EFFICIENT COMPUTATION OF THE PRIVACY LOSS DISTRIBUTION FOR RANDOM ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the privacy amplification properties of a sampling scheme in which a user's data is used in k steps chosen randomly and uniformly from a sequence (or set) of t steps. This sampling scheme has been recently applied in the context of differentially private optimization (Chua et al., 2024a; Choquette-Choo et al.) and communication-efficient high-dimensional private aggregation (Asi et al., 2025) as well as studied theoretically in (Feldman & Shenfeld, 2025; Dong et al.). Existing analysis techniques lead to several ways to numerically approximate the privacy parameters of random allocation yet they all suffer from two drawbacks. First, the resulting privacy parameters are not tight due the approximation steps in the analysis. Second, the computed parameters are either the hockey stick divergence or Renyi DP both of which introduce overheads when additional composition and/or subsampling are needed (such as in multi-epoch optimization algorithms). In this work, we demonstrate that the privacy loss distribution (PLD) of random allocation applied to any differentially private algorithm can be computed efficiently. In particular, our PLD computation enables essentially lossless subsampling and composition. When applied to the Gaussian mechanism, our results demonstrate that random allocation can be used in place of Poisson subsampling with no degradation in resulting privacy guarantees.

1 INTRODUCTION

Privacy amplification by data sampling is one of the central techniques in the analysis of differentially private (DP) algorithms. In this technique a differentially private (DP) algorithm (or a sequence of DP algorithms) is executed on a randomly chosen set of data elements without revealing which of the elements were used. As first demonstrated Kasiviswanathan et al. (2011) this additional randomness can significantly improve the privacy guarantees of the resulting algorithm, that is, privacy amplification.

Privacy amplification by sampling has found numerous applications, most notably in the analysis of the differentially private stochastic gradient descent (DP-SGD) algorithm (Bassily et al., 2014) for training neural networks with differential privacy. In DP-SGD the gradients are computed on randomly chosen batches of data points and then privatized through Gaussian noise addition. Privacy analysis of this algorithm is based on the so-called Poisson sampling: elements in each batch and across batches are chosen randomly and independently of each other. The absence of dependence implies that the algorithm can be analyzed relatively easily as an independent composition of single step amplification results. This simplicity is also the key to accurate numerical analysis of the privacy parameters of DP-SGD that are necessary for the practical applications.

The downside of the simplicity of Poisson sampling is that independently resampling every batch is less efficient and harder to implement within the standard ML pipelines. As a result, in practice typically some form of data shuffling is used to define the batches in DP-SGD even though the privacy analysis relies on Poisson sampling (e.g. (McKenna et al., 2025)). Data shuffling in which the elements are randomly permuted before being assigned to steps of the algorithm is also known to lead to privacy amplification. However, the analysis of this sampling scheme is more involved and nearly tight numerical results are known only for relatively simple pure DP ($\delta = 0$) algorithms (Erlingsson et al., 2019; Feldman et al., 2021; 2023; Girgis et al., 2021a;b). In particular, for the case of Gaussian noise addition there is no practically useful method of computing the privacy parameters of DP-SGD with shuffling.

The discrepancy between the implementations of DP-SGD and their analysis has been explored in several recent works demonstrating that shuffling can be less private than Poisson subsampling (Chua et al., 2024b;c; Annamalai et al., 2024). Motivated by these findings, Chua et al. (2024a) study training of neural networks via DP-SGD with batches sampled via *balls-and-bins sampling*. In this sampling scheme, each data element is assigned randomly and independently (of other elements) to exactly one out of t possible batches. Their main results show that from the point of view of utility (namely, accuracy of the final model) such sampling is essentially identical to shuffling and is noticeably better than Poisson sampling. Concurrently, Choquette-Choo et al. considered the same sampling scheme for the matrix mechanism in the context of DP-FTRL. The privacy analysis in these two works reduces the problem to analyzing the divergence of a specific pair of distributions on $\mathbb{R}^t$. They then used Monte Carlo simulations to estimate the privacy parameters of this pair. These simulations suggest that privacy guarantees of balls-and-bins sampling for Gaussian noise are similar to those of the Poisson sampling with rate $1/t$. While very encouraging, such simulations do not establish formal guarantees. In addition, achieving high-confidence estimates for small δ and supporting composition appear to be computationally impractical.

Another important application of privacy amplification is for reducing communication in private federated learning (Chen et al., 2024; Asi et al., 2025). In this application, each user subsamples the coordinates of the vector it holds (typically representing a model update) and then communicates the selected coordinates. Secure aggregation protocols are used to ensure that the server does not learn which coordinates were sampled by which user, thereby achieving privacy amplification. In this setting, it is also typically necessary to limit the maximum number of coordinates a user sends due to computational or communication constraints on the protocol. Poisson subsampling results in a random (binomial) number of coordinates to communicate and thus does not allow to fully exploit the available limit. Thus in (Asi et al., 2025), a natural alternative is the sampling scheme in which each user contributes a random k out of the total t times (but with users still doing this independently). For $k = 1$ this sampling scheme is a special case of the balls-and-bins sampling (Chua et al., 2024a).

Motivated by the applications above, Feldman & Shenfeld (2025) propose and analyze a general sampling scheme where each element participates in exactly k randomly chosen steps out of the total t, independently of other elements, referred to as k-out-of-t *random allocation*. They show a reduction of the general k scheme to $k = 1$ and describe several ways to analyze the 1-out-of-t sampling scheme for general differentially private algorithms. Dong et al., independently derived an additional analysis of the privacy of k-out-of-t random allocation for the Gaussian noise addition.

The analyses in (Feldman & Shenfeld, 2025; Dong et al.) and the numerical methods they entail demonstrate that in most practical settings the privacy amplification achieved by random allocation is comparable to that of Poisson sampling with the best results being typically within 20% increase in ϵ. While reasonably close, these bounds are worse than the bounds estimated via Monte Carlo simulations (Chua et al., 2024a; Choquette-Choo et al.) and bounds that can be computed exactly in some special cases (Feldman & Shenfeld, 2025). Further, these analyses bound either the (ε, δ) parameters (Feldman & Shenfeld, 2025) or the Rényi DP parameters (Feldman & Shenfeld, 2025; Dong et al.) of the resulting algorithm. Both of these bounds have important limitations when used with additional processing steps. For example, the algorithm used in (Asi et al., 2025) relies on random allocation to reduce communication for each user but on top of it uses DP-SGD to sample batches of users using Poisson sampling and composition (for batches and epochs). In such an application, using an (ε, δ)-bound for random allocation would require performing composition for general (ε, δ) algorithms which is known to be suboptimal. On the other hand, the general subsampling bounds based on Rényi DP are typically loose. Further, conversion from Rényi DP to final (ε, δ) guarantees also typically introduces overheads.

1.1 Our contribution

We demonstrate how to overcome both shortcomings of the existing numerical methods for computing the privacy parameters of random allocation. Specifically, we show a method that, given a privacy loss distribution (PLD) of some t-step differentially private algorithm, computes an upper bound on the PLD of the 1-out-of-t random allocation applied to that algorithm. PLD is now the standard representation of privacy loss used in privacy accounting libraries (e.g (Google, 2022; Microsoft, 2021; Meta, 2021)) that can be losslessly composed and converted to other notions of DP such as (ε, δ)-DP and Rényi DP.

Our algorithm is efficient in that its running time is $O(\log^3(t) \cdot \log(t/\beta)/\alpha^2)$, where α is the approximation parameter of loss (roughly corresponding to the error in ε) and β an additional probability of unbounded loss (translating to an increase in δ). For comparison, the complexity of the standard algorithm for Poisson subsampling and t-wise composition via FFT is $O(t \cdot \sqrt{\log(t/\beta)} \cdot \log(t/\alpha)/\alpha)$ (Koskela et al., 2020; 2021; Gopi et al., 2021). Combining this with the reduction from the general k to $k = 1$ from (Feldman & Shenfeld, 2025) we also obtain an algorithm for computing the PLD of the k-out-of-t random allocation.

Technical overview: We now briefly outline our approach. As in the prior work, the starting point to our result is a relatively simple fact that a dominating pair of distributions[1] for a 1-out-of-t random allocation applied to a t-step algorithm M is the pair of distributions $\bar{Q}_t = Q^t$ and

$$\bar{P}_t = \frac{1}{t} \sum_{i \in [t]} Q^{i-1} \times P \times Q^{t-i},$$

where Q and P is a dominating pair of distributions for M. Equivalently, we can reduce the analysis of a potentially very complicated algorithm like DP-SGD where steps can depend on the outputs of previous steps to the analysis of random allocation applied to a fixed randomizer (specifically, one that samples from a distribution P when its input is the user's data and samples from distribution Q otherwise).

Now, our goal is to compute the PLD, or the distribution of $\ln(\bar{P}_t(x)/\bar{Q}_t(x))$ for $x \sim \bar{P}_t$. Somewhat more formally, we need to produce sufficiently accurate upper and lower bounds on this random variable to allow computation of the privacy parameters for both directions of the divergence. In general, computing a PLD of a mixture of high-dimensional distributions is unlikely to be computationally tractable. Our main observation is that for *parallel mixtures*, or mixtures in which each component of the mixture has its own output dimension, such a computation is feasible (see Thm. 3.2 for a formal statement). Specifically, for a pair of distributions P and Q we refer to the random variable $P(x)/Q(x)$ when $x \sim P$ as the privacy ratio distributions (PRD) of P, Q (or, $\exp$ of PLD). We observe that the PRD of the parallel mixture is just the weighted sum of the independent copies of the PRDs of the component distribution pairs. Thus the necessary computations can be performed using convolutions of PRDs.

