta_{s_i}(s). \tag{A.2}$$

To obtain the bounds of Theorem D.2 for the compositions (and subsequently strict upper bound for $\delta(\varepsilon)$), the error analysis has to be carried out for the distribution $\omega_{\max}^{\infty}$. To this end, we need bounds for the moment generating functions of $-\omega_{\max}^{\infty}$ and $\omega_{\max}^{\infty}$.

To show that $\omega_{\max}^{\infty}$ indeed leads to an upper bound for $\delta(\varepsilon)$, we refer to (Koskela et al., 2020b), where this is shown for the compositions of the subsampled Gaussian mechanism. The proof here goes analogously, and we have that for all $\varepsilon > 0$,

$$\delta(\varepsilon) \leq \delta_{\max}^{\infty}(\varepsilon),$$

where $\delta_{\max}^{\infty}(\varepsilon)$ is the tight bound for the composition involving $\omega_{\max}^{\infty}$.

To evaluate $\alpha^+(\lambda)$ and $\alpha^-(\lambda)$ for the upper bound of Theorem D.2, we need the moment generating functions of $-\omega_{\max}^{\infty}$ and $\omega_{\max}^{\infty}$. We have the following bound for $\omega_{\max}^{\infty}$. We note that $\mathbb{E}[\mathrm{e}^{\lambda\omega_{\max}}]$ can be evaluated numerically.

Lemma A.1. *Let $0 < \lambda \leq L$ and assume $\sigma \geq 1$ and $\Delta x \leq c \cdot L$, $0 < c < 1$. The moment generating function of $\omega_{\max}^{\infty}$ can be bounded as*

$$\mathbb{E}[\mathrm{e}^{\lambda\omega_{\max}^{\infty}}] \leq \mathbb{E}[\mathrm{e}^{\lambda\omega_{\max}}] + \mathrm{err}(\lambda, L, \sigma),$$

where

$$\mathrm{err}(\lambda, L, \sigma) = \exp\left(\frac{3\lambda}{2\sigma^2}\right)\left(\int_{-\infty}^{-L} \widetilde{\omega}(s)\,\mathrm{d}s + \int_{L-\Delta x}^{\infty} \widetilde{\omega}(s)\,\mathrm{d}s\right), \quad \widetilde{\omega} \sim \mathcal{N}\left(\frac{1+2\lambda}{2\sigma^2}, \frac{1}{\sigma^2}\right). \tag{A.3}$$

Proof. The moment generating function of $\omega_{\max}^{\infty}$ is given by

$$\begin{aligned}\mathbb{E}[\mathrm{e}^{\lambda\omega_{\max}^{\infty}}] &= \int_{-L}^{L} \mathrm{e}^{\lambda s}\omega_{\max}^{\infty}(s)\,\mathrm{d}s + \int_{-\infty}^{-L} \mathrm{e}^{\lambda s}\omega_{\max}^{\infty}(s)\,\mathrm{d}s + \int_{L}^{\infty} \mathrm{e}^{\lambda s}\omega_{\max}^{\infty}(s)\,\mathrm{d}s \\ &\leq \mathbb{E}[\mathrm{e}^{\lambda\omega_{\max}}] + \int_{-\infty}^{-L} \mathrm{e}^{\lambda s}\omega_{\mathrm{G}}(s)\,\mathrm{d}s + \int_{L-\Delta x}^{\infty} \mathrm{e}^{\lambda s}\omega_{\mathrm{G}}(s)\,\mathrm{d}s\end{aligned} \tag{A.4}$$

We arrive at the claim by observing that for $\omega_{\mathrm{G}} \sim \mathcal{N}\left(\frac{1}{2\sigma^2}, \frac{1}{\sigma^2}\right)$,

$$\int_{-\infty}^{-L} \mathrm{e}^{\lambda s}\omega_{\mathrm{G}}(s)\,\mathrm{d}s = \exp\left(\frac{3\lambda}{2\sigma^2}\right)\int_{-\infty}^{-L} \widetilde{\omega}(s)\,\mathrm{d}s,$$

where $\widetilde{\omega} \sim \mathcal{N}\left(\frac{1+2\lambda}{2\sigma^2}, \frac{1}{\sigma^2}\right)$ and similarly for the second term in (A.3). □

Using a reasoning analogous to the proof of Lemma A.1, we get the following. We note that $\mathbb{E}[\mathrm{e}^{-\lambda\omega_{\min}}]$ can be evaluated numerically.

Corollary A.2. *The moment generating function of $-\omega_{\max}^{\infty}$ can be bounded as*

$$\mathbb{E}[\mathrm{e}^{-\lambda\omega_{\max}^{\infty}}] \leq \mathbb{E}[\mathrm{e}^{-\lambda\omega_{\max}}] + \mathrm{err}(\lambda, L, \sigma),$$

where $\mathrm{err}(\lambda, L, \sigma)$ *is defined as in* (A.3)*.*

***Remark* A.3.** *In the experiments, the error term* $\mathrm{err}(\lambda, L, \sigma)$ *was found to be negligible.*

B Privacy Loss Distribution

We here introduce in more detail the basic tool for obtaining tight privacy bounds: the privacy loss distribution (PLD). The results in Subsection B.1 are reformulations of the results given by Meiser and Mohammadi (2018) and Sommer et al. (2019).

B.1 Privacy Loss Distribution Formalism

We consider discrete-valued one-dimensional mechanisms $\mathcal{M}$ which can be seen as mappings from $\mathcal{X}^N$ to the set of discrete-valued random variables. The *generalised probability density functions* of $\mathcal{M}(X)$ and $\mathcal{M}(Y)$, denoted $f_X(t)$ and $f_Y(t)$, respectively, are given by

$$\begin{aligned} f_X(t) &= \sum\nolimits_i a_{X,i} \cdot \delta_{t_{X,i}}(t), \\ f_Y(t) &= \sum\nolimits_i a_{Y,i} \cdot \delta_{t_{Y,i}}(t), \end{aligned} \tag{B.1}$$

where $\delta_t(\cdot)$, $t \in \mathbb{R}$, denotes the Dirac delta function centred at t, and $t_{X,i}, t_{Y,i} \in \mathbb{R}$ and $a_{X,i}, a_{Y,i} \geq 0$. We refer to Koskela et al. (2020b) for more details of the notation. The privacy loss distribution is defined as follows.

