Sampling-Free Privacy Accounting for Matrix Mechanisms under Random Allocation

Jan Schuchardt
Machine Learning Research, Morgan Stanley
jan.a.schuchardt@morganstanley.com

Nikita Kalinin
Institute of Science and Technology Austria
nikita.kalinin@ist.ac.at

Abstract

We study privacy amplification for differentially private model training with matrix factorization under random allocation (also known as the balls-in-bins model). Recent work by Choquette-Choo et al. (2025) proposes a sampling-based Monte Carlo approach to compute amplification parameters in this setting. However, their guarantees either only hold with some high probability or require random abstention by the mechanism. Furthermore, the required number of samples for ensuring (ε, δ)-DP is inversely proportional to δ. In contrast, we develop sampling-free bounds based on Rényi divergence and conditional composition. The former is facilitated by a dynamic programming formulation to efficiently compute the bounds. The latter complements it by offering stronger privacy guarantees for small ε, where Rényi divergence bounds inherently lead to an over-approximation. Our framework applies to arbitrary banded and non-banded matrices. Through numerical comparisons, we demonstrate the efficacy of our approach across a broad range of matrix mechanisms used in research and practice.

1 Introduction

Differential privacy (DP) [18] offers one of the most well-established and mathematically rigorous definitions of privacy in statistical learning. In the context of deep learning, it is typically enforced via differentially private stochastic gradient descent (DP-SGD) [1]. In DP-SGD, per-example gradients are clipped to bound the contribution of any individual data point, and Gaussian noise is added to guarantee privacy. While the injected noise inevitably degrades model utility, this loss can be mitigated through privacy *amplification by subsampling* [29]. Specifically, by randomly sampling a subset of the data at each iteration, we can achieve the same privacy guarantees with less additive noise. This improvement arises from the adversary's uncertainty about which examples are included in each batch, introducing additional randomness that strengthens the overall privacy guarantees.

Several subsampling schemes have been studied in the context of differentially private training. One of the most commonly used is Poisson subsampling [1, 32, 25, 39], in which each data point is independently included in a batch with a fixed probability chosen to achieve a desired expected batch size. However, this scheme poses practical challenges for implementation [27, 6]. In particular, for large datasets and long training runs, two issues arise. First, drawing an independent random subset at every iteration is often computationally infeasible. Second, subsampling can lead to uneven data coverage, with a substantial fraction of examples (approximately $1/e$) not appearing within an epoch, and may increase the variance of gradients [20]. As a result, a common but problematic practice is to train on shuffled data while reporting privacy guarantees derived for Poisson subsampling [27].

A principled alternative that has emerged to address this issue is the "balls-in-bins" scheme [13, 12], also known as balanced iteration subsampling [15] or as random allocation [20, 21]. This approach predefines batches for an entire epoch and reuses them throughout training, ensuring a fixed number of participations per data point. As a result, it mitigates both the per-iteration sampling overhead

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and the uneven within-epoch data coverage, while requiring a noise multiplier comparable to that of Poisson subsampling to attain the same privacy guarantees.

Beyond better subsampling schemes, another principled approach to improve upon the utility DP-SGD is to introduce correlated noise. The key idea is to add more noise at each iteration while carefully correlating it across steps so that the aggregate noise has lower variance than in standard DP-SGD, thereby improving model utility. The most common way to implement such noise correlations is through the matrix factorization mechanism [28, 14, 10, 9], where Gaussian noise Z is correlated across iterations via the linear transformation C^{-1}, i.e., sampling $C^{-1}Z$, and then adding it to the gradients. Matrix factorization, however, can be impractical at scale, motivating memory-efficient structured variants. Prior work proposes banded matrix factorizations [23, 30], where the strategy matrix C is banded and Toeplitz; banded-inverse factorizations [24], where C^{-1} is banded; and hybrid constructions such as the Buffered Linear Toeplitz (BLT) matrix [17, 31]. These designs enable correlated-noise mechanisms with substantially reduced memory requirements.

Privacy amplification by subsampling and noise correlation techniques can be combined. This has been demonstrated for banded matrix mechanisms under Poisson subsampling [9, 11], and more recently for general correlation matrices using balls-in-bins subsampling [12]. The latter, however, relies on Monte Carlo estimation, which can require an intractably large number of samples in the high-privacy regime and thus become computationally expensive. Moreover, it does not directly yield deterministic DP guarantees, but rather privacy bounds that only hold with high probability.

Motivated by the recent independent works of Feldman & Shenfeld [20, 21] and Dong et al. [15], who derived privacy bounds for DP-SGD under balls-in-bins batching without Monte Carlo sampling, we propose two procedures for computing amplification bounds for matrix mechanisms under this batching scheme. Our methods avoid Monte Carlo sampling and instead compute the bounds explicitly, leveraging Rényi divergence via dynamic programming as well as conditional composition while accounting for the inter-step correlations unique to matrix mechanisms.

Contribution

- We introduce deterministic, Rényi divergence based and conditional composition based privacy analyses (accountants) for subsampled matrix mechanisms.
- For the special case of DP-SGD, we improve the computational complexity of Rényi accounting under the random allocation (balls-in-bins) batching scheme.