We then describe how to appropriately discretize the PRDs and compute the t-wise convolution in time logarithmic in t and inverse quadratic in the desired accuracy. We note that upper and lower bound discretizations and convolution computation need to be handled differently for both directions to ensure correctness and avoid numerical stability issues. The dependence on accuracy is quadratic since these convolutions do not lend themselves to fast computations via a FFT. FFT relies on additive discretization whereas the privacy ratio has an extremely large dynamic range. Instead, we use a multiplicative discretization (which is equivalent to the standard additive approximation of the PLD). The logarithmic dependence on t is achieved by doubling the number of steps via a convolution of PRD with itself and using the binary representation of t.

To compute an upper bound on the PLD for general k-out-of-t, we use the reduction in (Feldman & Shenfeld, 2025), showing that k-out-of-t is at least as private as k-composition of 1-out-of-$\lfloor t/k \rfloor$ random allocation. While this reduction is lossy, in particular, when $\lfloor t/k \rfloor$ is relatively small we remark that the reduction is exact for Poisson sampling at the same rate. Namely, sampling independently at the rate of k/t for t steps is equivalent to sampling at the rate of k/t for t/k steps (which is the analog of 1-out-of-t/k random allocation) composed k times. Thus our empirical results showing that in most practical regimes 1-out-of-t random allocation is no less private than $1/t$-rate Poisson subsampling imply that k-out-of-t random allocation is no less private than k/t-rate Poisson subsampling.

Finally, to enable additional downstream applications of random allocation such as the PREAMBLE algorithm in (Asi et al., 2025), we derive and implement Poisson subsampling applied directly to a

[1]Informally, a pair of distributions is dominating for M if it realizes all the worst case privacy parameters of M (see Defn. 2.5).

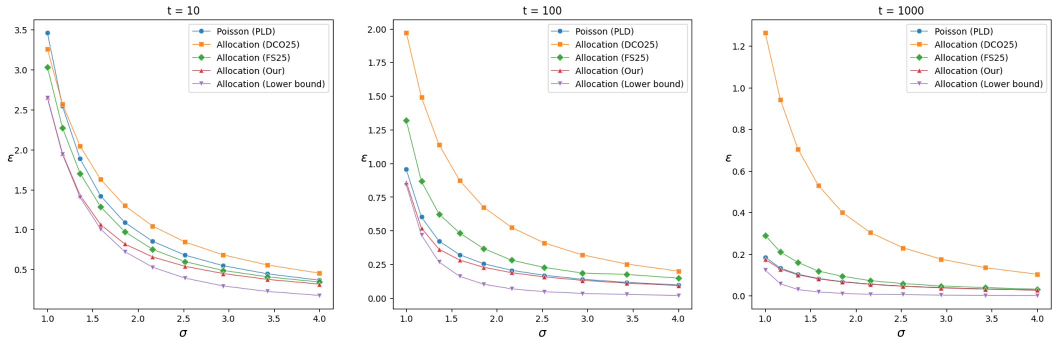

Figure 1: Upper and lower bounds on privacy parameter ε as a function of the noise parameter σ for various values of t, all using the Gaussian mechanism with fixed $\delta = 10^{-6}$. We compare our methods to upper bounds on Poisson and random allocation (Feldman & Shenfeld, 2025; Dong et al.), and lower bound on random allocation Chua et al. (2024a), and to the Poisson scheme with $\lambda = 1/t$.

(upper bound on a) PLD as we are not aware of any public description or implementation of this step. Instead, existing libraries rely directly on an analytic expression for the PLD of a subsampled Gaussian and Laplace noise addition (Google, 2022; Microsoft, 2021; Meta, 2021) See Appendix A.3 for details.

Numerical evaluation:

We compare our approach to existing techniques as well as Poisson subsampling in a variety of parameter settings. While our technique is general, we focus our evaluation on the Gaussian noise addition since that's the motivating application and the only case handled by most of the prior works. We note that we do not provide explicit results on the utility of random allocation, as such results can be found in prior work (Chua et al., 2024a; Choquette-Choo et al.; Feldman & Shenfeld, 2025; Dong et al.; Asi et al., 2025). Our privacy bounds only require knowing the noise and sampling parameters used there. Additional details on these numerical evaluations and additional evaluations can be found in Appendix C.

We start with a basic comparison with existing analysis methods for $k = 1$ and a range of t and σ (Figure 1). As can be seen from the plots, our results improve on all prior bounds and are never worse than the bounds for Poisson subsampling. We remark that the privacy bounds for these sampling techniques are incomparable in general (see Figure 6).

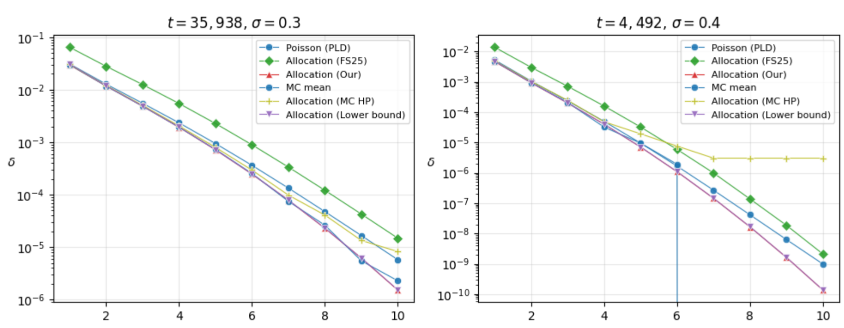

Figure 2: Comparison of the privacy profile of the Poisson scheme and various bounds for the random allocation scheme; the combined methods by Feldman & Shenfeld (2025), the high probability and the average estimations using Monte Carlo simulation and the lower bound by Chua et al. (2024a), and our numerical method, following the setting in Chua et al. (2024a) (detailed description can be found in Appendix C).

In Figure 2 we show that our results match those obtained via Monte Carlo simulations in the regimes where the latter produce reliable results. These experiments are in the regime of parameters studied (Chua et al., 2024a).

We additionally show the results for more general $k = 10$ (Appendix C). We note that for our method and results in (Feldman & Shenfeld, 2025) this setting is equivalent to testing k-wise composition for k, 1-out-t/k rounds or random allocation. The RDP-based bounds in (Dong et al.) handle general k directly.

Finally, we include a plot demonstrating the runtime efficiency of our algorithm in Fig. 3. It also demonstrates that the runtime scaling in t and α agrees with our theoretical claims.

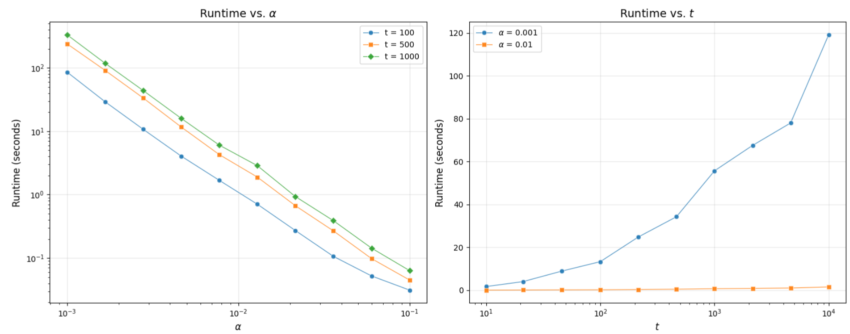

Figure 3: Runtime as a function of accuracy α and steps t on Apple MacBook Pro M1. The left panel was computed for Gaussian noise with $\sigma = 2.0$ and the right one for $\sigma = 5.0$.

1.2 Related work

Our work is most closely related to a long line of research on privacy amplification by subsampling and composition. This combination of tools was first defined and theoretically analyzed in the setting of convex optimization (Bassily et al., 2014). The resulting DP-SGD algorithm has found numerous applications in both theoretical and practical work and is currently the state-of-the-art method for training LLMs with provable privacy guarantees (VaultGemma Team, 2025). Applications of DP-SGD in machine learning were spearheaded by the landmark work of Abadi et al. (2016) who significantly improved the privacy analysis via the moments accounting technique formalized via Rényi DP (Mironov, 2017). This work has also motivated the development of more advanced techniques for analysis of sampling and composition. A more detailed technical and historical overview of subsampling and composition for DP can be found in the survey by Steinke (2022).

One of the important tools that emerged for the analysis of DP-SGD is privacy accounting via numerical tracking of the privacy loss random variable. This was first proposed by Koskela et al. (2020; 2021) who also demonstrated that privacy parameters of composition correspond to the convolution of PLDs and can be (approximately) computed via FFT applied to a discretization of the PLD. This approach to composition improved on the moments accountant technique since it avoids the somewhat lossy conversion from RDP parameters to (ε, δ) and is now the standard approach for the analysis of DP-SGD supported by several libraries (Google, 2022; Microsoft, 2021; Meta, 2021). We first note that while our computation also involves convolutions, we are adding privacy ratios and not their logarithms while at the same time ensuring the same kind of approximation guarantees. As a result, our algorithm is substantially different. At the same time, our algorithmic results, which we intend to publish as a Python library, fit naturally with the rest of the PLD toolkit and expand it to random allocation and general subsampling.