Definition B.1. *Let $\mathcal{M} : \mathcal{X}^N \to \mathcal{R}$, $\mathcal{R} \subset \mathbb{R}$, be a discrete-valued randomised mechanism and let $f_X(t)$ and $f_Y(t)$ be probability density functions of the form* (B.1)*. We define the generalised privacy loss distribution (PLD) $\omega_{X/Y}$ as*

$$\omega_{X/Y}(s) = \sum\nolimits_{t_{X,i}=t_{Y,j}} a_{X,i} \cdot \delta_{s_i}(s), \tag{B.2}$$

where $s_i = \log\left(\frac{a_{X,i}}{a_{Y,j}}\right)$.

B.2 Tight (ε, δ)-Bounds for Compositions via PLDs

Let the generalised probability density functions f_X and f_Y of the form (B.1). We define the convolution $f_X * f_Y$ as

$$(f_X * f_Y)(t) = \sum\nolimits_{i,j} a_{X,i}\, a_{Y,j} \cdot \delta_{t_{X,i}+t_{Y,j}}(t).$$

We consider non-adaptive compositions of the form

$$\mathcal{M}(X) = \big(\mathcal{M}_1(X), \ldots, \mathcal{M}_k(X)\big)$$

and we denote by $f_{X,i}(t)$ the density function of $\mathcal{M}_i(X)$ for each i, and by $f_{Y,i}(t)$ that of $\mathcal{M}_i(Y)$. For each i, $1 \leq i \leq k$, we denote the PLD as defined by Def. B.1 and densities $f_{X,i}(t)$ and $f_{Y,i}(t)$ by $\omega_{X/Y,i}$.

The following theorem shows that the tight (ε, δ)-bounds for compositions of non-adaptive heterogeneous mechanisms are obtained using convolutions of PLDs (see also Thm. 1 by Sommer et al. (2019)). A proof is given in the Appendix.

Theorem B.2. *Consider a non-adaptive composition of k independent mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ and neighbouring data sets X and Y. The composition is tightly (ε, δ)-DP for $\delta(\varepsilon)$ given by*

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

where

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

$$\delta_{X/Y,\ell}(\infty) = \sum_{\{t_i\,:\,\mathbb{P}(\mathcal{M}_\ell(X)=t_i)>0,\,\mathbb{P}(\mathcal{M}_\ell(Y)=t_i)=0\}} \mathbb{P}(\mathcal{M}_\ell(X)=t_i) \tag{B.3}$$

and $\omega_{X/Y,1} * \cdots * \omega_{X/Y,k}$ denotes the convolution of the density functions $\omega_{X/Y,\ell}$, $1 \leq \ell \leq k$. $\delta_{Y/X}(\varepsilon)$ can be analogously obtained using the PLDs $\omega_{Y/X,1}, \ldots, \omega_{Y/X,k}$.

Proof. See (Koskela and Honkela, 2021). □

We remark that finding the outputs $\mathcal{M}_i(X)$ and $\mathcal{M}_i(Y)$, $1 \leq i \leq k$, that give the maximal $\delta(\varepsilon)$ is application-specific and has to be carried out individually for each case, similarly as, e.g., in the case of RDP (Mironov, 2017). In the experiments of Section 5 it will be clear how to determine the worst-case distributions $f_{X,i}$ and $f_{Y,i}$.

C Fourier Accountant for Heterogeneous Compositions of Discrete Mechanisms

We next describe in more detail the numerical method for computing tight DP guarantees for heterogeneous compositions of discrete-valued mechanisms. The method is closely related to the homogenous case described in (Koskela et al., 2020b). However, the error analysis is tailored to the heterogeneous case and we consider here also the error induced by the grid approximation.

C.1 Fast Fourier Transform

Let

$$x = [x_0, \ldots, x_{n-1}]^{\mathrm{T}},\ w = [w_0, \ldots, w_{n-1}]^{\mathrm{T}} \in \mathbb{R}^n.$$

The discrete Fourier transform $\mathcal{F}$ and its inverse $\mathcal{F}^{-1}$ are defined as (Stoer and Bulirsch, 2013)

$$\begin{aligned}(\mathcal{F}x)_k &= \sum\nolimits_{j=0}^{n-1} x_j \mathrm{e}^{-\mathrm{i}\,2\pi kj/n},\\ (\mathcal{F}^{-1}w)_k &= \frac{1}{n}\sum\nolimits_{j=0}^{n-1} w_j \mathrm{e}^{\mathrm{i}\,2\pi kj/n},\end{aligned} \tag{C.1}$$

where $\mathrm{i} = \sqrt{-1}$. Using the Fast Fourier Transform (FFT) (Cooley and Tukey, 1965) reduces the running time complexity to $O(n \log n)$. Also, FFT enables evaluating discrete convolutions efficiently. The convolution theorem (Stockham Jr, 1966) states that

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

where $\odot$ denotes the element-wise product and the summation indices are modulo n.

C.2 Grid Approximation

Similarly as in (Koskela et al., 2020b), we place the PLD on a grid

$$X_n = \{x_0, \ldots, x_{n-1}\}, \quad n \in \mathbb{Z}^+, \tag{C.2}$$

where

$$x_i = -L + i\Delta x, \quad \Delta x = 2L/n.$$

Suppose the distribution ω of the PLD is of the form

$$\omega(s) = \sum\nolimits_i a_i \cdot \delta_{s_i}(s), \tag{C.3}$$

where $a_i \geq 0$ and $-L \leq s_i \leq L - \Delta x$ for all i. We define the grid approximations

$$\begin{aligned} \omega^{\mathrm{L}}(s) &:= \sum\nolimits_i a_i \cdot \delta_{s_i^{\mathrm{L}}}(s), \\ \omega^{\mathrm{R}}(s) &:= \sum\nolimits_i a_i \cdot \delta_{s_i^{\mathrm{R}}}(s), \end{aligned} \tag{C.4}$$

where

$$\begin{aligned} s_i^{\mathrm{L}} &= \max\{x \in X_n \,:\, x \leq s_i\}, \\ s_i^{\mathrm{R}} &= \min\{x \in X_n \,:\, x \geq s_i\}. \end{aligned}$$

We note that s_i's correspond to the logarithmic ratios of probabilities of individual events. Thus, often a moderate L is sufficient for the condition

$$-L \leq s_i \leq L - \Delta x$$

to hold for all i. We also provide analysis for the case where this assumption does not hold in the Appendix. This is the case, for example, for the discrete Gaussian distribution (Canonne et al., 2020). From (C.4) we directly get the following result:

