Privacy Loss Distributions

Differential Privacy Team
Google

March 31, 2025

This document is a supplementary material for the implementation of the algorithms for building and manipulating privacy loss distributions in the privacy accounting library.

1 Notation and Preliminaries

Discrete Distributions. For a discrete distribution $\mathcal{D}$, when there is no ambiguity, we abbreviate $\Pr_{X\sim\mathcal{D}}[X = x]$ as $\mathcal{D}(x)$. For two discrete distributions μ and μ', we use $\mathfrak{D}_{e^\varepsilon}(\mu||\mu')$ to denote their *ε-hockey stick divergence*, i.e.,

$$\mathfrak{D}_{e^\varepsilon}(\mu||\mu') := \sum_{y\in\mathrm{supp}(\mu)} [\mu(y) - e^\varepsilon \cdot \mu'(y)]_+,$$

where $[x]_+$ denotes $\max\{x, 0\}$.

Continuous Distributions. For a continuous distribution $\mathcal{D}$, we use $f_\mathcal{D}(\cdot)$ to denote its probability density function. For two continuous distributions μ and μ', their ε-hockey stick divergence is defined as

$$\mathfrak{D}_{e^\varepsilon}(\mu||\mu') := \int [f_\mu(y) - e^\varepsilon \cdot f_{\mu'}(y)]_+ dy.$$

For two (discrete or continuous) distributions $\mathcal{D}$ and $\mathcal{D}'$, we use $\mathcal{D}\otimes\mathcal{D}'$ to denote the product distribution of $\mathcal{D}$ and $\mathcal{D}'$. Furthermore, for $\alpha, \alpha' \geq 0$ such that $\alpha + \alpha' = 1$, we use $\alpha\mathcal{D} + \alpha'\mathcal{D}'$ to denote the mixture distribution that returns a sample from $\mathcal{D}$ with probability α or a sample from $\mathcal{D}'$ with probability α'. On the other hand, for a real number $k \in \mathbb{R}$, we use $k + \mathcal{D}$ to denote distribution of $k + X$ for $X \sim \mathcal{D}$.

Differential Privacy. We view datasets $\boldsymbol{x}$ as a sequence of records. For a mechanism $\mathcal{M}$ and an input dataset $\boldsymbol{x}$, we use $\mathcal{M}(\boldsymbol{x})$ to denote the distribution of the output. The definition of differential privacy hinges on the notion of neighboring relation over datasets. We consider the two typical neighboring relations studied, namely two datasets $\boldsymbol{x}$ and $\boldsymbol{x}'$ are said to be:

- *add-remove neighbors*, denoted $\boldsymbol{x} \simeq_{ar} \boldsymbol{x}'$, if one can be obtained from the other by adding or removing a record from the other, and
- *substitution neighbors*, denoted $\boldsymbol{x} \simeq_s \boldsymbol{x}'$, if one can be obtained from the other by substituting a record in the other.

Since the add-remove neighboring relation is asymmetric, we will use $\boldsymbol{x} \to_r \boldsymbol{x}'$ to denote that $\boldsymbol{x}'$ can be obtained by removing a record from $\boldsymbol{x}$; so, $\boldsymbol{x} \simeq_{ar} \boldsymbol{x}'$ if $\boldsymbol{x} \to_r \boldsymbol{x}'$ or $\boldsymbol{x}' \to_r \boldsymbol{x}$.

The standard definition of differential privacy [DMNS06, DKM+06] may be rephrased as follows, for any neighboring relation.

Observation 1. *A mechanism $\mathcal{M}$ is (ε, δ)-differentially private (or (ε, δ)-DP for short) if and only if, for any neighboring input datasets $\boldsymbol{x} \simeq \boldsymbol{x}'$, it holds that $\mathfrak{D}_{e^\varepsilon}(\mathcal{M}(\boldsymbol{x}) || \mathcal{M}(\boldsymbol{x}')) \le \delta$.*

2 Privacy Loss Distribution

A notion that will be useful to us is the so-called *Privacy Loss Distribution (PLD)* defined in [DR16]. Here we will mostly follow the notations from [MM18, SMM19, KJH20, KJPH20], from which most of the results we use follow.

Definition 1. *For two discrete distributions μ_{up} and μ_{lo}, their* privacy loss *at $o \in \mathrm{supp}(\mu_{\mathrm{up}})$ is defined as*

$$\mathcal{L}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}(o) := \begin{cases} \ln\left(\frac{\mu_{\mathrm{up}}(o)}{\mu_{\mathrm{lo}}(o)}\right) & \text{if } \mu_{\mathrm{lo}}(o) > 0 \\ +\infty & \text{if } \mu_{\mathrm{lo}}(o) = 0 \end{cases}.$$

The privacy loss distribution (PLD) *of μ_{up} and μ_{lo}, denoted by $PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$, is a distribution on $\mathbb{R} \cup \{\infty\}$ where $y \sim PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ is generated as follows: sample $o \sim \mu_{\mathrm{up}}$ and let $y = \mathcal{L}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}(o)$.*

For two continuous distributions μ_{up} and μ_{lo}, their privacy loss at o is defined as

$$\mathcal{L}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}(o) := \begin{cases} \ln\left(\frac{f_{\mu_{\mathrm{up}}}(o)}{f_{\mu_{\mathrm{lo}}}(o)}\right) & \text{if } f_{\mu_{\mathrm{lo}}}(o) > 0 \\ +\infty & \text{if } f_{\mu_{\mathrm{lo}}}(o) = 0 \end{cases}.$$

The PLD of μ_{up} and μ_{lo}, denoted by $PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$, is again a distribution on $\mathbb{R} \cup \{\infty\}$ where $y \sim PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ is generated by picking $o \sim \mu_{\mathrm{up}}$ and then letting $y = PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}(o)$.