The shuffle model was first proposed by Bittau et al. (2017). The formal analysis of the privacy guarantees in this model was initiated in (Erlingsson et al., 2019; Cheu et al., 2019). Erlingsson et al. (2019) defined the sequential shuffling scheme that we discuss here and proved the first general privacy amplification results for this scheme, albeit only for pure DP algorithms. Improved analyses and extensions to approximate DP were given in (Balle et al., 2019; 2020; Feldman et al., 2021; 2023; Girgis et al., 2021a;b; Koskela et al., 2022). The privacy amplification guarantees of shuffling also apply to 1-out-of-t random allocation. Indeed, random 1-out-of-t allocation is a special case of the *random check-in* model of defining batches for DP-SGD in (Balle et al., 2020). Their analysis of this variant relies on the amplification properties of shuffling and thus does not lead to better privacy guarantees for random allocation than those that are known for shuffling.

Two recent works give formal analyses of k-out-of-t random allocation (Feldman & Shenfeld, 2025; Dong et al.). Feldman & Shenfeld (2025) describe three approximation approaches that are incomparable and also analyze the asymptotic behavior of random allocation. In the first analysis, they show that the approximate DP (ε, δ) privacy parameters of random allocation are upper bounded by those of the Poisson scheme with sampling probability $\approx k/t$ up to lower order terms which are asymptotically vanishing in t/k. This analysis does not lead to tight bounds when t/k is small and can at best match the bounds for the Poisson sampling. In the second analysis, Feldman & Shenfeld (2025) show that ε of random allocation with $k = 1$ is at most a constant (≈ 1.6) factor times larger than ε of the Poisson sampling with rate $1/t$ for the same δ. This analysis gives better bounds for small t, but is typically worse by the said factor than Poisson sampling.

Feldman & Shenfeld (2025) also describe a direct analysis of the divergence for the dominating pair of distributions. In the remove direction, they derive a closed form expression and relatively efficient algorithm for computing the integer $\alpha \geq 2$ order RDP parameters of random aloocation in terms of the RDP parameters of the original algorithm. For the add direction, they give an approximate upper bound directly on the (ε, δ) parameters. While this bound is approximate, the divergence for the add direction is typically significantly lower than the one for the remove direction and therefore even reasonably loose approximation of the add direction tends to not harm the overall bound. A similar approach to the analysis of random allocation was independently proposed in (Dong et al.). They provide upper bounds on the RDP parameters of the dominating pair of distributions in the Gaussian case for both add and remove directions. Their efficiently computable bound is exact for $\alpha = 2$ for the add direction and general k and is approximate otherwise.

Methods based on RDP parameters are particularly well-suited for subsequent composition (which simply adds up the RDP parameters) . The primary disadvantage of this technique is that the conversion from RDP bounds to the regular (ε, δ) bounds is known to be somewhat lossy (typically within 10-20% range in multi-epoch settings). The bounds in (Feldman & Shenfeld, 2025; Dong et al.) are also harmed by the restriction $\alpha \geq 2$ since lower order α lead to the best (ε, δ) parameters in some cases. As mentioned above, subsampling of the RDP bounds typically incurs overheads making this approach less viable in the complex settings such as (Asi et al., 2025).

2 PRELIMINARIES

In this work we consider *t-step algorithms* defined using a randomized algorithm $M : \mathcal{X}^* \times \mathcal{Y}^* \to \mathcal{Y}$, which given a *dataset* $\boldsymbol{s} \in \mathcal{X}^*$ of *elements* in the input space and a *view* $\boldsymbol{v} \in \mathcal{Y}^*$ consisting of output values, produces a new output. It first uses some scheme to define t subsets $\boldsymbol{s}^1, \ldots, \boldsymbol{s}^t \subseteq \boldsymbol{s}$, then sequentially computes $y_i = M\left(\boldsymbol{s}^i, \boldsymbol{v}^{i-1}\right)$, where $\boldsymbol{v}^i := (y_1, \ldots, y_i)$ are the intermediate views consisting of the outputs produced so far, and $\boldsymbol{v}^0 = \emptyset$. Such algorithms include DP-SGD, where each step consists of a call to the Gaussian mechanism (A.2), with gradient vectors adaptively defined as a function of previous outputs.

The assignment of the elements in $\boldsymbol{s}$ to the various subsets can be done in a deterministic manner (e.g., $\boldsymbol{s}^1 = \ldots = \boldsymbol{s}^t = \boldsymbol{s}$), or randomly using a *sampling scheme*. We consider two sampling schemes.

1. *Poisson scheme* parametrized by sampling probability $\lambda \in [0, 1]$, where each element is added to each subset $\boldsymbol{s}^i$ with probability λ independent of the other elements and other subsets,
2. *Random allocation scheme* parametrized by a number of selected steps $k \in [t]$, which uniformly samples k indices $\boldsymbol{i} = (i_1, \ldots, i_k) \subseteq [t]$ for each element and adds it to the corresponding subsets $\boldsymbol{s}^{i_1}, \ldots, \boldsymbol{s}^{i_k}$.

For a t-step algorithm defined by M, we denote by $\mathcal{P}_{t,\lambda}(M) : \mathcal{X}^* \to \mathcal{Y}^t$ an algorithm using M with the Poisson sampling scheme and $\mathcal{A}_{t,k}(M) : \mathcal{X}^* \to \mathcal{Y}^t$ when M is used with the random allocation scheme. When $k = 1$ we omit it from the notation for clarity.

Differential privacy and Privacy loss distribution: We start by defining the abstract notion of privacy loss distribution (PLD).

Definition 2.1 (PLD (Dwork & Rothblum, 2016))**.** Given two distributions P, Q over some domain Ω, the *privacy loss random variable* $L_{P,Q}$ is defined by $\ell\left(\omega; P, Q\right) \coloneqq \ln\left(\frac{P(\omega)}{Q(\omega)}\right)$ where $\omega \sim P$. We refer to its distribution as the *privacy loss distribution (PLD)* and denote its CDF by $F^{\ell}_{P,Q}$.

We use the PLD to define a the standard hockey stick divergence between distributions.

Definition 2.2 (Hockey-stick divergence Barthe et al. (2012))**.** Given $\kappa \in [0, \infty]$, the *κ-hockey-stick divergence* between two distributions P, Q is defined as $\boldsymbol{H}_{\kappa}\left(P \parallel Q\right) \coloneqq \mathbb{E}\left[\left[1 - \kappa \cdot e^{-L_{P,Q}}\right]_+\right]$, where $[x]_+ \coloneqq \max\{0, x\}$.

We note that this definition extends to any random variable L defining its *κ-hockey-stick functional* as $\boldsymbol{H}_{\kappa}\left(L\right) \coloneqq \mathbb{E}\left[\left[1 - \kappa \cdot e^{-L}\right]_+\right]$.

For adjacency we consider the standard add/remove notion in which datasets $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{X}^*$ are adjacent if $\boldsymbol{s}$ can be obtained from $\boldsymbol{s}'$ via adding or removing a single element. To appropriately define sampling schemes that operate over a fixed number of elements we augment the domain with a "null" element $\perp$, that is, we define $\mathcal{X}' \coloneqq \mathcal{X} \cup \{\perp\}$. When a t-step algorithm assigns $\perp$ to M we treat it as an empty set, that is, for any $\boldsymbol{s} \in \mathcal{X}^*$, $\boldsymbol{v} \in \mathcal{Y}^*$ we have $M(\boldsymbol{s}, \boldsymbol{v}) = M((\boldsymbol{s}, \perp), \boldsymbol{v})$. We say that two datasets $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{X}^n$ are *adjacent* and denote it by $\boldsymbol{s} \simeq \boldsymbol{s}'$, if one of the two can be created by replacing a single element in the other dataset by $\perp$.

Using this notion we define the privacy profile of a mechanism, and use it to define differential privacy.

Definition 2.3 (Privacy profile (Balle et al., 2018))**.** Given an algorithm $M : \mathcal{X}^* \times \mathcal{Y}^* \to \mathcal{Y}$, the privacy profile $\delta_M : \mathbb{R} \to [0, 1]$ is defined to be the maximal hockey-stick divergence between the distributions induced by any adjacent datasets and past view. Formally,

$$\delta_M(\varepsilon) \coloneqq \sup_{\boldsymbol{s} \simeq \boldsymbol{s}' \in \mathcal{X}^*, \boldsymbol{v} \in \mathcal{Y}^*} \left(\boldsymbol{H}_{e^{\varepsilon}}\left(M(\boldsymbol{s}, \boldsymbol{v}) \parallel M(\boldsymbol{s}', \boldsymbol{v})\right)\right).$$

Since the hockey-stick divergence is asymmetric in the general case, we use $\vec{\delta}_M$ to denote the *remove* direction where $\perp \in \boldsymbol{s}'$ and $\overleftarrow{\delta}_M$ to denote the *add* direction when $\perp \in \boldsymbol{s}$. Consequently, $\delta_M(\varepsilon) = \max\{\vec{\delta}_M(\varepsilon), \overleftarrow{\delta}_M(\varepsilon)\}$.

We can now formally define the standard notion of DP.

Definition 2.4 (Differential privacy (Dwork et al., 2006))**.** Given $\varepsilon > 0$; $\delta \in [0, 1]$, an algorithm M will be called *(ε, δ)-differentially private (DP)*, if $\delta_M(\varepsilon) \leq \delta$.

Dominating pairs: A key concept for characterizing the privacy guarantees of an algorithm is that of a *dominating pair* of distributions (Zhu et al., 2022).

Definition 2.5 (Dominating pair (Zhu et al., 2022))**.** Given distributions P, Q over some domain Ω, and P', Q' over Ω', we say (P', Q') *dominate* (P, Q) if for all $\kappa \geq 0$ we have $\boldsymbol{H}_{\kappa}\left(P \parallel Q\right) \leq \boldsymbol{H}_{\kappa}\left(P' \parallel Q'\right)$. If $\vec{\delta}_M(\varepsilon) \leq \boldsymbol{H}_{e^{\varepsilon}}\left(P \parallel Q\right)$ for all $\varepsilon \in \mathbb{R}$, we say (P, Q) is a *dominating pair* of distributions for M in the remove direction, and replacing $\vec{\delta}_M$ by $\overleftarrow{\delta}_M$ this hold for the add direction.