Lemma C.1. *Let $\delta(\varepsilon)$ be given by the integral formula of Theorem B.2 for PLDs $\omega_1, \cdots, \omega_k$ of the form* (C.3)*. Let $\delta^{\mathrm{L}}(\varepsilon)$ and $\delta^{\mathrm{R}}(\varepsilon)$ correspondingly be determined by the left and right approximations $\omega_1^{\mathrm{L}}, \ldots, \omega_k^{\mathrm{L}}$ and $\omega_1^{\mathrm{R}}, \ldots, \omega_k^{\mathrm{R}}$, as defined in* (C.4)*. Then for all $\varepsilon > 0$:*

$$\delta^{\mathrm{L}}(\varepsilon) \leq \delta(\varepsilon) \leq \delta^{\mathrm{R}}(\varepsilon).$$

Proof. See (Koskela and Honkela, 2021). □

C.3 Truncation and Periodisation

By truncating convolutions and periodising the PLD distributions we arrive at periodic sums to which the FFT is directly applicable. These operations are analogous to the homogeneous case described in (Koskela et al., 2020b). We describe them next shortly.

Suppose ω_1 and ω_2 are defined such that

$$\omega_1(s) = \sum\nolimits_i a_i \cdot \delta_{s_i}(s), \quad \omega_2(s) = \sum\nolimits_i b_i \cdot \delta_{s_i}(s), \tag{C.5}$$

where for all i: $a_i, b_i \geq 0$ and $s_i = i\Delta x$. The convolution $\omega_1 * \omega_2$ can then be written as

$$\begin{aligned} (\omega_1 * \omega_2)(s) &= \sum\nolimits_{i,j} a_i b_j \cdot \delta_{s_i + s_j}(s) \\ &= \sum\nolimits_i \Big(\sum\nolimits_j a_j b_{i-j} \Big) \cdot \delta_{s_i}(s). \end{aligned} \tag{C.6}$$

Let $L > 0$. We truncate convolutions to the interval $[-L, L]$:

$$\begin{aligned} (\omega_1 * \omega_2)(s) &\approx \sum\nolimits_i \Big(\sum\nolimits_{-L \leq s_j < L} a_j b_{i-j} \Big) \cdot \delta_{s_i}(s) \\ &=: (\omega_1 \circledast \omega_2)(s). \end{aligned}$$

For ω_1 of the form (C.5), we define $\widetilde{\omega}_1$ to be a $2L$-periodic extension of ω_1 from $[-L, L)$ to $\mathbb{R}$, i.e., $\widetilde{\omega}_1$ is of the form

$$\widetilde{\omega}_1(s) = \sum\nolimits_{m \in \mathbb{Z}} \sum\nolimits_i a_i \cdot \delta_{s_i + m \cdot 2L}(s).$$

For ω_1 and ω_2 of the form (C.5), we approximate the convolution $\omega_1 * \omega_2$ as

$$\omega_1 * \omega_2 \approx \widetilde{\omega}_1 \circledast \widetilde{\omega}_2. \tag{C.7}$$

Since ω_1 and ω_2 are defined on an equidistant grid, FFT can be used to evaluate the approximation $\widetilde{\omega}_1 \circledast \widetilde{\omega}_2$ as follows:

Lemma C.2. *Let ω_1 and ω_2 be of the form* (C.5)*, such that $s_i = -L + i\Delta x$, $0 \leq i \leq n-1$, where $L > 0$, n is even and $\Delta x = 2L/n$. Define*

$$\boldsymbol{a} = [a_0 \quad \ldots \quad a_{n-1}]^{\mathrm{T}}, \quad \boldsymbol{b} = [b_0 \quad \ldots \quad b_{n-1}]^{\mathrm{T}}$$

and

$$D = \begin{bmatrix} 0 & I_{n/2} \\ I_{n/2} & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Then,

$$(\widetilde{\omega}_1 \circledast \widetilde{\omega}_2)(s) = \sum\nolimits_{i=0}^{n-1} c_i \cdot \delta_{s_i}(s),$$

where

$$c_i = \left[D\, \mathcal{F}^{-1}\big(\mathcal{F}(D\,\boldsymbol{a}) \odot \mathcal{F}(D\,\boldsymbol{b})\big)\right]_i,$$

and $\odot$ denotes the element-wise product of vectors.

Proof. See (Koskela and Honkela, 2021). □

Since the coefficients of $\widetilde{\omega}_1 \circledast \widetilde{\omega}_2$ are exactly given by the discrete Fourier transform, we are able to analyse the error induced by the FFT approximation by only considering the error of the approximation (C.7).

Algorithm 2 Fourier Accountant Algorithm for Heterogeneous Discrete-Valued Mechanisms

Input: distributions $\omega_1, \ldots, \omega_m$ of the form

$$\omega_j(s) = \sum\nolimits_i a_i^j \cdot \delta_{s_i}(s),$$

$1 \leq j \leq m$, such that $s_i = -L + i\Delta x$, where n is even and, $0 \leq i \leq n-1$, $\Delta x = 2L/n$. Numbers of compositions for each mechanism, $k_1, \ldots, k_m$.

Set

$$\boldsymbol{a}^j = \begin{bmatrix} a_0^j & \ldots & a_{n-1}^j \end{bmatrix}^{\mathrm{T}}, \quad 1 \leq j \leq m.$$

Evaluate the convolutions using Lemma C.2 and FFT.
For each j, $1 \leq j \leq m$, evaluate the FFT:

$$\widetilde{\boldsymbol{a}}^j = \mathcal{F}(D\boldsymbol{a}^j).$$

Compute the element-wise products and apply $\mathcal{F}^{-1}$:

$$\boldsymbol{b} = \left[D\, \mathcal{F}^{-1}\big((\widetilde{\boldsymbol{a}}^1)^{\odot k_1} \odot \cdots \odot (\widetilde{\boldsymbol{a}}^m)^{\odot k_m}\big)\right].$$

Approximate $\delta(\varepsilon)$:

$$\delta(\varepsilon) \approx 1 - \prod_{\ell=1}^{m} (1 - \delta_{X/Y,\ell}(\infty))^{k_\ell} + \sum_{\{\ell\,:\,-L+\ell\Delta x > \varepsilon\}} \left(1 - \mathrm{e}^{\varepsilon - (-L+\ell\Delta x)}\right) b_\ell,$$

where $\delta_{X/Y,\ell}(\infty)$ is defined in Theorem B.2.