2 Background

2.1 Definitions of differential privacy

Differential Privacy (DP) is a widely accepted framework for providing rigorous privacy guarantees. We recall its definition below:

Definition 2.1 (Differential Privacy [18])**.** A randomized mechanism is said to provide (ε, δ)-differential privacy under a neighboring relation $\simeq$ if, for all datasets $D \simeq D'$, and for all measurable subsets S of the mechanism's output space, $\Pr[M(D) \in S] \leq e^{\varepsilon} \cdot \Pr[M(D') \in S] + \delta$.

In the following, we write $M(\cdot \mid D)$ for the distribution of random variable $M(D)$. We further use an equivalent definition of differential privacy that is more convenient for privacy accounting, i.e., tracking privacy over multiple applications of a mechanism.

Lemma 2.2 (Proposition 2 from [5])**.** *A mechanism provides (ε, δ)-DP under $\simeq$ if and only if, for all $D \simeq D'$, $H_{e^\varepsilon}\left(M(\cdot \mid D), M(\cdot \mid D')\right) \leq \delta$ with hockey-stick divergence*

$$H_\gamma(P||Q) = \mathbb{E}_{x\sim P}\left[\max\left\{1 - \gamma\frac{\mathrm{d}Q}{\mathrm{d}P}(x),\ 0\right\}\right]. \tag{1}$$

While (ε, δ)-DP corresponds to a bound on H_γ for a single choice of $\gamma = e^\varepsilon$, tight privacy accounting requires bounds for all $\gamma \geq 0$. In the following, we recall multiple equivalent [39] characterizations of this *privacy profile* that are immediately relevant to our work, beginning with dominating pairs:

Definition 2.3. We say that the pair of distributions $(\hat{P}, \hat{Q})$ dominates (P, Q), denoted by $(P, Q) \prec (\hat{P}, \hat{Q})$, if $H_\gamma(P||Q) \leq H_\gamma(\hat{P}, \hat{Q})$ for all $\gamma > 0$.

Definition 2.4 (Zhu et al. [39])**.** We say that $(\hat{P}, \hat{Q})$ are a *dominating pair* of mechanism M under neighboring relation $\simeq$ if $(M(\cdot \mid D), M(\cdot \mid D')) \prec (\hat{P}, \hat{Q})$ for all $D \simeq D'$.

Given a dominating pair (P, Q), the problem of privacy accounting reduces to quantifying the hockey-stick divergence $H_\gamma(P||Q)$. Specifically, the divergence does not directly depend on the (potentially high-dimensional) structure of (P, Q) but on their log-likelihood ratio, the "privacy loss":

Definition 2.5. The privacy loss random variable $L_{P,Q}$ of P, Q is $\log(\frac{\mathrm{d}P}{\mathrm{d}Q}(x))$ with $x \sim P$.

From Equation (1) and $\gamma \frac{\mathrm{d}P}{\mathrm{d}Q}(x) = e^{\log(\gamma) - L_{P,Q}}$, we see that the hockey-stick divergence quantifies how much the privacy loss random variable (PLRV) exceeds $\varepsilon = \log(\gamma)$ in expectation. While the distribution of $L_{P,Q}$ fully determines the privacy profile, it is often analytically more convenient to work with its moment generating properties. Specifically, the distribution is uniquely characterized by its cumulant generating function [39], which is proportional to the Rényi divergence: [1]

Definition 2.6. The Rényi divergence of P and Q is $R_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \mathbb{E}\left[\exp((\alpha - 1) L_{P,Q})\right]$.

2.2 (Subsampled) matrix mechanisms

Training a model with d parameters using differentially private stochastic gradient descent (DP-SGD) for k epochs with b batches per epoch (and thus $N = k \cdot b$ iterations) can be viewed as computing the noisy prefix sums $\mathbf{E}(\boldsymbol{G} + \boldsymbol{Z})$ row by row, where $\mathbf{E} \in \{0, 1\}^{N \times N}$ is the all-ones lower-triangular matrix; the entries of $\boldsymbol{Z} \in \mathbb{R}^{N \times d}$ are drawn i.i.d. from a normal distribution with standard deviation $\sigma > 0$; and $\boldsymbol{G} \in \mathbb{R}^{N \times d}$ contains the aggregated gradients for each batch, clipped to a fixed ℓ_2-norm.

Over the past few years, various works have theoretically and empirically demonstrated that improved estimation can be achieved by correlating the noise across iterations. Specifically, by introducing a so-called correlation matrix $\boldsymbol{C}^{-1} \in \mathbb{R}^{N \times N}$, the matrix factorization mechanism [28] adds correlated noise via $\mathbf{E}\big(\boldsymbol{G} + \boldsymbol{C}^{-1}\boldsymbol{Z}\big)$. This approach, however, requires adjustments to the privacy accounting. Without subsampling, it is equivalent (by the post-processing property) to applying the Gaussian mechanism to make $\boldsymbol{C}\boldsymbol{G}$ private by adding noise $\boldsymbol{Z}$, and it therefore requires computing the corresponding sensitivity [9]. The resulting scheme can offer superior utility: although the sensitivity may increase (and thus the noise scale must be adjusted accordingly), the total injected noise $\mathbf{E}\boldsymbol{C}^{-1}\boldsymbol{Z}$ can have a much smaller expected norm. By adding less noise to the intermediate models, training utility can improve; see the recent survey [34] for further discussion on selecting a correlation matrix.