If the inequality can be replaced by an equality for all ε, we say it is a *tightly dominating pair*. If there exist some $\boldsymbol{s} \simeq \boldsymbol{s}' \in \mathcal{X}^*$ such that $P = M(\boldsymbol{s})$, $Q = M(\boldsymbol{s}')$ we say $(\boldsymbol{s}, \boldsymbol{s}')$ are the dominating pair of datasets for M. By definition, a dominating pair of input datasets is tightly dominating.

Zhu et al. (2022) provide several useful properties of dominating pairs; A tightly dominating pair (P, Q) always exists (Proposition 8), if (P, Q) dominate $\vec{\delta}_M$, then (Q, P) dominate $\overleftarrow{\delta}_M$ (Lemma 28), and domination is preserved under composition (Theorem 10) and sampling (Theorem 11).

Using the PLD definition introduces another natural domination notion.

Definition 2.6 (Approximate Stochastic Domination)**.** A random variable X (first order) *stochastically dominates* another random variable X' if the complementary cumulative distribution function

(CCDF) of X upper bounds the CCDF of X', that is, for any value $x \in \mathbb{R}$ we have $\bar{F}_{X'}(x) \leq \bar{F}_X(x)$, where $\bar{F}_X = 1 - F_X$. Further, given $\alpha \geq 0$; $\beta \in [0, 1]$, we say this domination is (α, β)-approximate if $X' + \alpha$ stochastically dominates X up to a gap of β in probability. Formally, $\forall x \in \mathbb{R}: \ \bar{F}_X(x) \leq \bar{F}_{X'}(x - \alpha) + \beta$.

Like hockey-stick domination, stochastic domination is preserved under composition (Claim A.1) and subsampling (Appendix A.3) as well . The next claim shows how these two domination notions are related to each other. A proof can be found in Appendix A.

Claim 2.7. *Stochastic domination implies domination in the hockey-stick sense. Formally, given* $\alpha \geq 0$*;* $\beta \in [0, 1]$*, if a random variable* X *stochastically dominates* X' *and this domination is* (α, β)*-approximate, then* $\boldsymbol{H}_{e^\varepsilon}(X') \leq \boldsymbol{H}_{e^\varepsilon}(X) \leq \boldsymbol{H}_{e^{\varepsilon-\alpha}}(X') + \delta$.

We use the notion of dominating pair to define a dominating randomizer, which captures the privacy guarantees of the algorithm independently of its algorithmic adaptive properties.

Definition 2.8 (Dominating randomizer)**.** Given an algorithm $M \colon \mathcal{X}^* \times \mathcal{Y}^* \to \mathcal{Y}$, we say that $R : \{*, \perp\} \to \mathcal{Y}$ is a *dominating randomizer* for M and set $R(*) = P$ and $R(\perp) = Q$, where (P, Q) is the dominating pair of M in the remove direction.

Lemma 2.9 (Allocation reduction to randomizer (Feldman & Shenfeld, 2025))**.** *Given* $t \in \mathbb{N}$*;* $k \in [t]$ *and an algorithm* M *dominated by a randomizer* R*, we have* $\delta_{\mathcal{A}_{t,k}(M)}(\varepsilon) \leq \delta_{\mathcal{A}_{t,k}(R)}(\varepsilon)$

For the general case of multiple allocations we rely on the following reduction.

Lemma 2.10 (Reduction to a single allocation (Feldman & Shenfeld, 2025))**.** *For any* $k \in \mathbb{N}$*,* $\varepsilon > 0$ *we have* $\delta_{\mathcal{A}_{t,k}(R)}(\varepsilon) \leq \delta^{\otimes k}_{\mathcal{A}_{\lfloor t/k \rfloor}(R)}(\varepsilon)$*, where* $\otimes k$ *denotes the composition of* k *runs of the algorithm or scheme which in our case is* $\mathcal{A}_{\lfloor t/k \rfloor}(R)$*.*

3 PRIVACY OF RANDOM ALLOCATION VIA PRD CONVOLUTION

We start by introducing a new privacy random variable complementary to the PLD, which proves more useful for our next claims.

Definition 3.1 (Privacy ratio distribution)**.** Given two distributions P, Q over some domain Ω, the *privacy ratio random variable* $R_{P,Q}$ is defined by $\mathcal{R}(\omega; P, Q) := e^{\ell(\omega; P, Q)}$ where $\omega \sim P$. We refer to its distribution as the *privacy ratio distribution (PRD)* and denote its CDF by $F^{\mathcal{R}}_{P,Q}$.

Since $L_{P,Q} = \ln(R_{P,Q})$, stochastic domination of PLDs and PRDs are equivalent.

We can now state our first result.

Theorem 3.2 (Parallel mixing)**.** *Given* $\lambda \in [0, 1]$*, two distributions* P, Q *over some domain* Ω*, and* P', Q' *over* Ω'*, denote by* $\bar{Q} := Q \times Q'$ *the base product distribution, and by* $\bar{P}_\lambda := \lambda P \times Q' + (1 - \lambda) Q \times P'$ *the mixture distribution which either replaces* Q *with* P *or* Q' *by* P' *w.p.* λ *and* $1 - \lambda$ *respectively.*

For any $\omega \in \Omega$*,* $\omega' \in \Omega'$ *we have* $\mathcal{R}\left((\omega,\omega'); \bar{P}_\lambda, \bar{Q}\right) = \lambda \cdot \mathcal{R}(\omega; P, Q) + (1 - \lambda) \cdot \mathcal{R}(\omega; P', Q')$ *which implies*

$$R_{\bar{P}_\lambda, \bar{Q}} := \lambda^2 R_{P,Q} + \frac{\lambda(1-\lambda)}{R_{Q',P'}} + \frac{\lambda(1-\lambda)}{R_{Q,P}} + (1-\lambda)^2 R_{P',Q'} \quad \textit{and} \quad R_{\bar{Q}, \bar{P}_\lambda} := \frac{1}{\frac{\lambda}{R_{Q,P}} + \frac{1-\lambda}{R_{Q',P'}}}.$$

The advantage of this representation is that it decomposes the PRD of the mixture into (the inverse of) a sum of independent random variables corresponding to the PRDs of the components (or their inverses). Notice that the mixture $\bar{P}_\lambda$ affects both $\mathcal{R}\left((\omega, \omega'); \bar{P}_\lambda, \bar{Q}\right)$ and the sampling of (ω, ω') as well in the case of $R_{\bar{P}_\lambda, \bar{Q}}$. This lemma can be generalized to an arbitrary number of distribution pairs recursively.

A direct application of this lemma is the PRD of random allocation.

Corollary 3.3. *Given two distributions* P, Q *and an integer* t *denote the uniform distribution over* t *steps,* $\bar{P}_t := \frac{1}{t} \sum_{i \in [t]} Q^{i-1} \times P \times Q^{t-i}$*.*

For any $v = (\omega_1, \ldots, \omega_t) \in \Omega^t$ we have $\mathcal{R}\left(v; \bar{P}_t, Q^t\right) = \frac{1}{t}\sum_{i\in[t]} \mathcal{R}\left(\omega_i; P, Q\right)$, which implies

$$R_{\bar{P}_t,Q^t} = \frac{1}{t}\left(R_{P,Q} + \sum_{i\in[t-1]} \frac{1}{R_{Q,P}}\right) \quad \textit{and} \quad R_{Q^t,\bar{P}_t} = \frac{t}{\sum_{i\in[t]} \frac{1}{R_{Q,P}}}.$$

We can now state our main result that relies on Cor. 3.3.

Theorem 3.4. *Given $\alpha \geq 0$; $\beta \in [0, 1]$; $t \in \mathbb{N}$, and a t-step algorithm M tightly dominated by a pair of distributions P, Q, there exists an algorithm that given α, β, t and access to $F^{\ell}_{P,Q}$ returns two discrete random variables $\vec{L}, \overleftarrow{L}$, such that:* ***(1) Validity:*** *$\vec{L}$ ($\overleftarrow{L}$) dominates M in the remove (add) direction,* ***(2) Tightness:*** *this domination is (α, β)-approximate, and* ***(3) Computation complexity:*** *The runtime of the algorithm is $O\left(\Delta^2 \cdot \ln^3(t)/\alpha^2\right)$, where Δ is the width of interval between the $\beta/(2t)$ and $1 - \beta/(2t)$ quantiles of $L_{Q,P}$.*

In the case of the Gaussian mechanism with sensitivity 1, $\Delta = O(\sqrt{\ln(t/\beta)}/\sigma)$ so the runtime of the algorithm is $O\left(\ln^3(t)\ln(t/\beta)/(\sigma^2\alpha^2)\right)$. A detailed version of the algorithm can be found in Appendix B. We provide here its outline.

We start by creating discrete random variables that stochastically dominate $R_{P,Q}$ and $1/R_{Q,P}$ for the remove direction (dominated by $R_{Q,P}$ for the add direction). In the case of the Gaussian mechanism, these are simply lognormal random variables. The range is chosen such that $O(\beta)$ probability mass is discarded at each side, and the grid points are geometrically spaced so that conversion of PRD to PLD by taking the log will result in an additive error of $O(\alpha)$.