C.4 Computing Upper Bounds for $\delta(\varepsilon)$

Given a discrete-valued PLD distribution ω, we get a strict upper $\delta(\varepsilon)$-DP bound as follows. Using parameter values $L > 0$ and $n \in \mathbb{Z}^+$, we form a grid X_n as defined in (C.2) and place each PLDs ω_i, $1 \leq i \leq k$, on X_n to obtain ω_i^{R} as defined in (C.4). We then approximate $\delta^R(\varepsilon)$ using Algorithm 2. We estimate the error incurred by truncation of convolutions periodisation of PLDs using Thm. D.1 (or Thm. D.2 in case the support of ω_i^{R} is no included in the interval $[-L, L]$). By adding this error to the approximation given by Algorithm 2 we obtain a strict upper bound for $\delta(\varepsilon)$.

The error for the truncation of convolutions periodisation of PLDs is given only in terms of the parameter L. The parameter n can then be increased in case the discretisation error bound given by Thm. D.3 is too large.

D DECOMPOSITION OF THE ERROR

When carrying out the approximations for $\delta(\varepsilon)$, we

1. First replace the PLDs $\omega_1, \dots, \omega_k$ by the right grid approximations $\omega_1^{\mathrm{R}}, \dots, \omega_k^{\mathrm{R}}$. Using the notation given above, this corresponds to the approximation $\delta(\varepsilon) \approx \delta^{\mathrm{R}}(\varepsilon)$, i.e. to the approximation
$$\int_{\varepsilon}^{L} (1 - \mathrm{e}^{\varepsilon - s})(\omega_1 * \cdots * \omega_k)(s) \, \mathrm{d}s \approx \int_{\varepsilon}^{L} (1 - \mathrm{e}^{\varepsilon - s})(\omega_1^{\mathrm{R}} * \cdots * \omega_k^{\mathrm{R}})(s) \, \mathrm{d}s.$$
2. Then, Algorithm 2 is used to approximate $\delta^{\mathrm{R}}(\varepsilon) \approx \widetilde{\delta^{\mathrm{R}}}(\varepsilon)$ which in exact arithmetic corresponds to the approximation
$$\int_{\varepsilon}^{L} (1 - \mathrm{e}^{\varepsilon - s})(\omega_1^{\mathrm{R}} * \cdots * \omega_k^{\mathrm{R}})(s) \, \mathrm{d}s \approx \int_{\varepsilon}^{L} (1 - \mathrm{e}^{\varepsilon - s})(\widetilde{\omega}_1^{\mathrm{R}} \circledast \cdots \circledast \widetilde{\omega}_k^{\mathrm{R}})(s) \, \mathrm{d}s,$$
where $\widetilde{\omega}_i$'s denote the periodised PLD distributions and $\circledast$ denotes the truncated convolutions (described above).

We separately consider the errors arising from the periodisation and truncation of the convolutions and from the grid approximation. This means that we bound the total error as
$$\begin{aligned} \left|\delta(\varepsilon) - \widetilde{\delta^{\mathrm{R}}}(\varepsilon)\right| &= \left|\delta(\varepsilon) - \delta^{\mathrm{R}}(\varepsilon) + \delta^{\mathrm{R}}(\varepsilon) - \widetilde{\delta^{\mathrm{R}}}(\varepsilon)\right| \\ &\leq \left|\delta(\varepsilon) - \delta^{\mathrm{R}}(\varepsilon)\right| + \left|\delta^{\mathrm{R}}(\varepsilon) - \widetilde{\delta^{\mathrm{R}}}(\varepsilon)\right| \end{aligned}$$
Theorem D.3 gives a bound for the term $\left|\delta(\varepsilon) - \delta^{\mathrm{R}}(\varepsilon)\right|$ and Theorems D.1 and D.2 of the main text give bounds for the term $\left|\delta(\varepsilon) - \widetilde{\delta}(\varepsilon)\right|$, in terms of the moment generating functions (MGFs) of $\omega_1, \dots, \omega_k$ and $-\omega_1, \dots, -\omega_k$. The bounds for the error $\left|\delta(\varepsilon) - \widetilde{\delta}(\varepsilon)\right|$ can be directly used to bound the error $\left|\delta^{\mathrm{R}}(\varepsilon) - \widetilde{\delta^{\mathrm{R}}}(\varepsilon)\right|$, either by numerically evaluating the MGFs of the PLDs $\omega_1^{\mathrm{R}}, \dots, \omega_k^{\mathrm{R}}$, or by using MGFs of the PLDs $\omega_1, \dots, \omega_k$ and Lemma 7 of (Koskela et al., 2020b), which states that when $0 < \lambda < (\Delta x)^{-1}$,
$$\mathbb{E}[\mathrm{e}^{-\lambda \omega^{\mathrm{R}}}] \leq \mathbb{E}[\mathrm{e}^{-\lambda \omega}] \quad \text{and} \quad \mathbb{E}[\mathrm{e}^{\lambda \omega^{\mathrm{R}}}] \leq \tfrac{1}{1 - \lambda \Delta x} \mathbb{E}[\mathrm{e}^{\lambda \omega}],$$
where $\Delta x = 2L/n$.

D.1 BOUNDING TAILS USING THE CHERNOFF BOUND

We obtain error bounds essentially using the Chernoff bound (Wainwright, 2019)
$$\mathbb{P}[Z \geq t] \leq \frac{\mathbb{E}[\mathrm{e}^{\lambda Z}]}{\mathrm{e}^{\lambda t}}$$
which holds for any random variable Z and all $\lambda > 0$. Suppose $\omega_{X/Y}$ is of the form
$$\omega_{X/Y}(s) = \sum\nolimits_i a_{X,i} \cdot \delta_{s_i}(s), \tag{D.1}$$
where $s_i = \log\left(\frac{a_{X,i}}{a_{Y,i}}\right)$ and $a_{X,i}, a_{Y,i} > 0$. Then, the moment generating function of $\omega_{X/Y}$ is given by
$$\mathbb{E}[\mathrm{e}^{\lambda \omega_{X/Y}}] = \sum\nolimits_i \mathrm{e}^{\lambda s_i} \cdot a_{X,i} = \sum\nolimits_i \left(\frac{a_{X,i}}{a_{Y,i}}\right)^{\lambda} a_{X,i}. \tag{D.2}$$