For a mechanism $\mathcal{M}$ and two input vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, the privacy loss distribution (PLD) between $\boldsymbol{x}$ and $\boldsymbol{x}'$ is defined as $PLD_{\mathcal{M}(\boldsymbol{x})/\mathcal{M}(\boldsymbol{x}')}$.

The main observation that makes PLD useful is that it allows one to calculate the ε-hockey stick divergence between the two distributions, or equivalently to check whether a mechanism is (ε, δ)-DP.

Observation 2 ([SMM19, KJH20])**.** *For any two distributions μ_{up} and μ_{lo} where both are discrete or both are continuous, it holds that*

$$\mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{up}} || \mu_{\mathrm{lo}}) = \mathbb{E}_{y \sim PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}}[1 - e^{\varepsilon - y}]_+.$$

Due to Observation 1, a mechanism $\mathcal{M}$ is (ε, δ)-DP if and only if the following holds for all neighboring input datasets $\boldsymbol{x}$ and $\boldsymbol{x}'$:

$$\delta \ge \mathbb{E}_{y \sim PLD_{\mathcal{M}(\boldsymbol{x})/\mathcal{M}(\boldsymbol{x}')}}[1 - e^{\varepsilon - y}]_+.$$

For convenience, we may write $\mathfrak{D}_{e^\varepsilon}(\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}})$ instead of $\mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{up}} || \mu_{\mathrm{lo}})$.

Another observation is that PLD is very compatible with composition of mechanisms. When the composition is non-adaptive, i.e., when mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ are run independently, the output distribution on input vector $\boldsymbol{x}$ is simply the product distribution $\mathcal{M}_1(\boldsymbol{x}) \otimes \mathcal{M}_2(\boldsymbol{x})$. The observation here is that the PLD of the product distribution is simply the *convolution* of the two PLDs. We state that formally and also recall the definition of convolution below.

Definition 2. *Let μ and μ' be any distributions on real numbers. Their convolution, denoted by $\mu * \mu'$, is a distribution on real numbers where a sample $t \sim \mu * \mu'$ is drawn by first independently sampling $a \sim \mu$, $a' \sim \mu'$ and then letting $t = a + a'$.*

Observation 3 ([SMM19]). *Let $\mu_{\mathrm{up}}, \mu'_{\mathrm{up}}, \mu_{\mathrm{lo}}$ and μ'_{lo} be any distributions such that all of them are discrete or all are continuous. Then, we have*

$$PLD_{(\mu_{\mathrm{up}} \otimes \mu'_{\mathrm{up}})/(\mu_{\mathrm{lo}} \otimes \mu'_{\mathrm{lo}})} = PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}} * PLD_{\mu'_{\mathrm{up}}/\mu'_{\mathrm{lo}}}.$$

For any mechanism $\mathcal{M}$, it is helpful to consider the notion of a *dominating* and a *worst-case* PLD, defined as follows.

Definition 3 (Defnition 7 in [ZDW22]). *$PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ is said to be a* dominating *PLD for a mechanism $\mathcal{M}$*

- *under add-remove neighboring relation, if for all $\varepsilon \in \mathbb{R}$, it holds that,*

$$\sup_{\boldsymbol{x} \to_r \boldsymbol{x}'} \mathfrak{D}_{e^\varepsilon}(\mathcal{M}(\boldsymbol{x}) || \mathcal{M}(\boldsymbol{x}')) \leq \mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{up}} || \mu_{\mathrm{lo}}), \text{ and}$$

- *under substitution neighboring relation, if for all $\varepsilon \in \mathbb{R}$, it holds that,*

$$\sup_{\boldsymbol{x} \simeq_s \boldsymbol{x}'} \mathfrak{D}_{e^\varepsilon}(\mathcal{M}(\boldsymbol{x}) || \mathcal{M}(\boldsymbol{x}')) \leq \mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{up}} || \mu_{\mathrm{lo}}).$$

A dominating $PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ is said to be a worst case *PLD if there exists adjacent $\boldsymbol{x} \to_r \boldsymbol{x}'$ (or $\boldsymbol{x} \simeq_s \boldsymbol{x}'$) such that $PLD_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}} = PLD_{\mathcal{M}(D)/\mathcal{M}(D')}$.*[1]

A worst-case PLD for a mechanism gives rise to a tight characterization of its privacy loss. Namely, if $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ is a dominating PLD for mechanism $\mathcal{M}$ under add-remove neighboring relation, then $\mathcal{M}$ satisfies (ε, δ)-DP for $\delta = \max\{\mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{up}} || \mu_{\mathrm{lo}}), \mathfrak{D}_{e^\varepsilon}(\mu_{\mathrm{lo}} || \mu_{\mathrm{up}})\}$.

2.1 Composition via Privacy Loss Buckets

Observations 2 and 3 provide a way to compute the privacy parameters for compositions of multiple mechanisms: first we calculate the dominating PLD of each mechanism, compute their convolutions, and finally compute the ε-hockey stick divergence of their convolution. An issue here is that the trivial implementation of this algorithm is not efficient; for instance, PLD itself can be a continuous distribution which cannot be represented finitely. Another consideration is that the convolution of multiple PLDs may blow up the support size. (That is, if we compose k mechanisms each with PLD support size n, then the resulting PLD may have support as large as n^k.)

[1] We depart slightly from the original definition in [ZDW22], which does not distinguish between $\boldsymbol{x} \to_r \boldsymbol{x}'$ and $\boldsymbol{x}' \to_r \boldsymbol{x}$, as this notation is more convenient for us.