To avoid exponential growth of the range, a new grid is computed at each convolution step by truncating $O(\beta)$ probability from each end and selecting new geometrically spaced points (Alg. 1). The convolution is computed directly and the probabilities are allocated to bins by upper / lower bounding the random variable, according to the required domination direction (Alg 2).

The computation is carried out only $O(\log(t))$ times, leveraging the fact that the t-step convolution of a random variable with itself can be computed recursively by representing t as a sum of powers of 2 (e.g., if $t = 10$, we can compute the 2, 4, and 8-fold convolutions, then convolve the 8th and the 2nd, Alg 3).

In practice, numerical stability affects probabilities close to machine accuracy (10^{-15} for float64), which can be mitigated by using float128, or double-double arithmetic, both at the cost of additional computation. Since these inaccuracies grow with the number of compositions, it requires careful care whenever $\delta \leq 10^{-15+\log(t)}$.

Combining this theorem with Claim 2.7 implies the privacy profile computed using this algorithm is valid and tight as well.

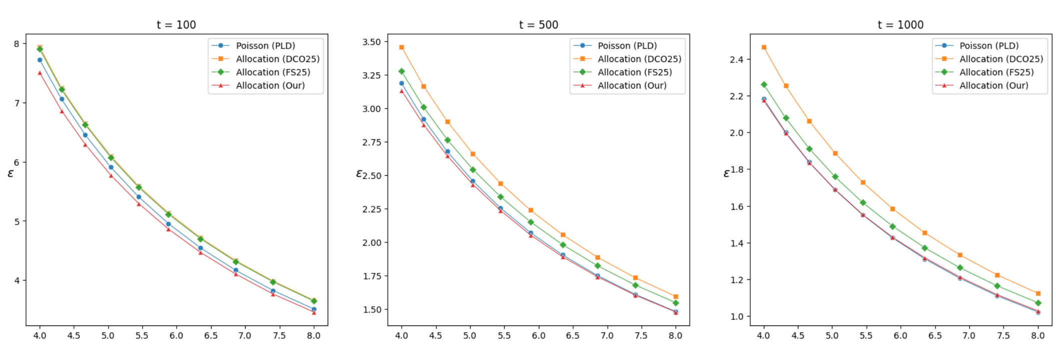

Figure 4: Privacy bounds using the same setting as in Figure 1with $\delta = 10^{-8}$ and $k = 10$ allocations.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Meenatchi Sundaram Muthu Selva Annamalai, Borja Balle, Emiliano De Cristofaro, and Jamie Hayes. To shuffle or not to shuffle: Auditing dp-sgd with shuffling. *arXiv preprint arXiv:2411.10614*, 2024.

Hilal Asi, Vitaly Feldman, Hannah Keller, Guy N. Rothblum, and Kunal Talwar. PREAMBLE: Private and efficient aggregation of block sparse vectors and applications. Cryptology ePrint Archive, Paper 2025/490, 2025. URL `https://eprint.iacr.org/2025/490`.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pp. 394–403. PMLR, 2018.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31, 2018.

Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pp. 638–667. Springer, 2019.

Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33:4623–4634, 2020.

Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Beguelin. Probabilistic relational reasoning for differential privacy. In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pp. 97–110, 2012.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pp. 464–473. IEEE, 2014.

Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*, pp. 441–459, 2017.

Wei-Ning Chen, Dan Song, Ayfer Ozgur, and Peter Kairouz. Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part I 38*, pp. 375–403. Springer, 2019.

Christopher A Choquette-Choo, Arun Ganesh, Saminul Haque, Thomas Steinke, and Abhradeep Guha Thakurta. Near-exact privacy amplification for matrix mechanisms. In *The Thirteenth International Conference on Learning Representations*.

Lynn Chua, Badih Ghazi, Charlie Harrison, Pritish Kamath, Ravi Kumar, Ethan Jacob Leeman, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Balls-and-bins sampling for dp-sgd. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2024a.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How private are dp-sgd implementations? In *Forty-first International Conference on Machine Learning*, 2024b.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Scalable dp-sgd: Shuffling vs. poisson subsampling. *Advances in Neural Information Processing Systems*, 37:70026–70047, 2024c.

Andy Dong, Wei-Ning Chen, and Ayfer Ozgur. Leveraging randomness in model and data partitioning for privacy amplification. In *Forty-second International Conference on Machine Learning*.

Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pp. 486–503. Springer, 2006.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

Vitaly Feldman and Moshe Shenfeld. Privacy amplification by random allocation. *arXiv preprint arXiv:2502.08202*, 2025.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2021.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Stronger privacy amplification by shuffling for rényi and approximate differential privacy. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 4966–4981. SIAM, 2023.

Antonious M. Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE Journal on Selected Areas in Information Theory*, 2(1):464–478, 2021a.

Antonious M Girgis, Deepesh Data, Suhas Diggavi, Ananda Theertha Suresh, and Peter Kairouz. On the renyi differential privacy of the shuffle model. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2321–2341, 2021b.

Google. dp-accounting. `https://github.com/google/differential-privacy/tree/main/python/dp_accounting`, 2022.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pp. 2560–2569. PMLR, 2020.

Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. In *International Conference on Artificial Intelligence and Statistics*, pp. 3358–3366. PMLR, 2021.

Antti Koskela, Mikko A Heikkilä, and Antti Honkela. Numerical accounting in the shuffle model of differential privacy. *Transactions on Machine Learning Research*, 2022.

Ryan McKenna, Yangsibo Huang, Amer Sinha, Borja Balle, Zachary Charles, Christopher A Choquette-Choo, Badih Ghazi, George Kaissis, Ravi Kumar, Ruibo Liu, et al. Scaling laws for differentially private language models. *arXiv preprint arXiv:2501.18914*, 2025.

Meta. Opacus: Differential privacy library for pytorch. https://github.com/pytorch/opacus, 2021.

Microsoft. PRV accountant. https://github.com/microsoft/prv_accountant, 2021.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Thomas Steinke. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

VaultGemma Team. VaultGemma: A differentially private Gemma model. Technical report, Google, 2025. URL https://services.google.com/fh/files/blogs/vaultgemma_tech_report.pdf.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*, pp. 4782–4817. PMLR, 2022.

A MISSING DETAILS

A.1 MISSING PROOFS

Claim A.1. *Given random variables X, X', Y, Y' denote $Z = X + Y$, $Z' = X' + Y'$. If X stochastically dominates X' and Y stochastically dominates Y', then Z stochastically dominates Z'.*

Proof. For any $z \in \mathbb{R}$ we have

$$\bar{F}_{Z'}(z) = \bar{F}_{X'+Y'}(z) = \int_{-\infty}^{\infty} \bar{F}_{X'}(x)\bar{F}_{Y'}(z-x)dx \leq= \int_{-\infty}^{\infty} \bar{F}_{X}(x)\bar{F}_{Y}(z-x)dx = \bar{F}_{X+Y}(z) = \bar{F}_{Z}(z).$$

□

Proof of Claim 2.7. We prove that stochastic domination implies domination in the hockey-stick sense, which implies both inequalities.

$$\begin{aligned}
\boldsymbol{H}_{\alpha}(X') &= \mathbb{E}\left[\left[1 - \alpha e^{-X'}\right]_+\right] \\
&= \int_0^1 \mathbb{P}\left(\left[1 - \alpha e^{-X}\right]_+ > t\right) dt \\
&= \int_0^1 \bar{F}_{X'}\left(\ln\left(\frac{\alpha}{1-t}\right)\right) dt \\
&\leq \int_0^1 \bar{F}_{X}\left(\ln\left(\frac{\alpha}{1-t}\right)\right) dt \\
&= \int_0^1 \mathbb{P}\left(\left[1 - \alpha e^{-X}\right]_+ > t\right) dt \\
&= \mathbb{E}\left[\left[1 - \alpha e^{-X}\right]_+\right] \\
&= \boldsymbol{H}_{\alpha}(X)
\end{aligned}$$

□

Proof of Theorem 3.2. From the definition,

$$\begin{aligned}
\mathcal{R}\left((\omega, \omega'); \bar{P}_\lambda, \bar{Q}\right) &= \frac{\bar{P}_\lambda(\omega, \omega')}{\bar{Q}(\omega, \omega')} \\
&= \lambda \frac{P(\omega)}{Q(\omega)} + (1-\lambda)\frac{P'(\omega')}{Q'(\omega')} \\
&= \lambda \cdot \mathcal{R}(\omega; P, Q) + (1-\lambda) \cdot \mathcal{R}(\omega'; P', Q')
\end{aligned}$$

Since $\mathcal{R}(\omega; P, Q) = \frac{1}{\mathcal{R}(\omega; Q, P)}$ for any ω, P, Q, we have

$$\mathcal{R}\left((\omega, \omega'); \bar{Q}, \bar{P}_\lambda\right) = \frac{1}{\lambda \cdot \mathcal{R}(\omega_1; P, Q) + (1-\lambda) \cdot \mathcal{R}(\omega_2; P', Q')} = \frac{1}{\frac{\lambda}{\mathcal{R}(\omega_1; Q, P)} + \frac{1-\lambda}{\mathcal{R}(\omega_2; Q', P')}},$$

which implies, $R_{\bar{Q}, \bar{P}_\lambda} := \frac{1}{\frac{\lambda}{R_{Q,P}} + \frac{1-\lambda}{R_{Q',P'}}}$.