In our analysis, we repeatedly use the Chernoff bound to bound tails of PLD distributions in terms of pre-computable moment-generating functions. Denote $S_k := \sum_{i=1}^k \omega_i$, where ω_i denotes the PLD random variable of the ith mechanism. If ω_i's are independent, we have that

$$\mathbb{E}[\mathrm{e}^{\lambda S_k}] = \prod_{i=1}^k \mathbb{E}[\mathrm{e}^{\lambda \omega_i}].$$

Then, the Chernoff bound shows that for any $\lambda > 0$

$$\int_L^\infty (\omega_1 * \cdots * \omega_k)(s)\, \mathrm{d}s = \mathbb{P}[S_k \geq L] \leq \prod_{i=1}^k \mathbb{E}[\mathrm{e}^{\lambda \omega_i}]\, \mathrm{e}^{-\lambda L} \leq \mathrm{e}^{\sum_{i=1}^k \alpha_i(\lambda)} \mathrm{e}^{-\lambda L}, \tag{D.3}$$

where $\alpha_i(\lambda) = \log(\mathbb{E}[\mathrm{e}^{\lambda \omega_i}])$.

D.2 Truncation and Periodisation Error

Denote the logarithms of the moment generating functions of the PLDs as

$$\alpha_i^+(\lambda) = \log\Big(\mathbb{E}[\mathrm{e}^{\lambda \omega_i}]\Big), \quad \alpha_i^-(\lambda) = \log\Big(\mathbb{E}[\mathrm{e}^{-\lambda \omega_i}]\Big),$$

where $1 \leq i \leq k$. Futhermore, denote

$$\alpha^+(\lambda) = \sum\nolimits_i \alpha_i^+(\lambda), \quad \alpha^-(\lambda) = \sum\nolimits_i \alpha_i^-(\lambda). \tag{D.4}$$

To obtain $\alpha^+(\lambda)$ and $\alpha^-(\lambda)$, we evaluate the moment generating functions using the finite sum (D.2).

Using the analysis given in the Appendix, we bound the errors arising from the periodisation of the distribution and truncation of the convolutions. As a result, when combining with the Chernoff bound (D.3), we obtain the following two bounds for the total error incurred by Algorithm 2.

Theorem D.1. *Let ω_i's be defined on the grid X_n as described above (i.e., $s_j \in [-L, L - \Delta x]$ for all j). Let $\delta(\varepsilon)$ give the tight (ε, δ)-bound for the PLDs $\omega_1, \ldots, \omega_k$ and let $\widetilde{\delta}(\varepsilon)$ be the result of Algorithm 2. Then, for all $\lambda > 0$*

$$\left|\delta(\varepsilon) - \widetilde{\delta}(\varepsilon)\right| \leq \left(\mathrm{e}^{\alpha^+(\lambda)} + \mathrm{e}^{\alpha^-(\lambda)}\right) \frac{\mathrm{e}^{-L\lambda}}{1 - \mathrm{e}^{-2L\lambda}}.$$

Proof. See (Koskela and Honkela, 2021). □

As s_i's correspond to the logarithmic ratios of probabilities of individual events, often a moderate L is sufficient for $-L \leq s_i \leq L - \Delta x$ to hold for all i. In the Appendix, we prove the following bound which holds also in case s_i's are not inside the interval $[-L, L)$. This happens for example in case of the discrete Gaussian mechanism (Canonne et al., 2020).

Theorem D.2. *Let the PLDs ω_ℓ, $1 \leq \ell \leq k$, take values at the equidistant points $s_i = i\Delta x$. Let $\delta(\varepsilon)$ give the tight (ε, δ)-bound for the PLDs $\omega_1, \ldots, \omega_k$ and let $\widetilde{\delta}(\varepsilon)$ be the result of Algorithm 2. Then, for all $\lambda > 0$*

$$\begin{aligned}\left|\delta(\varepsilon) - \widetilde{\delta}(\varepsilon)\right| \leq \Bigg(&\frac{\mathrm{e}^{(k+1)\max_i \alpha_i^+(\lambda)} - \mathrm{e}^{\max_i \alpha_i^+(\lambda)}}{\mathrm{e}^{\max_i \alpha_i^+(\lambda)} - 1} \\ &+ \frac{\mathrm{e}^{(k+1)\max_i \alpha_i^-(\lambda)} - \mathrm{e}^{\max_i \alpha_i^-(\lambda)}}{\mathrm{e}^{\max_i \alpha_i^-(\lambda)} - 1}\Bigg) \frac{\mathrm{e}^{-L\lambda}}{1 - \mathrm{e}^{-2L\lambda}}.\end{aligned} \tag{D.5}$$

Proof. See (Koskela and Honkela, 2021). □

D.3 BOUND FOR THE DISCRETISATION ERROR

Let $\omega_1, \dots, \omega_k$ be PLD distributions of the form (C.3), i.e., s_i's are not necessarily on the equidistant grid. Denote

$$\omega_\ell(s) = \sum\nolimits_i a_i^\ell \cdot \delta_{s_i^\ell}(s).$$

Denote the right grid approximations (as defined in (C.4))

$$\omega_\ell^{\mathrm{R}}(s) = \sum\nolimits_i a_i^\ell \cdot \delta_{s_i^{\mathrm{R},\ell}}(s)$$

and the tight (ε, δ)-bound corresponding to the PLD $\omega_1^{\mathrm{R}} * \cdots * \omega_k^{\mathrm{R}}$ by $\delta^{\mathrm{R}}(\varepsilon)$. We have the following bound for the error arising from the grid approximation.

Theorem D.3. *Let $\delta(\varepsilon)$ denote the tight (ε, δ)-bound for the convolution PLD $\omega_1 * \cdots * \omega_k$. The discretisation error $\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon)$ can be bounded as*

$$\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) \le k\Delta x \left(\mathbb{P}(\omega_1 + \cdots + \omega_k \ge \varepsilon) - \delta(\varepsilon)\right). \tag{D.6}$$

Proof. From the definition of the discrete convolution we see that

$$(\omega_1 * \cdots * \omega_k)(s) = \sum_{i_1,\dots,i_k} a_{i_1}^1 \cdots a_{i_k}^k \cdot \delta_{s_{i_1}^1 + \dots + s_{i_k}^k}(s)$$

and that

$$\begin{aligned}\delta(\varepsilon) &= \int_\varepsilon^\infty (1 - \mathrm{e}^{\varepsilon - s})\,(\omega_1 * \cdots * \omega_k)(s)\,\mathrm{d}s \\ &= \sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \cdot (1 - \mathrm{e}^{\varepsilon - s_{i_1}^1 - \dots - s_{i_k}^k})\end{aligned}$$