This brings us to an algorithm of Meiser and Mohammadi [MM18] called *Privacy Buckets*. This simple algorithm allows us to "approximate" PLDs in such a way that the convolution is efficient, and still gives good numerical approximation for privacy parameters. As its name suggest, privacy buckets rounds the value of the PLD into buckets, which are integer multiples of a chosen positive real number (called `value_discretization_interval` in our implementation). The point here is that, once the values are integer multiples of such a number, we may use (inverse) Fast Fourier Transform (FFT) to quickly compute the convolution. (The idea of using FFT has been suggested by [KJH20, KJPH20].) We basically implement this; there are several subtleties in the implementation, which are listed below:

- We separately account for the "infinity mass", i.e., $\Pr_{o\sim\mu_{\mathrm{up}}}[\mu_{\mathrm{lo}}(o) = 0]$.
- Our code allows one to compute both the "pessimistic" (i.e., safe) and "optimistic" estimates of the hockey stick divergence between μ_{up} and μ_{lo}. In the former case, the PLD values are rounded up. In the latter case, the PLD values are rounded down. The pessimistic estimate results in a larger δ value than the true value, whereas the optimistic estimate results in a smaller δ than the true value.
- To make the implementation efficient, we also make sure that the support is not too large. This is done by truncating a tail mass of μ_{up} that is smaller than a certain threshold. For the pessimistic case, this mass is accounted in `infinity_mass`. For the optimistic case, this mass is discarded.

2.2 Tighter discrete approximation via Connect-the-Dots

A different algorithm by Doroshenko et al. [DGK+22] called *Connect the Dots* gives tighter discrete approximations of PLDs, building on Definition 3. The library currently supports the pessimistic connect-the-dots approximation method given in the above paper.

We describe the final construction of the pessimistic estimate of the PLD below, deferring the ideas and proofs behind the algorithm to the paper.

Suppose we want a discrete approximation of a given PLD $= \mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ that is supported on a given set of epsilon values $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n\}$, in addition to $+\infty$. Let $\varepsilon_0 = -\infty$ for ease of notation and let $\delta_i = \mathfrak{D}_{e^{\varepsilon_i}}(\mu_{\mathrm{up}}||\mu_{\mathrm{lo}})$ (note: $\delta_0 = 1$). The connect-the-dots pessimistic approximation $\mathrm{PLD}^{\uparrow}$ of PLD is given as follows.

$$\begin{aligned}
\Pr[\mathrm{PLD}^{\uparrow} = \varepsilon_i] &= \frac{\delta_i - \delta_{i-1}}{1 - e^{\varepsilon_{i-1}-\varepsilon_i}} - \frac{\delta_{i+1} - \delta_i}{e^{\varepsilon_{i+1}-\varepsilon_i} - 1} \qquad \text{for } 1 \le i \le n-1 \\
\Pr[\mathrm{PLD}^{\uparrow} = \varepsilon_n] &= \frac{\delta_n - \delta_{n-1}}{1 - e^{\varepsilon_{n-1}-\varepsilon_n}} \\
\Pr[\mathrm{PLD}^{\uparrow} = +\infty] &= \delta_n
\end{aligned}$$

Note that for any PLD $= \mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$, all the probability masses computed above are guaranteed to be non-negative and sum to 1 because the function $h : \mathbb{R}_{\ge 0} \to [0,1]$ defined as $h(e^{\varepsilon}) = \mathfrak{D}_{e^{\varepsilon}}(\mu_{\mathrm{up}}||\mu_{\mathrm{lo}})$ is non-increasing, convex and satisfies $h(0) = 1$ and $h(e^{\varepsilon}) \ge \max\{0, 1 - e^{\varepsilon}\}$. Refer to [DGK+22] for more details.

3 PLDs of Specific Mechanisms

In this section, we calculate the privacy loss distributions for several well-known mechanisms. Throughout this section, we only consider a scalar-valued function f, and only consider the add-remove neighboring relation. Recall that its sensitivity is defined as $\Delta(f) := \max_{\boldsymbol{x} \simeq_{ar} \boldsymbol{x}'} |f(\boldsymbol{x}) - f(\boldsymbol{x}')|$ where the maximum is over two add-remove neighboring datasets $\boldsymbol{x}$ and $\boldsymbol{x}'$. In case of substitution neighboring relation, we adopt the convention that $\Delta_{\text{sub}}(f) = 2\Delta_{\text{add-remove}}(f)$; note that this is not always the case for all mechanisms, so we advise paying special attention to setting the sensitivity parameter under substitution neighboring relation.

3.1 Laplace Mechanism

The Laplace mechanism [DMNS06] simply outputs $f(x) + \text{Lap}(0, b)$ where $\text{Lap}(\mu, b)$ is the Laplace random variable with mean μ and scale parameter b; its probability density function at point x is equal to $\frac{1}{2b} \cdot e^{-|x-\mu|/b}$.

In this case, the worst-case PLD of the mechanism is $\text{PLD}_{\mu_{\text{up}}/\mu_{\text{lo}}}$ where $\mu_{\text{up}} = \text{Lap}(0, b)$ and $\mu_{\text{lo}} = \text{Lap}(\Delta(f), b)$. This is equivalent to choosing $\mu_{\text{up}} = \text{Lap}(0, 1)$ and $\mu_{\text{lo}} = \text{Lap}(\tilde{\Delta}, 1)$, where $\tilde{\Delta} := \Delta(f)/b$. That is, the privacy loss variable is generated by first picking $x \sim \text{Lap}(0, 1)$ and letting the privacy loss be

$$\ln\left(\frac{\frac{1}{2} \cdot e^{-|x|}}{\frac{1}{2} \cdot e^{-|x-\tilde{\Delta}|}}\right) = |x - \tilde{\Delta}| - |x| = \begin{cases} \tilde{\Delta} & \text{if } x \le 0, \\ -\tilde{\Delta} & \text{if } x \ge \tilde{\Delta}, \\ \tilde{\Delta} - 2x & \text{if } 0 < x < \tilde{\Delta}. \end{cases}$$