The utility of the matrix factorization mechanism can be further improved by combining it with amplification by subsampling. A recent work by Choquette-Choo et al. [12] introduces an efficient balls-in-bins subsampling scheme, along with a corresponding privacy accountant that supports matrix factorization with an arbitrary positive lower-triangular *strategy* matrix $\boldsymbol{C}$.

Definition 2.7. In *balls-in-bins* subsampling with $k \in \mathbb{N}$ epochs and $b \in \mathbb{N}$ batches per epoch, a record contributes to iterations $(i, b + i, 2b + i, \ldots, (k - 1)b + i)$ with $i \sim \mathrm{Uniform}(\{1, \ldots, b\})$.

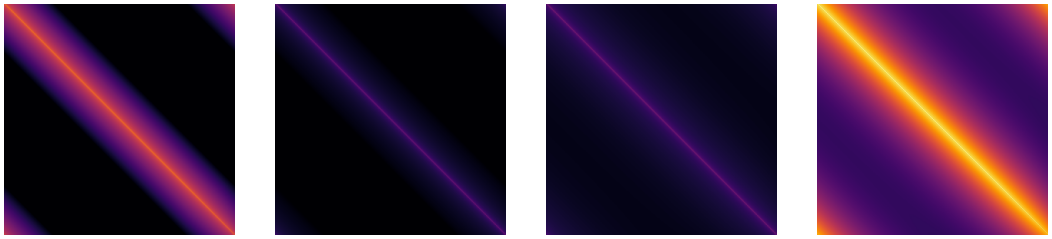
They further derive (almost) tight dominating pairs for matrix mechanism under this scheme.

Lemma 2.8. *[Lemma 3.2 from [12]] Assume lower-triangular, non-negative strategy matrix $\boldsymbol{C}$, gradient clipping norm $\Delta = 1$, and balls-in-bins sampling with $k \in \mathbb{N}$ epochs and $b \in \mathbb{N}$ batches per epoch. Define distributions $P = \frac{1}{b}\sum_{i=1}^{b} \mathcal{N}(\boldsymbol{m}_i, \sigma^2 \boldsymbol{I}_N)$ and $Q = \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_N)$ supported on $\mathbb{R}^N$ with $N = k \cdot b$ and mixture means $\boldsymbol{m}_i = \sum_{j=0}^{k-1} |\mathbf{C}|_{:,\, i+jb}$, Then the subsampled matrix mechanism is dominated by (P, Q) under the "remove" [2] relation and by (Q, P) under the "add" relation.*

However, the main algorithmic challenge in privacy accounting is not to just specify a dominating pair (P, Q), but to efficiently evaluate or bound the hockey-stick divergence $H_\gamma(P||Q)$. Monte Carlo privacy accountants [37] compute a bound by leveraging the equality of $H_\gamma(P||Q)$ to the expectation $\mathbb{E}\left[\max\{1 - e^{\log(\gamma) - L_{P,Q}}, 0\}\right]$ we discussed in Section 2.1: They sample from the privacy loss distribution and determine a confidence interval on the expectation to show that the mechanism is (ε, δ)-DP with some high probability. This is however a qualitatively weaker guarantee than a

[1] $R_\alpha(P||Q)$ of dominating pair P, Q is not to be confused with Rényi DP [32], which is strictly less informative, see [39].

[2] The original Lemma 3.2 swaps the "add" and "remove" direction, which appears to be a typing error, see Appendix H.



(a) BandMF, $p = 64$ (b) BSR, $p = 64$ (c) BISR, $p = 64$ (d) BandInvMF, $p = 4$

Figure 1: The Gram matrix $\mathbf{G}$ for matrices of size $N = 3000$ with $k = 10$. For banded matrices $\mathbf{C}$, we consider Banded Matrix Factorization (BandMF) [30] and Banded Square Root Factorization (BSR) [23] with bandwidth $p = 64$. For banded inverse matrices, we consider Banded Inverse Matrix Factorization (BandInvMF) and Banded Inverse Square Root (BISR) [24] with bandwidths $p = 4$ and $p = 64$ accordingly. We note that although the matrix $\mathbf{C}$ is banded, the matrix $\mathbf{G}$ is not; rather, it is *cyclically banded*, with the corner blocks filled by strictly positive entries. Moreover, we observe that banded inverse matrices show similar behavior: their Gram matrices have entries that decay rapidly outside the cyclic band.

deterministic bound on δ.[3] One can recover deterministic guarantees, but this requires modification of the mechanism with random abstentions.

Furthermore, to provide high probability bounds for the privacy level, the instantiation for matrix mechanisms in [12] requires a sample size inversely proportional to δ. Thus, showing $(\varepsilon, 10^{-8})$-DP is 100 000 times more expensive than showing $(\varepsilon, 10^{-3})$-DP. So far, no sampling-free method is available to overcome these inherent limitations of Monte Carlo accounting for matrix mechanisms.

3 Problem statement

Our goal is to develop methods that enable efficient, deterministic privacy accounting for matrix mechanisms under random allocation. The main challenge is that the multidimensional mixture P in Lemma 2.8 results in a privacy loss random variable (PLRV) $L_{P,Q}$ that cannot be decomposed into simpler PLRVs, ruling out a central technique in privacy accounting.