Then, since for $a \le b$:

$$\exp(b) - \exp(a) \le \exp(b)(b - a),$$

we have that

$$\begin{aligned}\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) &= \sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \cdot (\mathrm{e}^{\varepsilon - s_{i_1}^1 - \dots - s_{i_k}^k} - \mathrm{e}^{\varepsilon - s_{i_1}^{\mathrm{R},1} - \dots - s_{i_k}^{\mathrm{R},k}}) \\ &\le \sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \cdot \left((s_{i_1}^{\mathrm{R},1} - s_{i_1}^1) + \dots + (s_{i_k}^{\mathrm{R},k} - s_{i_k}^k)\right) \cdot \mathrm{e}^{\varepsilon - s_{i_1}^1 - \dots - s_{i_k}^k}.\end{aligned}$$

Since

$$s_i^{\mathrm{R},\ell} - s_i^\ell \le \Delta x$$

for all i and ℓ, we have that

$$\begin{aligned}\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) &\le k\Delta x \sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \cdot \mathrm{e}^{\varepsilon - s_{i_1}^1 - \dots - s_{i_k}^k} \\ &= k\Delta x \Big(\sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \\ &\quad - \sum_{\{i_1,\dots,i_k \,:\, s_{i_1}^1 + \dots + s_{i_k}^k \ge \varepsilon\}} a_{i_1}^1 \cdots a_{i_k}^k \cdot (1 - \mathrm{e}^{\varepsilon - s_{i_1}^1 - \dots - s_{i_k}^k}) \Big) \\ &= k\Delta x \left(\mathbb{P}(\omega_1 * \cdots * \omega_k \ge \varepsilon) - \delta(\varepsilon)\right).\end{aligned}$$

□

***Remark* D.4.** *Theorem D.3 instantly gives the bound*

$$\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) \le k\Delta x \left(1 - \delta(\varepsilon)\right) \le k\Delta x. \tag{D.7}$$

On the other hand, the bound (D.6) *and the Chernoff bound* (D.3) *give*

$$\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) \le k\Delta x\, \mathbb{P}(\omega_1 + \cdots + \omega_k \ge \varepsilon) \le k\Delta x\, \mathrm{e}^{\sum_i \alpha_i(\lambda)} \mathrm{e}^{-\lambda\varepsilon} \tag{D.8}$$

which holds for any $\lambda > 0$. By choosing λ appropriately, this leads to a considerably tighter a priori bound than (D.7).

Experimental Illustration. Tables 1 to 3 illustrate the periodisation error bound (D.5) and the discretisation error bound (D.8). We consider the one-dimensional binomial mechanism (Agarwal et al., 2018), where a binomially distributed noise Z with parameters $n \in \mathbb{N}$ and $0 < p < 1$ is added to the output of a query f. Denoting the sensitivity of f by Δ, tight (ε, δ)-bounds are obtained by considering the PLD $\omega_{X/Y}$ given by the distributions f_X and f_Y, where

$$f_X \sim \Delta + \mathrm{Bin}(N, p) \quad \text{and} \quad f_Y \sim \mathrm{Bin}(N, p).$$

We set $N = 1000$, $p = 0.5$, $\Delta = 1$ and $L = 5.0$. We compute logarithmic probabilities using the digamma function and use those to evaluate the values of $\alpha^+(\lambda)$ and $\alpha^+(\lambda)$ required by the error bounds. The error bound (D.5) is evaluated using $\lambda = L$ and for the upper bound (D.8) we take the minimum of the bounds computed with $\lambda \in \{0.5L, 1.0L, 2.0L, 3.0L, 4.0L\}$.

n	error bound (D.8)	$\delta(\varepsilon)$
10^4	$6.31 \cdot 10^{-5}$	$2.59954 \cdot 10^{-5}$
10^5	$6.31 \cdot 10^{-6}$	$2.37864 \cdot 10^{-5}$
10^6	$6.31 \cdot 10^{-7}$	$2.35330 \cdot 10^{-5}$
10^7	$6.31 \cdot 10^{-8}$	$2.35039 \cdot 10^{-5}$

Table 2: The discretisation error bound (D.8) for different values of n when $\varepsilon = 1.0$ and $k = 20$ and the corresponding $\delta(\varepsilon)$-upper bound. The table indicates that the magnitude of the error bound (D.8) is not far from the magnitude of the actual error.

ε	error bound (D.8)	$\delta(\varepsilon)$
0.7	$1.32 \cdot 10^{-6}$	$8.62596 \cdot 10^{-4}$
1.1	$1.79 \cdot 10^{-8}$	$5.66127 \cdot 10^{-6}$
1.5	$3.31 \cdot 10^{-11}$	$6.03580 \cdot 10^{-9}$
1.9	$8.36 \cdot 10^{-15}$	$9.82392 \cdot 10^{-13}$

Table 3: Illustration of the discretisation error bound (D.8) for different values of ε when $n = 10^7$ and $k = 20$ and the corresponding tight $\delta(\varepsilon)$-upper bound. We see that the bound (D.8) stays small in relation to $\delta(\varepsilon)$ as δ decreases.

D.4 Upper Bound for the Computational Complexity

The results by Murtagh and Vadhan (2018) state that there is no algorithm for computing tight (ε, δ)-bounds that would have polynomial complexity in k. However, Theorem 1.7 by Murtagh and Vadhan (2018) states that allowing a small error in the output, the bounds can be evaluated efficiently. More precisely, given a non-adaptive composition of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$, each mechanism $\mathcal{M}_i$ being tightly $(\varepsilon_i, \delta_i)$-DP, the result states that there exists an algorithm that outputs $\widetilde{\varepsilon}(\delta)$ such that

$$\varepsilon(\delta) \leq \widetilde{\varepsilon}(\delta) \leq \varepsilon(\mathrm{e}^{-\eta^2/2} \cdot \delta) + \eta,$$

where $\varepsilon(\delta)$ gives a tight bound for the composition, and the algorithm runs in time

$$\mathcal{O}\left(\frac{k^3 \cdot \overline{\varepsilon} \cdot (1 + \overline{\varepsilon})}{\eta} \log \frac{k^2 \cdot \overline{\varepsilon} \cdot (1 + \overline{\varepsilon})}{\eta}\right), \tag{D.9}$$

where $\overline{\varepsilon} = \frac{1}{k} \sum_i \varepsilon_i$.