3.2 Gaussian Mechanism

The Gaussian mechanism (see [BW18] and the references therein) simply outputs $f(x) + \mathcal{N}(0, \sigma^2)$ where $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian random variable with mean μ and standard deviation σ; its probability density function at point x is equal to $\frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Here, the worst-case PLD of the mechanism is $\text{PLD}_{\mu_{\text{up}}/\mu_{\text{lo}}}$ where $\mu_{\text{up}} = \mathcal{N}(0, \sigma^2)$ and $\mu_{\text{lo}} = \mathcal{N}(\Delta(f), \sigma^2)$. This is equivalent to choosing $\mu_{\text{up}} = \mathcal{N}(0, 1)$ and $\mu_{\text{lo}} = \mathcal{N}(\tilde{\Delta}, 1)$ where $\tilde{\Delta} := \Delta(f)/\sigma$. That is, the privacy loss variable is generated by first picking $x \sim \mathcal{N}(0, 1)$ and letting the privacy loss be

$$\ln\left(\frac{\frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-x^2/2}}{\frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-(x-\tilde{\Delta})^2/2}}\right) = \frac{\tilde{\Delta}}{2} \cdot \left(\tilde{\Delta} - 2x\right).$$

3.2.1 Calculating ε-hockey stick divergence of Gaussian Mechanism

The ε-hockey stick divergence between $\mathcal{N}(0, \sigma^2)$ and $\mathcal{N}(\Delta(f), \sigma^2)$ is equal to the ε-hockey stick divergence between $\mathcal{N}(0, 1)$ and $\mathcal{N}(\tilde{\Delta}, 1)$. Let ϕ and Φ denote the PDF and CDF of the standard normal distribution respectively. We can write

$$\mathfrak{D}_{e^\varepsilon}(\mathcal{N}(0, 1) \| \mathcal{N}(\tilde{\Delta}, 1)) = \int_{-\infty}^{\infty} [\phi(x) - e^\varepsilon \cdot \phi(x - \tilde{\Delta})]_+ dx.$$

Now, as stated above, $\frac{\phi(x)}{\phi(x-\tilde{\Delta})} = e^{\frac{\tilde{\Delta}}{2}\cdot(\tilde{\Delta}-2x)}$, which is greater than e^{ε} if and only if $x < x_{upper} := 0.5\tilde{\Delta} - \varepsilon/\tilde{\Delta}$. As a result, we have

$$\begin{aligned}\mathfrak{D}_{e^{\varepsilon}}(\mathcal{N}(0,1)\|\mathcal{N}(\tilde{\Delta},1)) &= \int_{-\infty}^{x_{upper}} \left(\phi(x) - e^{\varepsilon}\cdot\phi(x-\tilde{\Delta})\right)dx. \\ &= \Phi(x_{upper}) - e^{\varepsilon}\cdot\Phi(x_{upper}-\tilde{\Delta}). \end{aligned} \tag{1}$$

3.3 Mixture of Gaussians Mechanism

The Mixture of Gaussians Mechanism [CGST24] captures a Gaussian mechanism where $f(x)$ and its sensitivity are random. For example, f could be a counting query on a random subset of the data, and if two adjacent databases can differ in up to k examples, the sensitivity would be a random variable with support $\{0, 1, \ldots, k\}$.

Fixing $\sigma = 1$ by rescaling, if the sensitivity is c_i with probability p_i, then the worst-case PLD of the mechanism is $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ where $\mu_{\mathrm{up}} = \sum_i p_i \cdot \mathcal{N}(c_i, 1)$ and $\mu_{\mathrm{lo}} = \mathcal{N}(0,1)$; note that this PLD is asymmetric, and thus, requires maintaining both $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ and $\mathrm{PLD}_{\mu_{\mathrm{lo}}/\mu_{\mathrm{up}}}$.

For $\mathrm{PLD}_{\mu_{\mathrm{lo}}/\mu_{\mathrm{up}}}$, the privacy loss variable is generated by sampling $x \sim N(0,1)$ and letting the privacy loss be

$$\ln\left(\frac{\frac{1}{\sqrt{2\pi}}\cdot e^{-x^2/2}}{\sum_i p_i \frac{1}{\sqrt{2\pi}}\cdot e^{-(x-c_i)^2/2}}\right) = -\ln\left(\sum_i p_i \cdot e^{-c_i^2/2 + c_i x}\right).$$

For $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$, the privacy loss variable is generated by sampling i with probability p_i, sampling $x \sim N(c_i, 1)$ and letting the privacy loss be

$$\ln\left(\frac{\sum_i p_i \frac{1}{\sqrt{2\pi}}\cdot e^{-(x-c_i)^2/2}}{\frac{1}{\sqrt{2\pi}}\cdot e^{-x^2/2}}\right) = \ln\left(\sum_i p_i \cdot e^{-c_i^2/2 + c_i x}\right).$$

3.4 Discrete Laplace Mechanism

The Discrete Laplace Mechanism (also known as the Symmetric Geometric Mechanism; e.g., see [GRS12]) outputs $f(\boldsymbol{x}) + \mathrm{DLap}(0, a)$ where $\mathrm{DLap}(\mu, a)$ is the Discrete Laplace distribution with mean μ and inverse-scale parameter a; its probability mass function at $x \in \mathbb{Z}$ is $\frac{e^a - 1}{e^a + 1}\cdot e^{-a|x-\mu|}$. (For simplicity, we assume that the image of f is a subset of the integers.)