Concretely, standard privacy accountants explicitly (e.g. [36, 25, 39]) or implicitly (e.g. [1]) use that the dominating pair (P, Q) for any N-fold adaptive composition of mechanisms can be factorized as $P = \bigtimes_{n=1}^{N} P_n$ and $Q = \bigtimes_{n=1}^{N} Q_n$, which makes the PLRV a simple sum $L_{P,Q} = \sum_{n=1}^{N} L_{P_n,Q_n}$. However, even without noise correlation ($\boldsymbol{C} = \boldsymbol{I}_N$), random allocation breaks this independence assumption, as can be seen from our P in Lemma 2.8 being a mixture distribution. In concurrent work, Feldman & Shenfeld [21] show that $L_{P,Q}$ can nevertheless be decomposed into a function of N simpler PLRVs by exploiting that, for $b = N$, each of the b mixture components only differs from Q in a single dimension ($\boldsymbol{m}_i = \boldsymbol{e}_i$). However, as we discuss more formally in Appendix I, matrix mechanisms ($\boldsymbol{C} \neq \boldsymbol{I}_N$) break this assumption and make their analysis inapplicable. In short, *random allocation prevents decomposition into per-step privacy losses, while noise correlation further prevents decomposition into per-component privacy losses*.

In the following, we propose two complementary algorithms to overcome this challenge: The first one analyzes the moment generating function of $L_{P,Q}$ instead of its distribution (Rényi accountant). The second one constructs a pessimistic dominating pair consisting of product distributions, allowing us to apply standard accounting techniques to obtain sound bounds (conditional composition accountant).

4 Rényi accountant

The main obstacle in evaluating the hockey-stick divergence $H_\gamma(P||Q)$ for our non-factorizable dominating pair from Lemma 2.8 is that it requires integration over the non-convex region $\frac{\mathrm{d}P}{\mathrm{d}Q}(x) > \gamma$. In contrast, the Rényi divergence can be evaluated by integrating a smooth function of the privacy

[3] Note that, despite a common misinterpretation, privacy parameter δ is not the probability of violating ε-DP.

loss over the entirety of $\mathbb{R}^N$, avoiding the need to explicitly characterize this region. In this section, we propose an efficient algorithm for evaluating or upper-bounding this integral. We then translate these Rényi divergence bounds at order α into (ε, δ)-differential privacy via the following theorem.

Theorem 4.1. *Prop. 12 in Canonne et al. [8]. Given two distributions P, Q, if $R_\alpha(P \,\|\, Q) \leq \rho$, then*

$$H_{e^\varepsilon}(P \,\|\, Q) \leq \frac{1}{\alpha - 1} e^{(\alpha-1)(\rho-\varepsilon)} \left(1 - \frac{1}{\alpha}\right)^\alpha. \tag{2}$$

In the following lemma, we compute the Rényi divergence for the dominating pair from Lemma 2.8, i.e., $\hat{P} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{N}(\mathbf{m}_i, \sigma^2 I)$ and $\hat{Q} = \mathcal{N}(0, \sigma^2 I)$, in the "remove" direction.

Lemma 4.2. *The Rényi divergence in the "remove" direction $R_\alpha(\hat{P} \,\|\, \hat{Q})$ is given by*

$$\frac{1}{\alpha - 1} \log \sum_{(r_1,\ldots,r_\alpha)\in[1,b]^\alpha} \exp\left(\sum_{\substack{j_1,j_2=1 \\ j_1\neq j_2}}^{\alpha} \frac{\mathbf{G}_{r_{j_1},r_{j_2}}}{2\sigma^2}\right) - \frac{\alpha \log b}{1-\alpha}, \tag{3}$$

where G is the Gram matrix with entries $\mathbf{G}_{i,j} = \langle \mathbf{m}_i, \mathbf{m}_j \rangle$.

For a general Gram matrix $\mathbf{G}$, there is no efficient way to evaluate this sum in equation 3. More precisely, in Lemma D.4, we prove by a reduction from clique counting that computing this sum is $\#P$-complete, where $\#P$ is the class of counting problems associated with NP decision problems. However, if we impose structural restrictions on $\mathbf{C}$, for instance by requiring it to be p banded, then $\mathbf{G}$ inherits additional structure and becomes *cyclic banded* (see Figure 1). In this case, the sum can be computed via dynamic programming in time $O\big(bp\,\alpha^{2p}\big)$. In particular, for $p = 1$, which corresponds to the DP-SGD case, this yields a runtime of $O(b\alpha^2)$, improving upon the exponential $O(2^\alpha)$ complexity reported by Feldman & Shenfeld [20]. In Appendix A.9, we verify this dramatic speedup via wall-clock runtime measurements.