Assuming there are $m < k$ distinct mechanisms in the composition, using the error analysis of Sections D.2 and D.3, we obtain a slightly tighter complexity bound for the evaluation of tight δ as a function of ε.

Lemma D.5. *Consider a non-adaptive composition of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ with corresponding worst-case pairs of distributions $f_{X,i}$ and $f_{Y,i}$, $1 \leq i \leq k$. Suppose the sequence $\mathcal{M}_1, \ldots, \mathcal{M}_k$ consists of m distinct mechanisms. Then, it is possible to have an approximation of $\delta(\varepsilon)$ with error less than η with number of operations*

$$\mathcal{O}\left(\frac{2m \cdot k^2 \cdot C_k}{\eta} \log \frac{k^2 \cdot C_k}{\eta}\right),$$

where

$$C_k = \max\{\tfrac{1}{k}\sum_i D_\infty(f_{X,i}||f_{Y,i}), \tfrac{1}{k}\sum_i D_\infty(f_{Y,i}||f_{X,i})\},$$
$$D_\infty(f_X||f_Y) = \sup_{a_{Y,i}\neq 0} \log \frac{a_{X,i}}{a_{Y,i}}$$

and the additional factor in the leading constant is the leading constant of the FFT algorithm.

Proof. We first determine a lower bound for the truncation parameter L in terms of k. Consider the right-hand-side of the error bound in Theorem D.1. Suppose $L \geq 1$ and $\lambda \geq 1$. Then, we have that

$$\begin{aligned}\left(\mathrm{e}^{\alpha^+(\lambda)} + \mathrm{e}^{\alpha^-(\lambda)}\right)\frac{\mathrm{e}^{-L\lambda}}{1-\mathrm{e}^{-2L\lambda}} &\leq \left(\mathrm{e}^{\alpha^+(\lambda)} + \mathrm{e}^{\alpha^-(\lambda)}\right)\cdot\frac{\mathrm{e}}{2}\cdot\mathrm{e}^{-L\lambda},\\ &\leq 2\cdot\mathrm{e}^{\max\{\alpha^-(\lambda),\alpha^+(\lambda)\}}\cdot\frac{\mathrm{e}}{2}\cdot\mathrm{e}^{-L\lambda},\\ &= \mathrm{e}^{\max\{\alpha^-(\lambda),\alpha^+(\lambda)\}+1}\mathrm{e}^{-L\lambda},\end{aligned} \tag{D.10}$$

where $\alpha^-(\lambda)$ and $\alpha^+(\lambda)$ are as given in eq. (D.4).

For each i, the logarithm of the moment-generating function of the PLD can be expressed in terms of the Rényi divergence (Mironov, 2017):

$$\begin{aligned}\log\left(\mathbb{E}[\mathrm{e}^{\lambda\omega_{X/Y,i}}]\right) &= \lambda\cdot\frac{1}{\lambda}\sum_i\left(\frac{a_{X,i}}{a_{Y,i}}\right)^\lambda a_{X,i}\\ &= \lambda\cdot\frac{1}{\lambda}\sum_i\left(\frac{a_{X,i}}{a_{Y,i}}\right)^{\lambda+1} a_{Y,i}\\ &= \lambda\cdot D_{\lambda+1}(f_X||f_Y),\end{aligned}$$

where D_λ denotes the Rényi divergence of order λ. From the monotonicity of Rényi divergence (see Proposition 9, (Mironov, 2017)) it follows that

$$\begin{aligned}\alpha^+(\lambda) &= \lambda\cdot\textstyle\sum_i D_{\lambda+1}(f_{X,i}||f_{Y,i})\\ &\leq \lambda\cdot\textstyle\sum_i D_\infty(f_{X,i}||f_{Y,i}),\end{aligned}$$

where

$$D_\infty(f_X||f_Y) = \sup_{a_{Y,i}\neq 0}\log\frac{a_{X,i}}{a_{Y,i}}.$$

With a similar calculation, we find that

$$\alpha^-(\lambda) \leq (\lambda-1)\cdot\textstyle\sum_i D_\infty(f_{Y,i}||f_{X,i}).$$

Thus,

$$\max\{\alpha^-(\lambda),\alpha^+(\lambda)\} \leq k\lambda\cdot\max\{\tfrac{1}{k}\sum_i D_\infty(f_{X,i}||f_{Y,i}), \tfrac{1}{k}\sum_i D_\infty(f_{Y,i}||f_{X,i})\}.$$

Now we can further bound (D.10) from above as

$$\mathrm{e}^{\max\{\alpha^-(\lambda),\alpha^+(\lambda)\}+1}\mathrm{e}^{-L\lambda} \leq \mathrm{e}^{k\lambda\cdot C_k+1}\mathrm{e}^{-L\lambda},$$

where

$$C_k = \max\{\tfrac{1}{k}\sum_i D_\infty(f_{X,i}||f_{Y,i}), \tfrac{1}{k}\sum_i D_\infty(f_{Y,i}||f_{X,i})\}.$$

Requiring this upper bound to be smaller than a prescribed $\eta > 0$, and setting $\lambda = 1$, we arrive at the condition

$$L \geq k\cdot C_k + 1 + \log\frac{1}{\eta}. \tag{D.11}$$

Next, we bound the computational complexity using a bound for the discretisation error. From Remark D.4 it follows that the discretisation error is bounded as

$$\delta^{\mathrm{R}}(\varepsilon) - \delta(\varepsilon) \leq k\Delta x = \frac{2Lk}{n}.$$

Requiring this discretisation error to be less than η, choosing L according to (D.11) and assuming $k \geq \log \frac{1}{\eta}$, we see that choosing

$$n = \mathcal{O}\left(\frac{k^2 C_k}{\eta}\right)$$

is sufficient for the sum of the error sources to be less than 2η. As we need to compute FFT for m different PLDs, and since FFT has complexity $n \log n$, we see that with

$$\mathcal{O}\left(\frac{2mk^2 C_k}{\eta} \log \frac{k^2 C_k}{\eta}\right)$$

operations it is possible to have an approximation of $\delta(\varepsilon)$ with error less than η, and that additional factor in the leading constant is given by the leading constant in the complexity of FFT.