Changing the distribution affects the sampling of (ω, ω') as well so $R_{\bar{P}_\lambda, \bar{Q}}$ becomes a mixture of two distributions. Using the fact that,

$$\mathcal{R}\left((\omega, \omega'); \bar{P}_\lambda, \bar{Q}\right) = \lambda \frac{1}{\mathcal{R}(\omega_1; Q, P)} + (1-\lambda) \cdot \mathcal{R}(\omega_2; P', Q') = \lambda \cdot \mathcal{R}(\omega_1; P, Q) + (1-\lambda)\frac{1}{\mathcal{R}(\omega_2; Q', P')},$$

and combining the two,

$$\begin{aligned}\mathcal{R}\left((\omega,\omega');\bar{P}_\lambda,\bar{Q}\right) &= \lambda\left(\lambda\cdot\mathcal{R}\left(\omega_1;P,Q\right)+(1-\lambda)\frac{1}{\mathcal{R}\left(\omega_2;Q',P'\right)}\right)\\ &\quad+(1-\lambda)\left(\lambda\frac{1}{\mathcal{R}\left(\omega_1;Q,P\right)}+(1-\lambda)\cdot\mathcal{R}\left(\omega_2;P',Q'\right)\right)\\ &= \lambda^2\mathcal{R}\left(\omega_1;P,Q\right)+\frac{\lambda(1-\lambda)}{\mathcal{R}\left(\omega_2;Q',P'\right)}+\frac{\lambda(1-\lambda)}{\mathcal{R}\left(\omega_1;Q,P\right)}+(1-\lambda)^2\mathcal{R}\left(\omega_2;P',Q'\right)\end{aligned}$$

we get

$$R_{\bar{P}_\lambda,\bar{Q}} \coloneqq \lambda^2 R_{P,Q}+\frac{\lambda(1-\lambda)}{R_{Q',P'}}+\frac{\lambda(1-\lambda)}{R_{Q,P}}+(1-\lambda)^2 R_{P',Q'}.$$

□

Proof of Corollary 3.3. This is a direct result of Theorem 3.2 using the recursive relation $\bar{P}_t = \frac{1}{t}\cdot P\times Q^{t-1}+\left(1-\frac{1}{t}\right)\cdot Q\times\bar{P}_{t-1}$.

In the base case $t=2$ we have,

$$R_{\bar{P}_2,Q^2}=\frac{1}{2}\left(R_{P,Q}+\frac{1}{R_{Q,P}}\right)\quad\text{and}\quad R_{Q^2,\bar{P}_2}=\frac{t}{\frac{1}{R_{Q,P}}+\frac{1}{R_{Q,P}}},$$

and for any $t>2$ we have

$$\mathcal{R}\left(v;\bar{P}_t,Q^t\right)=\frac{1}{t}\cdot\mathcal{R}\left(\omega_1;P,Q\right)+\left(1-\frac{1}{t}\right)\cdot\mathcal{R}\left(v_{2:t};\bar{P}_{t-1},Q^{t-1}\right).$$

□

Proof of Theorem 3.4. We state the analysis in terms of the remove direction. The analysis for the add directions is identical, except for the direction of the domination, since the last step consists of the monotonically decreasing transformation $-\ln$.

Validity: From Claim A.1, it suffices to show that every step of the algorithm maintains stochastic domination, to prove its output stochastically dominates the true PLD as well. The first step consists of lower bounding the underlying PLD's CDF which results in a stochastically dominating random variable. The left tail is treated as 0 and the right tail is treated as some probability mass at infinity. At each step of the convolution, the output random variable is defined by lower bounding the product random variable's CDF (moving some additional probability mass to infinity as needed), which results in a stochastically dominating random variable. Since ln is a monotonically increasing function, domination is maintained for the PLD.

Tightness: From Claim A.1, if X (Y) stochastically dominates X' (Y'), and these dominations are (α,β)-approximate, then $X+Y$ dominates $X'+Y'$ and this domination is $(2\alpha,2\beta)$-approximate. We analyze the two slackness components separately. The β terms are accumulated additively. Since there are t convolution steps, each one contributing at most $\beta'=\beta/(2t)$ to the probability loss from truncation, the overall loss from this part is $\beta/2$. Additionally, the initial PLD discards $\beta/2$ probability mass, leading to a combined loss of β. The α part results from the choice of the bins. Since they are geometrically spaced, the resulting error from rounding into bins is *relative* rather than additive, so the convolution over t steps results only in $\log(t)$ blowup in error.

Computational complexity: Since we do not use evenly spaced bins, we cannot rely on FFT, so the convolution must be carried out in $O(n^2)$ time, where n is the number of bins. This number is the ratio of the range Δ to the desired resolution α'. Δ is determined by the dropped probability mass β, and since $\alpha'=\alpha/\log(t)$, the first convolution requires $O(\Delta^2\log^2(t)/\alpha^2))$ steps. Using Algorithm 1 we maintain the same number of bins, and using Algorithm 3 requires only $O(\log(t))$ convolution steps, which completes the proof. □

A.2 GAUSSIAN MECHANISM

One of the most common algorithms is the Gaussian mechanism N_σ, which simply reports the sum of (some function of) the elements in the dataset with the addition of Gaussian noise of scale σ. One of its main advantages is that we have closed form expressions of its privacy

Lemma A.2 (Gaussian mechanism DP guarantees, (Balle & Wang, 2018))**.** *Given $d \in \mathbb{N}$; $\sigma > 0$, let $\mathcal{X} = \mathcal{Y} := \mathbb{R}^d$. The* Gaussian mechanism *N_σ is defined as $N_\sigma(\boldsymbol{s}) := \mathcal{N}(\sum_{x \in \boldsymbol{s}} x, \sigma^2 I_d)$.*

If the domain of N_σ is the unit ball in $\mathbb{R}^d$, we have $\delta_{N_\sigma}(\varepsilon) = \Phi\left(\frac{1}{2\sigma} - \varepsilon\sigma\right) - e^\varepsilon \Phi\left(-\frac{1}{2\sigma} - \varepsilon\sigma\right)$, where Φ is the CDF of the standard Normal distribution.

The dominating pair of the Gaussian mechanism N_σ is simply $\mathcal{N}(1, \sigma^2)$ and $\mathcal{N}(0, \sigma^2)$ (Zhu et al., 2022), which implies N_σ is dominated by the random variable $\frac{1}{\sigma}\mathcal{N}(0,1) + \frac{1}{2\sigma^2}$.

We note that in the case of the Gaussian mechanism, the PRD is simply the log-normal random variable, and the PRD of the random allocation is simply the sum of T such random variables. Formally, stating Corollary 3.3 for the Gaussian case yields the following expression, which is used in our experiments.

Corollary A.3. *Given $t \in \mathbb{N}$; $\sigma > 0$, the random allocation scheme over the Gaussian mechanism N_σ is dominated by $L_{\bar{\mu}_t, Q^t} = \ln\left(e^{\frac{1}{\sigma}\mathcal{N}(0,1) + \frac{1}{\sigma^2}} + \sum_{i \in [t-1]} e^{\frac{1}{\sigma}\mathcal{N}(0,1)}\right) - \ln(t) - \frac{1}{2\sigma^2}$ in the remove direction, and $L_{Q^t, \bar{\mu}_t} = \ln(t) + \frac{1}{2\sigma^2} - \ln\left(\sum_{i \in [t]} e^{\frac{1}{\sigma}\mathcal{N}(0,1)}\right)$ in the add direction.*

A.3 SUBSAMPLING

An additional advantage of providing a privacy bound in the form of a PLD, is that it can be used to further subsample and compose it. This is done using the following lemma which is stated in terms of PRD but naturally extends to PLD.

Lemma A.4 (PRD amplification by subsampling)**.** *Given two distributions P, Q denoting $P' := (1-\lambda)Q + \lambda P$ and $Q' := (1-\lambda)P + \lambda Q$ we have for any ω,*

$$\mathcal{R}(\omega; P', Q) = 1 - \lambda + \lambda \mathcal{R}(\omega; P, Q) \quad \text{and} \quad \mathcal{R}(\omega; P, Q') = \frac{1}{1 - \lambda + \lambda \mathcal{R}(\omega; Q, P)},$$

which implies that for any $r \in \mathbb{R}$,

1. *$F^{\mathcal{R}}_{P',Q}(r) = (1-\lambda)(1 - F^{\mathcal{R}}_{Q,P}(1/r')) + \lambda F^{\mathcal{R}}_{P,Q}(r')$ where $r' := 1 + \frac{r-1}{\lambda}$.*
2. *$F^{\mathcal{R}}_{P,Q'}(r) = F^{\mathcal{R}}_{P,Q}(r')$ where $r' := \frac{\lambda r}{1 - r(1-\lambda)}$.*

Proof. We first provide an explicit relation of the privacy ratio.

$$\mathcal{R}(\omega; P', Q) = \frac{P'(\omega)}{Q(\omega)} = \frac{(1-\lambda)Q(\omega) + \lambda P(\omega)}{Q(\omega)} = (1-\lambda) + \lambda \frac{P(\omega)}{Q(\omega)} = (1-\lambda) + \lambda \mathcal{R}(\omega; P, Q)$$

$$\mathcal{R}(\omega; P, Q') = \frac{P(\omega)}{Q'(\omega)} = \frac{P(\omega)}{(1-\lambda)P(\omega) + \lambda Q(\omega)} = \frac{1}{(1-\lambda) + \lambda \frac{Q(\omega)}{P(\omega)}} = \frac{1}{(1-\lambda) + \frac{\lambda}{\mathcal{R}(\omega; P, Q)}}$$

Next we use in to provide a similar relation for the PRD.