If $\mathbf{C}$ is not naturally p banded, as is the case for banded inverse matrices in Kalinin et al. [24], the procedure above is no longer efficient. Nevertheless, many such matrices are effectively close to banded in the sense that entries sufficiently far from the diagonal are small (see Figure 1). This observation leads to the following bound. Let

$$\tau \coloneqq \max_{\min(|i-j|, b-|i-j|)\geq p} \mathbf{G}_{i,j}. \tag{4}$$

Then $\mathbf{G}$ admits the elementwise bound $\mathbf{G} \leq \mathbf{G}^p + \tau\mathbf{E}$, where $\mathbf{G}^p$ is the p cyclic banded truncation of $\mathbf{G}$ and $\mathbf{E}$ denotes the all ones matrix. Consequently, by adding an extra term $\frac{\alpha\tau}{2\sigma^2}$ to the Rényi divergence, we obtain a computable upper bound on the privacy loss (for an empirical demonstration of the effect of truncation on bound tightness, see Appendix A.4.1).

Combining these observations, Algorithm 4, which we present in Appendix C for space reasons, computes the Rényi divergence in the "remove" direction exactly when $\mathbf{C}$ is p banded, and an upper bound otherwise. Formally:

Lemma 4.3. *Algorithm 4 computes the Rényi divergence based value in eq. 3 exactly for a p-banded strategy matrix $\mathbf{C}$, and returns an upper bound otherwise. The algorithm runs in time $O\big(bp\alpha^{2p}\big)$ and requires $O(\alpha^p + bp)$ memory.*

Algorithm 4 evaluates the sum in equation 3 exactly when the Gram matrix $\mathbf{G}$ is p cyclically banded, and returns an upper bound otherwise by truncating $\mathbf{G}$ to its p cyclically banded part $\mathbf{G}^{(p)}$ and adding the correction term $\frac{\tau\alpha}{2\sigma^2}$. The indices in equation 3 interact cyclically; we break this cycle by conditioning on the short prefix $\mathbf{l} = (l_0, \ldots, l_{p-2})$ and then compute the remaining contribution via a forward dynamic program. We glue the cycle back at the end by adding the interactions between indices $b-p+2, \ldots, b-1$ and $0, \ldots, p-2$. For numerical stability, all aggregations are performed using the log-sum-exp (LSE) operation. See the proof of Lemma 4.3 for a detailed description.

However, this is not sufficient to prove (ε, δ)-DP, since the notion of neighboring datasets also requires a bound in the "add" direction. We found it substantially harder to compute the divergence $\mathrm{R}_\alpha(\hat{Q} \,\|\, \hat{P})$ explicitly. Nevertheless, it admits a tight and easily computable upper bound, which we present in the following lemma and formalize in Algorithm 2 and Appendix C.

Algorithm 1 Rényi accountant full algorithm

Require: Correlation matrix $\mathbf{C}$, bandwidth p, separation b, parameter $\alpha \in \mathbb{N}_+$.

$\rho_{\text{remove}} \leftarrow \text{ReAcc}_{\text{remove}}(\mathbf{C}, p, b, \alpha)$ # Algorithm 4

$\rho_{\text{add}} \leftarrow \text{ReAcc}_{\text{add}}(\mathbf{C}, p, b, \alpha)$ # Algorithm 2

$\rho \leftarrow \max(\rho_{\text{remove}}, \rho_{\text{add}})$

$\delta(\alpha, \varepsilon) \leftarrow \exp\big((\rho - \varepsilon)(\alpha - 1)\big)(\alpha - 1)^{-1}(1 - 1/\alpha)^{\alpha}$

return $\delta(\alpha, \varepsilon)$

Algorithm 2 Rényi accountant "add" direction

Require: Correlation matrix C, bandwidth p, separation b, parameter $\alpha \in \mathbb{N}_+$.

$\mathbf{m}_i \leftarrow \sum_{j=0}^{k-1} |\mathbf{C}|_{:,\, i+jb}$

$\mathbf{G}_{i,j} \leftarrow \langle \mathbf{m}_i, \mathbf{m}_j \rangle \quad \forall i, j \in \{1, \ldots, b\}$

$\rho \leftarrow \frac{1}{2b\sigma^2} \sum_{j=1}^{b} \mathbf{G}_{j,j} \; + \; \frac{\alpha - 1}{2b^2\sigma^2} \sum_{i=1}^{b} \sum_{j=1}^{b} \mathbf{G}_{i,j}$

return ρ

Lemma 4.4. *The Rényi divergence in the "add" direction is bounded as*

$$\mathrm{R}_\alpha(\hat{Q} \| \hat{P}) \leq \frac{1}{2b\sigma^2} \sum_{j=1}^{b} \mathbf{G}_{j,j} \; + \; \frac{\alpha - 1}{2b^2\sigma^2} \sum_{i=1}^{b} \sum_{j=1}^{b} \mathbf{G}_{i,j}, \tag{5}$$

where $\mathbf{G}$ *is the Gram matrix with entries* $\mathbf{G}_{i,j} = \langle \mathbf{m}_i, \mathbf{m}_j \rangle$.

Taking the maximum over the "add" and "remove" directions, $\max\big(\mathrm{R}_\alpha(\hat{Q} \,\|\, \hat{P}),\, \mathrm{R}_\alpha(\hat{P} \,\|\, \hat{Q})\big)$, and converting it to a hockey stick divergence via Theorem 4.1 then completes the computation; see Algorithm 1. In practice, we found the "remove" direction bound to consistently dominate, which aligns with observations from prior work (see details in Appendix A.4.2). To run the algorithm, we must choose the Rényi order α that yields the tightest privacy guarantee. For practical values of ε and δ, the optimal α typically lies between 2 and 40. Theoretically, Theorem 4.1 suggests that for small δ, the optimal α grows logarithmically with $1/\delta$.