□

***Remark* D.6.** *We see from the proof, that the periodisation error is less than η for*

$$L \geq \frac{\log \eta^{-1} + \max\{\alpha^-(\lambda), \alpha^+(\lambda)\} + 1}{\lambda}.$$

As this is true for all $\lambda \geq 1$, a minimal sufficient value of L can be found via an optimisation problem w.r.t. λ. Notice also that since $\alpha^-(\lambda)$ and $\alpha^+(\lambda)$ correspond to cumulant generating functions (CGFs) (Abadi et al., 2016) of the compositions, and since the minimisation

$$\min_{\lambda} \frac{\log \delta^{-1} + \alpha(\lambda)}{\lambda}$$

corresponds to the conversion of CGF-values to $(\varepsilon(\delta), \delta)$-DP values (Abadi et al., 2016), we see that approximately (assuming λ^{-1} is small) L has to be chosen as

$$L \geq \varepsilon(\eta),$$

where $\varepsilon(\eta)$ gives (ε, δ)-DP of the composition $(\mathcal{M}_1, \ldots, \mathcal{M}_k)$ at $\delta = \eta$.

***Remark* D.7.** *For simplicity, we have assumed above that the compositions consist of $(\varepsilon, 0)$-DP mechanisms and that the parameter L is chose sufficiently large so that for all i: $|s_i| \leq L$, where $s_i = \log \frac{a_{X,i}}{a_{Y,i}}$. Then, we can bound the periodisation error using Theorem D.1. In case this does not hold, finding a priori conditions for the parameters n and L could also be carried out using Theorem D.2.*

D.5 Fast Evaluation Using the Plancherel Theorem

When using Algorithm 2 to approximate $\delta(\varepsilon)$, we need to evaluate an expression of the form

$$\boldsymbol{b}^k = D\,\mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m}\big) \tag{D.12}$$

and the sum

$$\widetilde{\delta}(\varepsilon) = \sum\nolimits_{-L+\ell\Delta x > \varepsilon} \big(1 - \mathrm{e}^{\varepsilon - (-L+\ell\Delta x)}\big)\, b_\ell^k. \tag{D.13}$$

When evaluating $\widetilde{\delta}(\varepsilon)$ for different numbers of compositions, the inverse transform $\mathcal{F}^{-1}$ is the most expensive part if the vectors $\mathcal{F}(D\boldsymbol{a}^i)$ are precomputed as the element-wise multiplications require $\mathcal{O}(n)$ operations. The following lemma shows that the updates of $\widetilde{\delta}(\varepsilon)$ can actually be performed without using the inverse transform $\mathcal{F}^{-1}$, i.e. using $\mathcal{O}(n)$ operations.

Lemma D.8. *Let $\widetilde{\delta}(\varepsilon)$ be given by* (D.13) *and let $\boldsymbol{b}^k$ be defined as in* (D.12)*. Denote $\boldsymbol{w}_\varepsilon \in \mathbb{R}^n$ such that*

$$(\boldsymbol{w}_\varepsilon)_\ell = \max\{1 - \mathrm{e}^{\varepsilon - (-L+\ell\Delta x)}, 0\}.$$

Then, we have that

$$\widetilde{\delta}(\varepsilon) = \frac{1}{n}\langle \mathcal{F}(D\boldsymbol{w}_\varepsilon), \mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m}\rangle. \tag{D.14}$$

Proof. We see that the sum (D.13) can be written as an inner product: $\widetilde{\delta}(\varepsilon) = \langle \boldsymbol{w}_\varepsilon, \boldsymbol{b}^k \rangle$. The Plancherel Theorem states that DFT (as defined in eq. C.1) preserves inner products: for $x, y \in \mathbb{R}^n$,

$$\langle x, y \rangle = \frac{1}{n} \langle \mathcal{F}x, \mathcal{F}y \rangle. \tag{D.15}$$

Using (D.15) and the fact that D is symmetric, we see that

$$\begin{aligned}
\widetilde{\delta}(\varepsilon) &= \langle \boldsymbol{w}_\varepsilon, \boldsymbol{b}^k \rangle \\
&= \langle \boldsymbol{w}_\varepsilon, D\mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m}\big) \rangle \\
&= \langle D\boldsymbol{w}_\varepsilon, \mathcal{F}^{-1}\big(\mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m}\big) \rangle \\
&= \frac{1}{n} \langle \mathcal{F}(D\boldsymbol{w}), \mathcal{F}(D\boldsymbol{a}^1)^{\odot k_1} \odot \cdots \odot \mathcal{F}(D\boldsymbol{a}^m)^{\odot k_m} \rangle.
\end{aligned}$$

□

We instantly see that if the vectors $\mathcal{F}(D\boldsymbol{w}_\varepsilon)$ and $\mathcal{F}(D\boldsymbol{a}^i)$, $1 \leq i \leq m$, are precomputed, $\widetilde{\delta}(\varepsilon)$ can be updated in $\mathcal{O}(n)$ time. We believe this approach can be used for designing efficient online (ε, δ)-accountants that also give tight guarantees.

Experimental Illustration. Consider computing tight $\delta(\varepsilon)$-bound for the subsampled Gaussian mechanism (see Section 5.2), for $q = 0.02$ and $\sigma = 2.0$. We evaluate $\delta(\varepsilon)$ after $k = 100, 200, \ldots, 500$ compositions at $\varepsilon = 1.0$. Table 4 illustrates the compute time for each update of $\delta(\varepsilon)$, using a) a pre-computed vector $\mathcal{F}(D\boldsymbol{a})$, the inverse transform $\mathcal{F}^{-1}$ and the summation (D.13) and b) pre-computed vectors $\mathcal{F}(D\boldsymbol{a})^{\odot 100}$ and $\mathcal{F}(D\boldsymbol{w}_\varepsilon)$ and the inner product (D.14).

n	t (ms) (D.13)	t (ms) (D.14)	$\delta(\varepsilon)$
$5 \cdot 10^4$	5.8	0.18	$2.900925 \cdot 10^{-6}$
$1 \cdot 10^5$	12	0.36	$2.851835 \cdot 10^{-6}$
$1 \cdot 10^6$	140	5.1	$2.846942 \cdot 10^{-6}$
$5 \cdot 10^6$	750	30	$2.846941 \cdot 10^{-6}$

Table 4: Compute times (in milliseconds) for an update of $\delta(\varepsilon)$-bound for different values of n using the summation (D.13) and the inner product (D.14) and the $\delta(\varepsilon)$-upper bound after $k = 500$ compositions. We see that using Lemma D.8 we can speed up the update more than 20-fold, and that accurate update of $\delta(\varepsilon)$ is possible in less than one millisecond. The alternatives (D.13) and (D.14) give equal results up to machine precision.