$$\begin{aligned}
F^{\mathcal{R}}_{P',Q}(r) &= \mathbb{P}_{\omega \sim P'}\left(\mathcal{R}(\omega; P', Q) \leq r\right) \\
&= \mathbb{P}_{\omega \sim P'}\left((1-\lambda) + \lambda \mathcal{R}(\omega; P, Q) \leq r\right) \\
&= \mathbb{P}_{\omega \sim P'}\left(\mathcal{R}(\omega; P, Q) \leq 1 + (r-1)/\lambda + \lambda\right) \\
&= \mathbb{P}_{\omega \sim P'}\left(\mathcal{R}(\omega; P, Q) \leq r'\right) \\
&= (1-\lambda) \mathbb{P}_{\omega \sim Q}\left(\mathcal{R}(\omega; P, Q) \leq r'\right) + \lambda \mathbb{P}_{\omega \sim P}\left(\mathcal{R}(\omega; P, Q) \leq r'\right) \\
&= (1-\lambda) \mathbb{P}_{\omega \sim Q}\left(\mathcal{R}(\omega; Q, P) \geq 1/r'\right) + \lambda \mathbb{P}_{\omega \sim P}\left(\mathcal{R}(\omega; P, Q) \leq r'\right) \\
&= (1-\lambda)(1 - F^{\mathcal{R}}_{Q,P}(1/r')) + \lambda F^{\mathcal{R}}_{P,Q}(r')
\end{aligned}$$

Similarly,

$$\begin{aligned}
F^{\mathcal{R}}_{P,Q'}(r) &= \underset{\omega\sim P}{\mathbb{P}}\left(\mathcal{R}\left(\omega;P,Q'\right)\leq r\right)\\
&= \underset{\omega\sim P}{\mathbb{P}}\left(\frac{1}{(1-\lambda)+\frac{\lambda}{\mathcal{R}(\omega;P,Q)}}\leq r\right)\\
&= \underset{\omega\sim P}{\mathbb{P}}\left(\mathcal{R}\left(\omega;P,Q\right)\leq \frac{r\lambda}{1-r(1-\lambda)}\right)\\
&= \underset{\omega\sim P}{\mathbb{P}}\left(\mathcal{R}\left(\omega;P,Q\right)\leq r'\right)\\
&= F^{\mathcal{R}}_{P,Q}(r')
\end{aligned}$$

□

While this lemma is stated in terms of a PRD (PLD), it holds for for any random variable that stochastically dominates a PRD (PLD). Notice that $F^{\mathcal{R}}_{P',Q}$ depends not only on $F^{\mathcal{R}}_{P,Q}$ but on $F^{\mathcal{R}}_{Q,P}$ as well. When using stochastically dominating random variables this requires either maintaining a random variable lower bounding $F_{Q,P}$ or using the fact that

$$F^{\mathcal{R}}_{Q,P}(r)=\int_{-\infty}^{r}f^{\mathcal{R}}_{Q,P}(x)dx=\int_{-\infty}^{r}\frac{f_{P,Q}(x)}{x}dx,$$

so $F^{\mathcal{R}}_{Q,P}$ can be numerically computed using only access to $F^{\mathcal{R}}_{P,Q}$ or its upper bound.

Implementing this amplification in practice requires maintaining simultaneous bounds on $F^{\mathcal{R}}_{Q,P}(1/r')$ and $F^{\mathcal{R}}_{Q,P}(1/r')$

B Full implementation details

In this section we provide detailed description of the implementation.

We start by describing the range normalization building block (Algorithm 1). This function is used to set the range of the convolved random variable, such that it looses at most β probability mass.

Algorithm 1 Range renormalization: $\text{Renorm}(\bar{x}_1, \bar{p}_1, \bar{x}_2, \bar{p}_2, \beta)$

Require: $\bar{x}_1, \bar{p}_1, \bar{x}_2, \bar{p}_2, \beta$

$n \leftarrow |\bar{x}_1|$ ▷ Assume $|\bar{x}_1| = |\bar{p}_1| = |\bar{x}_2| = |\bar{p}_2|$

$i_1^{\min} \leftarrow \arg\max_{i \in [n]} \left(\sum_{j=1}^{i} \bar{p}_1[j] \leq \sqrt{\beta/2} \right), \quad i_1^{\max} \leftarrow \arg\min_{i \in [n]} \left(\sum_{j=i}^{n} \bar{p}_1[j] \leq \sqrt{\beta/2} \right)$

$i_2^{\min} \leftarrow \arg\max_{i \in [n]} \left(\sum_{j=1}^{i} \bar{p}_2[j] \leq \sqrt{\beta/2} \right), \quad i_2^{\max} \leftarrow \arg\min_{i \in [n]} \left(\sum_{j=i}^{n} \bar{p}_2[j] \leq \sqrt{\beta/2} \right)$

$(x_{\min}, x_{\max}) \leftarrow \left(\bar{x}_1[i_1^{\min}] + \bar{x}_2[i_2^{\min}], \bar{x}_1[i_1^{\max}] + \bar{x}_1[i_2^{\max}] \right)$

$(y_{\min}, y_{\max}) \leftarrow (\ln(x_{\min}), \ln(x_{\max}))$

$\Delta \leftarrow (y_{\max} - y_{\min})/(n-1)$

$\bar{y} \leftarrow [y_{\min} + (i-1)\Delta]_{i=1}^{n}$

$\bar{x}_{\text{out}} \leftarrow [e^{y_i}]_{i=1}^{n}$

return $\bar{x}_{\text{out}}$

Next we describe the convolution step (Algorithm 2). Since the product grid is not identical to the new chosen grid, the probability is assigned such that the resulting random variable stochastically dominates the convolution.

Algorithm 2 Distribution convolution: $\text{Conv}(\bar{x}_1, \bar{p}_1, \bar{x}_2, \bar{p}_2, \beta)$

Require: $\bar{x}_1, \bar{p}_1, \bar{x}_2, \bar{p}_2, \beta$, dir

$\bar{x}_{\text{new}} \leftarrow \text{Renorm}(\bar{x}, \bar{p}, \bar{x}', \bar{p}', \beta)$

if dir = 'lower' **then**

$$\bar{P}_{\text{new}} \leftarrow \{0\} \cup \left[\sum_{\bar{x}_1[j] + \bar{x}_2[k] \leq \bar{x}_{\text{new}}[i]} \bar{p}_1[j] \cdot \bar{p}_2[k] \right]_{i=1}^{n}$$

$$\bar{p}_{\text{new}} \leftarrow \left[\bar{P}_{\text{new}}[i] - \bar{P}_{\text{new}}[i-1] \right]_{i=1}^{n}$$

else

$$\bar{P}_{\text{new}} \leftarrow \left[\sum_{\bar{x}_1[j] + \bar{x}_2[k] \leq \bar{x}_{\text{new}}[i]} \bar{p}_1[j] \cdot \bar{p}_2[k] \right]_{i=1}^{n} \cup \{1\}$$

$$\bar{p}_{\text{new}} \leftarrow \left[\bar{P}_{\text{new}}[i+1] - \bar{P}_{\text{new}}[i] \right]_{i=1}^{n}$$

end if

return $\bar{x}_{\text{new}}, \bar{p}_{\text{new}}$

Next, we use this function to create a self-convolution function, which given a distribution and number of convolutions t computes the convolution of the distribution with itself t times (Algorithm 3). This is done in $\log(t)$ steps by computing the self convolution for all powers of 2 that are $\leq t$, and using them to compose t times.

Using the convolution and self convolution functions, we can define the full algorithm (Algorithm 4 for the remove direction and 5 for add). Both algorithms start by computing a discrete random variable upper (lower) bounding the true PRD over a geometrically spaced grid (see proof of Theorem 3.4) then self compose it t times (in the case of the remove direction, one of the t random variables is sampled w.r.t. the first distribution following Corollary 3.3). Finally, the PRD is converted to a PLD by taking the $\ln$ ($-\ln$ in the add direction). We note that the remove direction requires maintaining an upper bound at each step, while the add direction requires a lower bound.

Remark B.1. Replacing lower bounds by upper bounds and vice versa, the same algorithm can be used to provide tight numerical lower bounds on the PLD. All results of Theorem 3.4 extend to this direction as well.

Algorithm 3 multi-conv$(\bar{x}, \bar{p}, t, \beta, \text{dir})$

Require: $\bar{x}$, $\bar{p}$, t, β, dir
  $(\bar{x}_{\text{base}}, \bar{p}_{\text{base}}) \leftarrow (\bar{x}, \bar{p})$
  init $\leftarrow$ False
  while $t > 0$ **do**
    if t is odd **then**
      if init **then**
        $(\bar{x}_{\text{acc}}, \bar{p}_{\text{acc}}) \leftarrow \text{Conv}(\bar{x}_{\text{base}}, \bar{p}_{\text{base}}, \bar{x}_{\text{acc}}, \bar{p}_{\text{acc}}, \beta, \text{dir})$
      else
        $(\bar{x}_{\text{acc}}, \bar{p}_{\text{acc}}) \leftarrow (\bar{x}_{\text{base}}, \bar{p}_{\text{base}})$
      end if
    end if
    $(\bar{x}_{\text{base}}, \bar{p}_{\text{base}}) \leftarrow \text{Conv}(\bar{x}_{\text{base}}, \bar{p}_{\text{base}}, \bar{x}_{\text{base}}, \bar{p}_{\text{base}}, \beta, \text{dir})$
    $t \leftarrow \lfloor t/2 \rfloor$
  end while
    return $\bar{x}_{\text{acc}}, \bar{p}_{\text{acc}}$

Algorithm 4 Random allocation numerical accounting (remove)