5 Conditional composition accountant

The proposed dynamic program allows for an exact evaluation of the Rényi divergence for any $\alpha_+ \in \mathbb{N}_+$.[4] These bounds can be tightly converted to privacy profiles $\delta(\varepsilon)$ via the inverse Laplace transform [39], but this is generally intractable. Conversion formulae such as Theorem 4.1 sacrifice some of this tightness, and are best suited for capturing the large deviation behavior of privacy loss $L_{P,Q}$ [2], i.e., "large ε". We therefore derive a complementary method that—as our experiments demonstrate—yields tighter bounds under strict privacy budgets, i.e., "small ε".

Concretely, recall from Section 3 that random allocation and noise correlation prevent decomposition of the PLRV $L_{P,Q}$ into multiple simpler PLRVs. This means that we cannot directly apply more recent privacy accountants that avoid the looseness of Rényi conversion formulae (e.g. PLD convolution [36, 25]). In principle, we could force a factorization via product rule $p(\boldsymbol{y}) = \prod_{n=1}^{N} p(y_n \mid \boldsymbol{y}_{1:n-1})$ and $q(\boldsymbol{y}) = \prod_{n=1}^{N} q(y_n \mid \boldsymbol{y}_{1:n-1})$ to derive dominating pairs (P_n, Q_n) with PLRV L_{P_n,Q_n} for each step n. However, by Definition 2.3, these dominating pairs would need to hold for *all* previous outcomes $\boldsymbol{y}_{1:n-1}$. This includes outcomes that provide certain evidence of $\boldsymbol{y}$ being drawn from a specific mixture component, i.e., a specific participation pattern chosen by random allocation. We would thus lose all privacy amplification. Instead, we propose to leverage that, due to Gaussianity, such outcomes are exceedingly unlikely, which allows us to retain the benefit of subsampling.

The tool we use for this purpose is conditional composition[5] It lets us bound the overall mechanism's privacy profile via a factorizable dominating pair for "good" outcomes that provide little evidence for the participation pattern while separately accounting for low-probability "bad" outcomes. The

[4] Real-valued orders α can be evaluated via interpolation, see Corollary 10 in [38]

[5] It is also referred to as posterior sampling in [20], following earlier work in [19].

hockey-stick divergence for this dominating pair can then be evaluated using standard accountants. In our work, we use the following formulation (for its relation to the general theorem, see Appendix E).

Lemma 5.1. *[Special case of Theorem 3.1 in [11]] Let P, Q be distributions on $\mathbb{R}^N$. For every step $n \in N$, choose a pair of distributions $P^{(n)}, Q^{(n)}$. Let $A^{(n)} \subseteq \mathbb{R}^N$ refer to all "good" outcomes $\tilde{y}$ such that the conditional distributions $P_{y_n|\tilde{y}_{1:n-1}}$ and $Q_{y_n|\tilde{y}_{1:n-1}}$ are dominated by $P^{(n)}, Q^{(n)}$. Define "bad" outcomes $E = \bigcup_{n=1}^{N} \overline{A^{(n)}}$ with probability $\delta_E = P(E)$. Then,*

$$H_\gamma(P||Q) \leq H_\gamma\left(\bigtimes_{n=1}^{N} P^{(n)} || \bigtimes_{n=1}^{N} Q^{(N)}\right) + \delta_E. \tag{6}$$

While Lemma 5.1 provides a valid composition rule, the challenge in applying it lies in (a) choosing appropriate per-step distributions $P^{(n)}, Q^{(n)}$, (b) finding a criterion for whether they are a valid dominating pair for the conditional distribution at each step n, (c) analyzing the probability of this criterion being violated. We address these challenges in Section 5.1. After that, we use these results to define an algorithm that automatically yields valid per-step dominating pairs for privacy accounting given a user-specified probability δ_E in Section 5.2. All proofs can be found in Appendix E.

5.1 Non-dominance probability given per-step distributions

Per-step distributions. From the definition of P, Q in Lemma 2.8 and the law of total probability, we know that for any previous outcome $\boldsymbol{y}_{1:n-1} \in \mathbb{R}^{n-1}$ the conditional density of step n is

$$p(y_n \mid \boldsymbol{y}_{1:n-1}) = \sum_{i=1}^{b} \mathcal{N}(y_n \mid m_{i,n}, \sigma) p(z = i \mid \boldsymbol{y}_{1:n-1}), \quad q(y_n \mid \boldsymbol{y}_{1:n-1}) = \mathcal{N}(y_n \mid 0, \sigma), \tag{7}$$

with non-negative means $m_{i,n} \in \mathbb{R}_+$ and $p(z = i) = \frac{1}{b}$, where b is the number of batches per epoch. Thus, a natural choice for our per-step distributions $P^{(n)}, Q^{(n)}$ that should be dominating is via

$$p^{(n)}(y_n) = \sum_{i=1}^{b} \mathcal{N}(y_n \mid m_{i,n}, \sigma) p^{(n)}(z = i), \quad q^{(n)}(y_n) = \mathcal{N}(y_n \mid 0, \sigma), \tag{8}$$

with some appropriate choice of categorical mixture weights $p^{(n)}(z = i)$. Note that $P^{(n)}$ and $Q^{(n)}$ in Equation (8) are fixed distributions. In particular, the mixture weights $p^{(n)}(z = i)$ do not depend on earlier outcomes $\boldsymbol{y}_{1:n-1}$, unlike the posterior mixture weights in Equation (7).