In this case, the worst-case PLD of the mechanism is $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ where $\mu_{\mathrm{up}} = \mathrm{DLap}(0, a)$ and $\mu_{\mathrm{lo}} = \mathrm{DLap}(\Delta(f), a)$. That is, the privacy loss variable is generated by first picking $x \sim \mathrm{DLap}(0, a)$ and letting the privacy loss be

$$\ln\left(\frac{\frac{e^a-1}{e^a+1}\cdot e^{-a|x|}}{\frac{e^a-1}{e^a+1}\cdot e^{-a|x-\Delta(f)|}}\right) = a(|x-\Delta(f)| - |x|) = \begin{cases} a\cdot\Delta(f) & \text{if } x \le 0, \\ -a\cdot\Delta(f) & \text{if } x \ge \Delta(f), \\ a(\Delta(f) - 2x) & \text{if } 0 < x < \Delta(f). \end{cases}$$

3.5 (Truncated) Discrete Gaussian Mechanism

The Discrete Gaussian Mechanism [CKS20] adds a noise supported on the integers such that the probability mass function at x is proportional to $e^{\frac{-x^2}{2\sigma^2}}$, where σ is the parameter of the distribution (unlike the continuous case, σ here is *not* equal to the standard deviation). Due to technical reasons, we truncate the noise so that it is supported in $\{-\tau, \dots, \tau\}$. That is, the mechanism outputs $f(x) + \mathcal{N}_{\mathbb{Z}}^{\tau}(0, \sigma^2)$, where $\mathcal{N}_{\mathbb{Z}}^{\tau}(\mu, \sigma^2)$ has probability mass proportional to $e^{\frac{-(x-\mu)^2}{2\sigma^2}}$ for integer $x \in \{\mu - \tau, \dots, \mu + \tau\}$ (defined only for $\mu, \tau \in \mathbb{Z}$ and $\tau > 0$).

The worst-case PLD of this mechanism is $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$ where $\mu_{\mathrm{up}} = \mathcal{N}_{\mathbb{Z}}^{\tau}(0, \sigma^2)$ and $\mu_{\mathrm{lo}} = \mathcal{N}_{\mathbb{Z}}^{\tau}(\Delta(f), \sigma^2)$. That is, the privacy loss variable is generated by first picking $x \sim \mathcal{N}_{\mathbb{Z}}^{\tau}(0, \sigma^2)$ and letting the privacy loss be

$$\begin{cases} \ln\left(\frac{e^{\frac{-x^2}{2\sigma^2}}}{e^{\frac{-(x-\Delta)^2}{2\sigma^2}}}\right) = \frac{\Delta}{2\sigma^2} \cdot (\Delta - 2x) & \text{if } -\tau + \Delta(f) \le x \le \tau, \\ \infty & \text{if } -\tau \le x < -\tau + \Delta(f) \end{cases}$$

To deal with the possible probability mass difference due to truncation, we use the following tail bound [CKS20, Proposition 25]:

Proposition 1. *For any $\tau \in \mathbb{N}$ and any $\sigma > 0$,*

$$\Pr_{X \sim \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)}[|X| \ge \tau + 1] \le \Pr_{X \sim \mathcal{N}(0,\sigma^2)}[|X| \ge \tau].$$

3.6 k-Randomized Response

In the k-Randomized Response [War65], the input is one of k values. The protocol outputs the input with probability $1 - p$. With the remaining probability p, the protocol outputs a uniformly random element from the k possible values (including the input itself).

Let $\mathcal{R}_k$ denote the randomized response. This mechanism is typically analyzed under the substitution neighboring relation, in which case, the PLD of the mechanism is equal to $\mathrm{PLD}_{\mathcal{R}_k(x)/\mathcal{R}_k(x')}$ where x and x' are two distinct inputs. That is, the privacy loss variable is generated by first picking $o \sim \mathcal{R}_k(x)$ and letting the privacy loss be

$$\ln\left(\frac{\Pr[\mathcal{R}_k(x) = o]}{\Pr[\mathcal{R}_k(x') = o]}\right) = \begin{cases} \ln\left(\frac{k(1-p)+p}{p}\right) & \text{if } o = x, \\ \ln\left(\frac{p}{k(1-p)+p}\right) & \text{if } o = x', \\ 0 & \text{if } o \notin \{x, x'\}. \end{cases}$$

In other words, the privacy loss random variable is equal to

$$\begin{cases} \ln\left(\frac{k(1-p)+p}{p}\right) & \text{with probability } 1 - p + \frac{p}{k}, \\ \ln\left(\frac{p}{k(1-p)+p}\right) & \text{with probability } \frac{p}{k}, \\ 0 & \text{with probability } \frac{p(k-2)}{k}. \end{cases}$$

However, it is also possible to analyze $\mathcal{R}_k$ under "Replace Special" neighboring relation, wherein we want the DP property to hold against replacing the true input x with a special symbol $\perp$ and we extend $\mathcal{R}_k$ to output a uniformly random value among the possible k

values on input $\perp$. In this case, the PLD of the mechanism is equal to the PLD between $\mathcal{R}_k(x)$ and $\mathcal{R}_k(\perp)$, which results in the following privacy loss variable for the adjacency $x \to \perp$ (we refer to this as "Remove" for convenience):

$$\begin{cases} \ln(k(1-p)+p) & \text{with probability } 1-p+\frac{p}{k}, \\ \ln(p) & \text{with probability } \frac{p(k-1)}{k}, \end{cases}$$

and the following privacy loss variable for the adjacency $\perp \to x$ (we refer to this as "Add" for convenience):

$$\begin{cases} -\ln(k(1-p)+p) & \text{with probability } \frac{1}{k}, \\ -\ln(p) & \text{with probability } \frac{(k-1)}{k}. \end{cases}$$

3.7 Pessimistic PLD for (ε, δ)-DP Algorithms

In some scenarios, we may not know the specific algorithm being applied or it may be hard to write down the PLD exactly, but we do know that the algorithm is (ε, δ)-DP. In this case, it is possible to define a dominating PLD, which is a pessimistic estimate of the true PLD. Specifically, [KOV15] proves the following:[2]