Require: P, Q, t, α, β
  $\beta' \leftarrow \beta/(2t)$, $\alpha' \leftarrow \alpha/(2 \cdot \ln(t))$
  $(l_{\min}, l_{\max}) \leftarrow \left(-(F^{\ell}_{Q,P})^{-1}(1-\beta/2), -(F^{\ell}_{Q,P})^{-1}(\beta/2)\right)$
  $n \leftarrow \lceil (l_{\max} - l_{\min})/\alpha' \rceil + 1$
  $\bar{r} \leftarrow \left[e^{l_{\min}+(i-1)\cdot\alpha'}\right]_{i=1}^{n}$ ▷ Quantization to bins of constant relative width
  $\bar{Q} \leftarrow \{0\} \cup \left[F^{\mathcal{R}}_{Q,P}(1/r_i)\right]_{i=1}^{n}$
  $\bar{q} \leftarrow \left[\bar{Q}[i] - \bar{Q}[i-1]\right]_{i=1}^{n}$ ▷ (α', β')-accurate privacy ratio bound of $R_{P,Q}$
  $(\bar{r}_{\text{conv}}, \bar{q}_{\text{conv}}) \leftarrow$ multi-conv$(\bar{r}, \bar{q}, t-1, \beta', \text{'upper'})$ ▷ Privacy ratio bound of the $t-1$-self convolution
  $\bar{P} \leftarrow \{0\} \cup \left[F^{\mathcal{R}}_{P,Q}(r_i)\right]_{i=1}^{n}$
  $\bar{p} \leftarrow \left[\bar{P}[i] - \bar{P}[i-1]\right]_{i=1}^{n}$ ▷ (α', β')-accurate privacy ratio bound of $1/R_{Q,P}$
  $(\bar{r}_{\text{final}}, \bar{p}_{\text{final}}) \leftarrow$ conv$(\bar{r}, \bar{p}, \bar{r}_{\text{conv}}, \bar{q}_{\text{conv}}, \beta', \text{'upper'})$ ▷ Privacy ratio bound of its convolution with the previous
  $\bar{l}_{\text{final}} \leftarrow [\ln(\bar{r}_{\text{final}}[i])]_{i=1}^{n}$ ▷ Privacy ratio to privacy loss
    return $\bar{l}_{\text{final}}, \bar{p}_{\text{final}}$

Algorithm 5 Random allocation numerical accounting (add)

Require: P, Q, t, α, β
  $\beta' \leftarrow \beta/(2t)$, $\alpha' \leftarrow \alpha/(2 \cdot \ln(t))$
  $(l_{\min}, l_{\max}) \leftarrow \left(-(F^{\ell}_{Q,P})^{-1}(1-\beta/2), -(F^{\ell}_{Q,P})^{-1}(\beta/2)\right)$
  $n \leftarrow \lceil (l_{\max} - l_{\min})/\alpha' \rceil + 1$
  $\bar{r} \leftarrow \left[e^{l_{\min}+(i-1)\cdot\alpha'}\right]_{i=1}^{n}$ ▷ Quantization to bins of constant relative width
  $\bar{Q} \leftarrow \{0\} \cup \left[F^{\mathcal{R}}_{Q,P}(1/r_i)\right]_{i=1}^{n}$
  $\bar{q} \leftarrow \left[\bar{Q}[i] - \bar{Q}[i-1]\right]_{i=1}^{n}$ ▷ (α', β')-accurate privacy ratio bound of $R_{P,Q}$
  $(\bar{r}_{\text{conv}}, \bar{q}_{\text{conv}}) \leftarrow$ multi-conv$(\bar{r}, \bar{q}, t, \beta')$ ▷ Privacy ratio bound of the $t-1$-self convolution
  $(\bar{r}_{\text{final}}, \bar{p}_{\text{final}}) \leftarrow$ conv$(\bar{r}, \bar{p}, \bar{r}_{\text{conv}}, \bar{q}_{\text{conv}}, \beta', \text{'lower'})$ ▷ Privacy ratio bound of its convolution with the previous
  $\bar{l}_{\text{final}} \leftarrow [-\ln(\bar{r}_{\text{final}}[i])]_{i=1}^{n}$ ▷ Privacy ratio to privacy loss
    return $\bar{l}_{\text{final}}, \bar{p}_{\text{final}}$

C EXPERIMENTAL RESULTS

In this section we provide several additional results. Figure 5 is an extended version of Figure 2. It follows the setting used by Chua et al. (2024a) to showcase their results. The number of steps is derived from the size of the training set and choice of batch size in their experimental results.

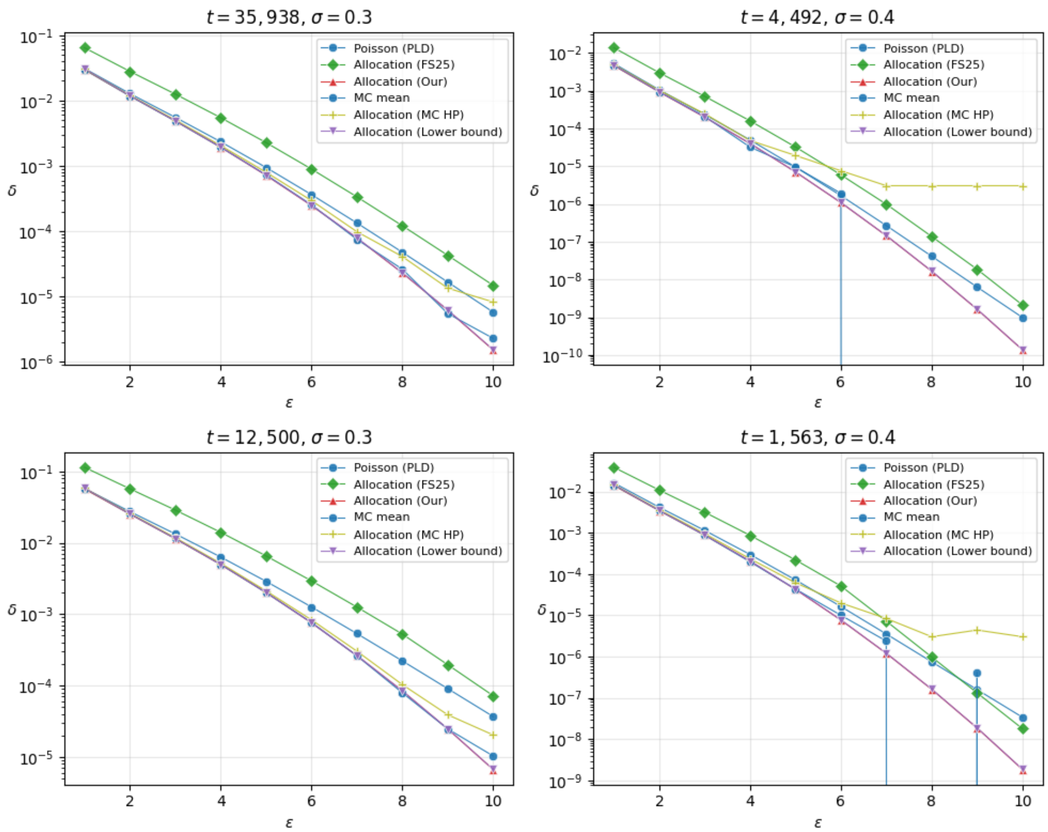

Figure 5: Comparison of the privacy profile of the Poisson scheme and various bounds for the random allocation scheme; the combined methods by Feldman & Shenfeld (2025), the high probability and the average estimations using Monte Carlo simulation and the lower bound by Chua et al. (2024a), and our numerical method, following the setting in Chua et al. (2024a) (detailed description can be found in Appendix C).

The Monte Carlo results were computed using importance sampling with 10^6 samples and 95% confidence. We note that the computation for the results derived by Chua et al. (2024a) was performed in parallel on a cluster of 60 CPU machines.

While all numerical examples in this work show superior privacy guarantees for random allocation relative to Poisson sampling, the Poisson scheme does not dominate random allocation for the same parameters. This was first proven theoretically by Chua et al. (2024a) for the limit of $\varepsilon \to 0$ and $\varepsilon \to \infty$. Figure 6 provides a clear demonstration of this fact.

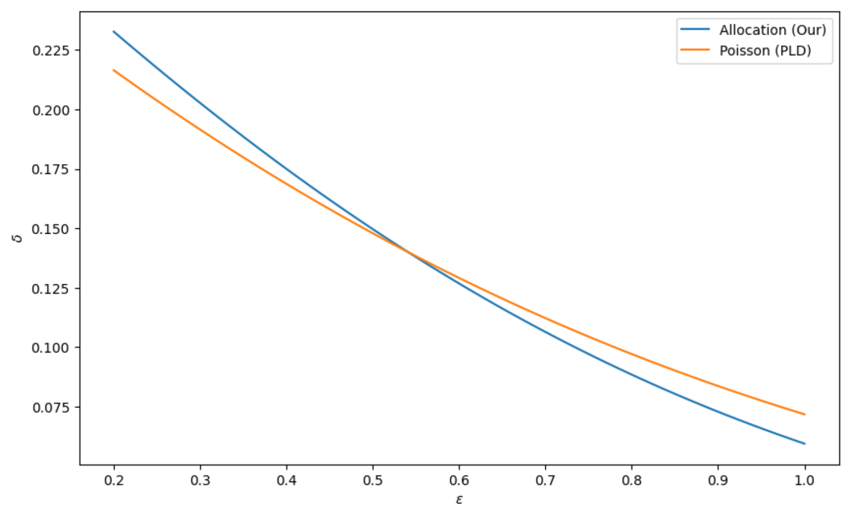

Figure 6: Privacy profile of the Poisson and random allocation schemes for $\sigma = 1.0$, $t = 2$, clearly demonstrating they do not dominate each other.