Dominance criterion. For any *fixed* earlier outcomes $\boldsymbol{y}_{1:n-1}$, Equation (7) and Equation (8) contain Gaussian mixtures with identical means but different weights. Intuitively, one pair dominates the other if it assigns more weight to larger mixture components, since these correspond to participation patterns with more privacy leakage. This intuition has been formally proven by Choquette-Choo et al. [11] in terms of stochastic dominance between priors. However, the following derived criterion in terms of *reverse hazard functions* is more convenient to work with for reasons we shall see shortly:

Lemma 5.2. *Assume non-decreasing mixture means $0 \leq m_{1,n} \leq \cdots \leq m_{b,n}$ in Equations* (7) *and* (8)*. If the posterior reverse hazard function is pointwise less or equal, i.e.,*

$$P(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) \leq P^{(n)}(z = i \mid z \leq i) \quad \textit{for all } 1 \leq i \leq b, \tag{9}$$

then the distributions from Equation (8) *dominate those from Equation* (7)*, i.e., $(P_{y_n|y_{1:n-1}}, Q_{y_n|y_{1:n-1}}) \prec (P^{(n)}_{y_n}, Q^{(n)}_{y_n})$ and $(Q_{y_n|y_{1:n-1}}, P_{y_n|y_{1:n-1}}) \prec (Q^{(n)}_{y_n}, P^{(n)}_{y_n})$.*

The benefit of this formulation is that it reduces the problem of testing for dominance to a standard machine learning problem: For each $i \in [b]$, we simply need to evaluate the probability of component i in a Gaussian mixture model with components $1, \ldots, i$. From a privacy perspective, Bayes law shows that Equation (9) is equivalent to testing a tail bound on a Gaussian mixture privacy loss:

Lemma 5.3. *Let $s(x) = (1+e^{-x})^{-1}$ be the logistic sigmoid function. For any $i \in [b]$, the conditional reverse hazard function is $P(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) = s(-\log(i-1) - L_i(\boldsymbol{y}_{1:n-1}))$ with*

$$L_i(\boldsymbol{y}_{1:n-1}) = \log\left(\frac{p(\boldsymbol{y}_{1:n-1} \mid z < i)}{p(\boldsymbol{y}_{1:n-1} \mid z = i)}\right) = \log\left(\frac{\sum_{j=1}^{i-1} \mathcal{N}(\boldsymbol{y}_{1:n-1} \mid \boldsymbol{m}_{j,:n-1}, \sigma^2 \boldsymbol{I}) \frac{1}{i-1}}{\mathcal{N}(\boldsymbol{y}_{1:n-1} \mid \boldsymbol{m}_{i,:n-1}, \sigma^2 \boldsymbol{I})}\right). \tag{10}$$

Non-dominance probability. Finally, we need to calculate the probability δ_E that our dominance criterion is violated ("bad" outcome $\bigcup_{n=1}^{N} \overline{A^{(n)}}$) under the randomness of the overall dominating pair from Lemma 2.8, i.e., $\boldsymbol{y} \sim P$ ("remove" relation) or $\boldsymbol{y} \sim Q$ ("add" relation). It follows from Lemma 5.3 that this corresponds to tail probabilities on Gaussian mixture privacy loss L_i:

Theorem 5.4. *Let $\boldsymbol{y} \sim R$ with $R = P$ ("remove") or $R = Q$ ("add') as defined in Lemma 2.8. Define the shorthand $\lambda_i = P^{(n)}(z = i \mid z \leq i)$ for the reverse hazard function. Choose thresholds τ_i with $\lambda_i = s(-\log(i-1) - \tau_i)$. Then, $\Pr[\overline{A^{(n)}}] \leq \sum_{i=1}^{b} \Pr_{\boldsymbol{y} \sim R}[L_i(\boldsymbol{y}_{1:n-1}) \leq \tau_i]$.*

Analytical bounds. Theorem 5.4 reduces our non-dominance probabilities to tail bounds on a ternary form of privacy loss $L_{\tilde{P},\tilde{Q},\tilde{R}}(\boldsymbol{x}) = \log(\mathrm{d}\tilde{P} \,/\, \mathrm{d}\tilde{Q})(\boldsymbol{x})$ where $\tilde{P}, \tilde{Q}, \tilde{R}$ are (mixtures of) Gaussians. However, if we could evaluate them exactly, we could also directly evaluate the privacy profile $\delta(\varepsilon)$ of our overall mechanism and would not need privacy accounting in the first place (see, e.g., Theorem 5 in [3]). Luckily, bounding the extreme tails (small δ_E) of privacy loss distributions is exactly what Rényi accounting excels at. We could thus adapt the methods from Section 4 to evaluate the Rényi divergence / MGF of $L_{\tilde{P},\tilde{Q},\tilde{R}}$ and compute a Chernoff bound (see Theorem 2 of Abadi et al. [1]). Instead, we choose to directly apply the AM-GM inequality underlying Lemma 4.4 to bypass the lossy Chernoff bound. This results in the following analytical tail bound (for "remove", see Theorem E.7).