Theorem 1. *For any (ε, δ)-DP mechanism $\mathcal{M}$ and neighboring input datasets $\boldsymbol{x}, \boldsymbol{x}'$, Let $\mathcal{M}^*$ be the following mechanism:*

$$\begin{aligned} &\Pr[\mathcal{M}^*(\boldsymbol{x}) = 0] = \delta, && \Pr[\mathcal{M}^*(\boldsymbol{x}) = 0] = 0, \\ &\Pr[\mathcal{M}^*(\boldsymbol{x}) = 1] = (1-\delta) \cdot \frac{e^\varepsilon}{1+e^\varepsilon}, && \Pr[\mathcal{M}^*(\boldsymbol{x}) = 1] = (1-\delta) \cdot \frac{1}{1+e^\varepsilon}, \\ &\Pr[\mathcal{M}^*(\boldsymbol{x}) = 2] = (1-\delta) \cdot \frac{1}{1+e^\varepsilon}, && \Pr[\mathcal{M}^*(\boldsymbol{x}) = 2] = (1-\delta) \cdot \frac{e^\varepsilon}{1+e^\varepsilon}, \\ &\Pr[\mathcal{M}^*(\boldsymbol{x}) = 3] = 0, && \Pr[\mathcal{M}^*(\boldsymbol{x}) = 3] = \delta. \end{aligned}$$

Then, there exists a transformation T such that $T(\mathcal{M}^*(\boldsymbol{x}))$ and $T(\mathcal{M}^*(\boldsymbol{x}'))$ are identically distributed as $\mathcal{M}(\boldsymbol{x})$ and $\mathcal{M}(\boldsymbol{x}')$ respectively.

By post-processing property of differential privacy, the above theorem means that $\mathcal{M}$ is more private than $\mathcal{M}^*$. As such, we may use the PLD of $\mathcal{M}^*$ as a pessimistic estimate of the PLD of $\mathcal{M}$. The privacy loss of $\mathcal{M}^*$ is equal to

$$\begin{cases} \infty & \text{with probability } \delta, \\ \varepsilon & \text{with probability } (1-\delta) \cdot \frac{e^\varepsilon}{1+e^\varepsilon}, \\ -\varepsilon & \text{with probability } (1-\delta) \cdot \frac{1}{1+e^\varepsilon}. \end{cases}$$

4 Additive Noise Mechanisms with sub-sampling

For any mechanism $\mathcal{M}$, the Poisson sub-sampled version of the mechanism $\mathcal{M} \circ \mathcal{S}_q$ with sampling probability q operates by including each data point in a sub-sampled dataset independently with probability q and then returning the output of the mechanism on this sub-sampled dataset. This improves the privacy parameters of the mechanism; known as "amplification by sub-sampling" (see e.g. [BBG18]).

[2]See also [MV18] for an alternative proof.

4.1 Analysis under Add-Remove Neighboring Relation

In this case of add-remove neighboring relation, Theorem 11 in [ZDW22] shows that for any mechanism $\mathcal{M}$ with dominating PLD $\mathrm{PLD}_{\mu_{\mathrm{up}}/\mu_{\mathrm{lo}}}$, it holds that $\mathrm{PLD}_{(1-q)\mu_{\mathrm{up}}+q\mu_{\mathrm{lo}}/\mu_{\mathrm{lo}}}$ is a dominating PLD for $\mathcal{M} \circ \mathcal{S}_q$.[3]

Let μ be any symmetric noise distribution, namely $\mu(x) = \mu(-x)$ (or $f_\mu(x) = f_\mu(-x)$ for continuous distributions). Let P, Q, R denote the distributions μ, $\Delta + \mu$ and $-\Delta + \mu$. In other words, $P(x) = \mu(x)$, $Q(x) = \mu(x - \Delta)$ and $R(x) = \mu(x + \Delta)$.

The privacy loss distribution we consider for the Add and Remove adjacencies are: $\mathrm{PLD}_{\mathrm{add}} := \mathrm{PLD}_{P/Q}$ and $\mathrm{PLD}_{\mathrm{rem}} := \mathrm{PLD}_{R/P}$, with corresponding privacy loss functions:

$$\mathcal{L}_{\mathrm{add}}(x) \;=\; \log \frac{P(x)}{Q(x)} \;=\; \log \frac{\mu(x)}{\mu(x-\Delta)} \quad \text{and} \quad \mathcal{L}_{\mathrm{rem}}(x) \;=\; \log \frac{R(x)}{P(x)} \;=\; \log \frac{\mu(x+\Delta)}{\mu(x)}$$

While both lead to the same PLDs, the distinction is convenient when using subsampling as explained below. Note that, since μ is symmetric, we have $P(x) = P(-x)$ and $Q(x) = R(-x)$, and hence $\mathcal{L}_{\mathrm{add}}(x) = -\mathcal{L}_{\mathrm{rem}}(-x)$.

When Poisson subsampling probability q, the PLDs we consider are given as $\mathrm{PLD}^q_{\mathrm{add}} := \mathrm{PLD}_{P/(1-q)\cdot P+q\cdot Q}$ and $\mathrm{PLD}^q_{\mathrm{rem}} := \mathrm{PLD}_{(1-q)\cdot P+q\cdot R/P}$, and the corresponding privacy loss functions are:

$$\begin{aligned}
\mathcal{L}^q_{\mathrm{add}}(x) = \;& \log \frac{P(x)}{(1-q)\cdot P(x) + q \cdot Q(x)} \;&=\; -\log\left(1 - q + q \cdot e^{-\mathcal{L}_{\mathrm{add}}(x)}\right) \\
\mathcal{L}^q_{\mathrm{rem}}(x) = \;& \log \frac{(1-q)\cdot P(x) + q \cdot R(x)}{P(x)} \;&=\; \log\left(1 - q + q \cdot e^{\mathcal{L}_{\mathrm{rem}}(x)}\right)
\end{aligned}$$

where we drop the superscript of q when $q = 1$. Thus, we still have $\mathcal{L}^q_{\mathrm{add}}(x) = -\mathcal{L}^q_{\mathrm{rem}}(-x)$.

The accounting library supports Poisson sub-sampling for all the considered additive noise mechanisms (Laplace, Gaussian, Discrete Laplace and Discrete Gaussian mechanisms).

Also, note that in the special case of additive noise mechanisms, the dominating PLDs we consider in fact are also worst-case PLDs. Hence optimistic estimates of the privacy loss for these PLDs are also optimistic estimates of the privacy loss of the Poisson subsampled additive noise mechanism.

4.2 Analysis under Substitution Neighboring Relation

In case of substitution neighboring relation, we use Proposition 13 of [LRKS25] which shows that in the case of Poisson subsampled Gaussian mechanism, $\mathrm{PLD}_{(1-q)\cdot P+q\cdot R/(1-q)\cdot P+q\cdot Q}$ is a worst-case PLD, where $P = \mathcal{N}(0, \sigma^2)$, $Q = \mathcal{N}(\Delta, \sigma^2)$ and $R = \mathcal{N}(-\Delta, \sigma^2)$.[4]

The privacy loss function is given as:

$$\begin{aligned}
\mathcal{L}^q_{\mathrm{sub}}(x) \;&=\; \log\left(\frac{(1-q)\cdot P(x) + q \cdot R(x)}{(1-q)\cdot P(x) + q \cdot Q(x)}\right) \\
&=\; \log \frac{(1-q)\cdot e^{-\frac{x^2}{2\sigma^2}} + q \cdot e^{\frac{-(x+\Delta)^2}{2\sigma^2}}}{(1-q)\cdot e^{-\frac{x^2}{2\sigma^2}} + q \cdot e^{\frac{-(x-\Delta)^2}{2\sigma^2}}} \;=\; \log \frac{1 - q + q \cdot e^{\frac{-2x\Delta - \Delta^2}{2\sigma^2}}}{1 - q + q \cdot e^{\frac{2x\Delta - \Delta^2}{2\sigma^2}}}
\end{aligned}$$

[3] This is precisely where our asymmetric notation for dominating PLDs comes in handy.

[4] We believe their analysis in [LRKS25] should extend to other "reasonable" noise distributions, in particular, for Laplace, discrete Laplace and discrete Gaussian distributions considered in this library, but we do not support the analysis of such subsampled mechanisms under substitution at this time.

We compute $(\mathcal{L}^q_{\mathrm{sub}})^{-1}(\varepsilon)$ by solving the following equation:

$$e^{\varepsilon} \;=\; \frac{1-q+q\cdot e^{\frac{-2x\Delta-\Delta^2}{2\sigma^2}}}{1-q+q\cdot e^{\frac{2x\Delta-\Delta^2}{2\sigma^2}}} \;=\; \frac{\alpha+e^{-X}}{\alpha+e^{X}}$$

where we set $X := \frac{x\Delta}{\sigma^2}$, $\alpha := \frac{1-q}{q}\cdot e^{\frac{\Delta^2}{2\sigma^2}}$. This gives us:

$$\begin{aligned}
e^{\varepsilon+X} + (e^{\varepsilon}-1)\alpha - e^{-X} &= 0\\
e^{\varepsilon/2+X} - e^{-\varepsilon/2-X} &= (e^{-\varepsilon/2}-e^{\varepsilon/2})\cdot\alpha\\
\sinh(\varepsilon/2+X) &= -\alpha\cdot\sinh(\varepsilon/2)
\end{aligned}$$

Thus $X = \sinh^{-1}(-\alpha\cdot\sinh(\varepsilon/2)) - \varepsilon/2$, and hence

$$(\mathcal{L}^q_{\mathrm{sub}})^{-1}(\varepsilon) \;=\; \frac{\sigma^2}{\Delta}\left(\sinh^{-1}(-\alpha\cdot\sinh(\varepsilon/2)) - \varepsilon/2\right)$$

Naively computing sinh and $\sinh^{-1}$ in the above expression using, e.g. `math.sinh` and `math.asinh` is not numerically stable. Therefore, we instead use numerically stable primitives for $\log(\sinh(x))$ (defined only for $x > 0$) and $\sinh^{-1}(b\cdot e^a)$ as follows. For all $x > 0$, it holds that

$$\log(\sinh(x)) \;=\; \log\left(\frac{e^x-e^{-x}}{2}\right) \;=\; x + \log(1-e^{-2x}) - \log 2$$

For all $a, b \in \mathbb{R}$, $\sinh^{-1}(b\cdot e^a)$ can be obtained by solving for x in $b\cdot e^a = \frac{e^x-e^{-x}}{2}$ or equivalently $e^{2x} - 2be^a\cdot e^x - 1 = 0$, which we get as

$$\begin{aligned}
\sinh^{-1}(b\cdot e^a) &= \log\left(be^a + \sqrt{(be^a)^2+1}\right) && \ldots\text{useful when } b > 0\\
&= -\log\left(-be^a + \sqrt{(be^a)^2+1}\right) && \ldots\text{useful when } b < 0\\
&= \mathrm{sign}(b)\cdot\log\left(|b|e^a + \sqrt{(be^a)^2+1}\right)
\end{aligned}$$

Thus, $\sinh^{-1}(b\cdot e^a)$ can be computed using numerically stable implementation available in `np.logaddexp`.

We apply these identities to re-write $(\mathcal{L}^q_{\mathrm{sub}})^{-1}(\varepsilon)$ as:

$$\begin{aligned}
(\mathcal{L}^q_{\mathrm{sub}})^{-1}(\varepsilon) &= \frac{\sigma^2}{\Delta}\left(\sinh^{-1}(-\alpha\cdot\sinh(\varepsilon/2)) - \varepsilon/2\right)\\
&= \frac{\sigma^2}{\Delta}\left(\sinh^{-1}(-\mathrm{sign}(\varepsilon)\cdot\alpha\cdot\sinh(|\varepsilon/2|)) - \varepsilon/2\right)\\
&= \frac{\sigma^2}{\Delta}\left(\sinh^{-1}\left(-\mathrm{sign}(\varepsilon)\cdot e^{\log\alpha+\log\sinh(|\varepsilon/2|)}\right) - \varepsilon/2\right)
\end{aligned}$$

5 Other Implementation Details

In this section, we discuss other implementation details that are included in the library.

5.1 Fast Computation of Divergence of Composition of Two PLDs

Suppose we would like to find the ε-hockey stick divergence of the composition of two PLDs ω, ω'. This can, of course, be computed by first computing the convolution $\omega * \omega'$ and then compute the ε-hockey stick divergence using the formula in Observation 2.

Here we also implement a faster way to compute this: we may write the desired hockey stick divergence as

$$\begin{aligned}\mathfrak{D}_{e^\varepsilon}(\omega * \omega') &= \sum_{v \in \mathrm{supp}(\omega)} \sum_{v' \in \mathrm{supp}(\omega')} \omega(v) \cdot \omega'(v') \cdot \max\{0, 1 - e^{\varepsilon - v - v'}\} \\ &= \sum_{v \in \mathrm{supp}(\omega)} \sum_{\substack{v' \in \mathrm{supp}(\omega') \\ v+v' > \varepsilon}} \omega(v) \cdot \omega'(v') \cdot (1 - e^{\varepsilon - v - v'}) \\ &= \sum_{v \in \mathrm{supp}(\omega)} \omega(v) \cdot \left(\left(\sum_{\substack{v' \in \mathrm{supp}(\omega') \\ v+v' > \varepsilon}} \omega'(v') \right) - e^{\varepsilon - v} \left(\sum_{\substack{v' \in \mathrm{supp}(\omega') \\ v+v' > \varepsilon}} \omega'(v') \cdot e^{-v'} \right) \right)\end{aligned} \tag{2}$$

The above formula can be computed efficiently by first iterating over $v \in \mathrm{supp}(\omega)$ in increasing order, and then keeping a cumulative sum for

$$\sum_{\substack{v' \in \mathrm{supp}(\omega') \\ v+v' > \varepsilon}} \omega'(v') \quad \text{and} \quad \sum_{\substack{v' \in \mathrm{supp}(\omega') \\ v+v' > \varepsilon}} \omega'(v') \cdot e^{-v'}.$$

5.2 Truncation

Recording an entire PLD (even after discretization) is often costly and sometimes even impossible if the privacy loss values can be very large. As a result, our implementation truncates the tails of the distribution after compositions. For composition of two PLDs, we compute the composition using convolution and then truncate the two ends of the distribution so that the truncated mass is no more than a given value. For composing PLD with itself a number of times, truncation is slightly more complicated as we cannot see the entire output beforehand but needs to decide on truncation threshold right away for efficient convolution. Thus, we resort to using a Chernoff bound (similar to [KJPH20]). To state the bound, recall that the moment-generating function (MGF) of a distribution μ over real numbers is defined as $M_\mu(t) = \mathbb{E}_{o \sim \mu}[e^{to}]$. The Chernoff bound states that $\Pr_{o \sim \mu}[\mu \geq a] \leq M_\mu(t)/e^{ta}$ for any $t > 0$. Recall also that, if we let ω^{*n} denote $\omega * \cdots * \omega$ where the convolution is done $n - 1$ times, then we have the identity $M_{\omega^{*n}}(t) = M_\omega(t)^n$. Thus, we may compute a truncation point a_{upper} such that $\Pr_{o \sim \omega^{*n}}[\mu \geq a_{upper}] \leq \tau$ by

$$a_{upper} = \frac{n \cdot \log(M_\omega(t)) + \log(1/\tau)}{t}. \tag{3}$$

In our code, we compute the above bound for many orders $t > 0$ and take the best (i.e. smallest) among the derived bounds. A similar approach can be used for $t < 0$ to derive a truncation point a_{lower} such that $\Pr_{o \sim \omega^{*n}}[\mu \leq a_{lower}] \leq \tau$.

References

[BBG18] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *NeurIPS*, pages 6280–6290, 2018.

[BW18] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018.

[CGST24] Christopher A. Choquette-Choo, Arun Ganesh, Thomas Steinke, and Abhradeep Guha Thakurta. Privacy amplification for matrix mechanisms. In *International Conference on Learning Representations (ICLR)*, 2024.

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In *NeurIPS*, 2020.

[DGK+22] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions. *Proc. Priv. Enhancing Technol.*, 2022(4):552–570, 2022.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[DR16] Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016.

[GRS12] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

[KJH20] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *Artificial Intelligence and Statistics (AISTATS)*, 2020.

[KJPH20] Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight approximate differential privacy for discrete-valued mechanisms using FFT. *arXiv preprint arXiv:2006.07134*, 2020.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *ICML*, pages 1376–1385, 2015.

[LRKS25] Christian Janos Lebeda, Matthew Regehr, Gautam Kamath, and Thomas Steinke. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

[MM18] Sebastian Meiser and Esfandiar Mohammadi. Tight on budget?: Tight bounds for r-fold approximate differential privacy. In *CCS*, pages 247–264, 2018.

[MV18] Jack Murtagh and Salil P. Vadhan. The complexity of computing the optimal composition of differential privacy. *Theory Comput.*, 14(1):1–35, 2018.

[SMM19] David M. Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *PoPETs*, 2019(2):245–269, 2019.

[War65] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[ZDW22] Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022.