Theorem 5.5. *Let $\boldsymbol{y} \sim R$ with $R = Q$ ("add') as defined in Lemma 2.8. Define standard normal CDF $\Phi : \mathbb{R} \to [0,1]$ and $\psi = \mathrm{Uniform}([i-1])$ Then $\Pr[L_i(\boldsymbol{y}_{1:n-1}) \leq \tau_i] \leq \Phi\left(\frac{\tau_i - \nu}{\xi}\right)$ with*

$$\nu = (||\boldsymbol{m}_{i,:n-1}||_2^2 - \mathbb{E}_{j\sim\psi}[||\boldsymbol{m}_{j,:n-1}||_2^2])] \,/\, (2\sigma^2), \qquad \xi = ||\boldsymbol{m}_{i:n-1} - \mathbb{E}_{j\sim\psi}[\boldsymbol{m}_{j,:n-1}]||_2 \,/\, \sigma.$$

5.2 Per-step distributions given non-dominance probability

Picking mixture weights for our per-step dominating pairs from Equation (8) by hand is not practical. We thus propose Algorithm 3, which returns a dominating pair given iteration n and significance $\beta = \Pr[\overline{A^{(n)}}]$. Via union bound, $\beta = \delta_E \,/\, N$ yields the desired non-dominance probability. Using our analytical bounds from Theorems 5.5 and E.7, the tails τ_i can be found via bisection.

Theorem 5.6. *For the "remove" relation, the distributions returned by Algorithm 3 dominate $P_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ and $Q_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ with probability $1-\beta$ under P. For the "add" relation, the returned distributions dominate $Q_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ and $P_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ with probability $1-\beta$ under Q.*

Complexity and amortization. When using Theorems 5.5 and E.7 with a constant number of bisection steps to obtain tail bounds τ_i, the overall complexity of evaluating Algorithm 3 for every iteration $n \in [N]$ is $\mathcal{O}(N^2b^2)$ for "remove" and $\mathcal{O}(N^2b)$ for "add". However, most of the computational cost in finding the τ_i arises from terms that are linear in noise multiplier $\sigma \in \mathbb{R}_+$. Thus, subsequent evaluations for different σ can be performed in $\mathcal{O}(Nb^2)$ ("remove) or $\mathcal{O}(Nb\log b)$ ("add"). This can be used to significantly speed up mechanism calibration, where σ is adjusted to attain desired privacy parameters (ε, δ) (see Appendix E.4 for proofs and Appendix A.4 for empirical runtimes).

Algorithm 3 Conditional composition pairs

Require: Mixture means $\boldsymbol{m} \in \mathbb{R}_+^{b\times N}$, iteration n, batches per epoch b, significance β, relation r
  $\pi \leftarrow \mathrm{argsort_asc}(\boldsymbol{m}_{:,n})$
  $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_b \leftarrow \boldsymbol{m}_{\pi(1),:n-1}, \ldots, \boldsymbol{m}_{\pi(b),:n-1}$
  for $i = b$ to 1 **do**
    if $i = 1$ **then**
      $\overline{\lambda}_i \leftarrow 1$
    else
      $\tilde{P} \leftarrow \sum_{j=1}^{i-1} \mathcal{N}(\boldsymbol{\mu}_j, \sigma^2\boldsymbol{I}) \cdot \frac{1}{i-1}$
      $\tilde{Q} \leftarrow \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\boldsymbol{I})$
      if r = `REMOVE` **then**
        $\tilde{R} \leftarrow \sum_{k=i}^{b} \mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I}_{n-1}) \cdot \frac{1}{b}$
      else
        $\tilde{R} \leftarrow \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}_{n-1})$
      end if
      $L_{\tilde{P},\tilde{Q},\tilde{R}} \leftarrow \log\left(\frac{\mathrm{d}\tilde{P}}{\mathrm{d}\tilde{Q}}(\boldsymbol{x})\right)$ with $\boldsymbol{x} \sim \tilde{R}$
      $\beta_i \leftarrow \beta \,/\, (b-1)$ # Per-bound significance
      $\tau_i \leftarrow$ `whp_lower`$(\mathrm{Law}(L_{\tilde{P},\tilde{Q},\tilde{R}}), \beta_i)$
      $\overline{\lambda}_i \leftarrow$ `sigmoid`$(-\log(i-1) - \tau_i)$
    end if
    $p_i \leftarrow \lambda_i \cdot \prod_{j=i+1}^{b}(1-\lambda_j)$
  end for
  if r = `REMOVE` **then**
    return $(\sum_{j=1}^{b} \mathcal{N}(\mu_j, \sigma) \cdot p_j), \mathcal{N}(0, \sigma)$
  else
    return $\mathcal{N}(0, \sigma), (\sum_{j=1}^{b} \mathcal{N}(\mu_j, \sigma) \cdot p_j)$
  end if

Variational tail bounds. A finding that may be of independent interest is that our privacy loss tail bounds from Theorems 5.5 and E.7 are an instance of a more general family of variational bounds

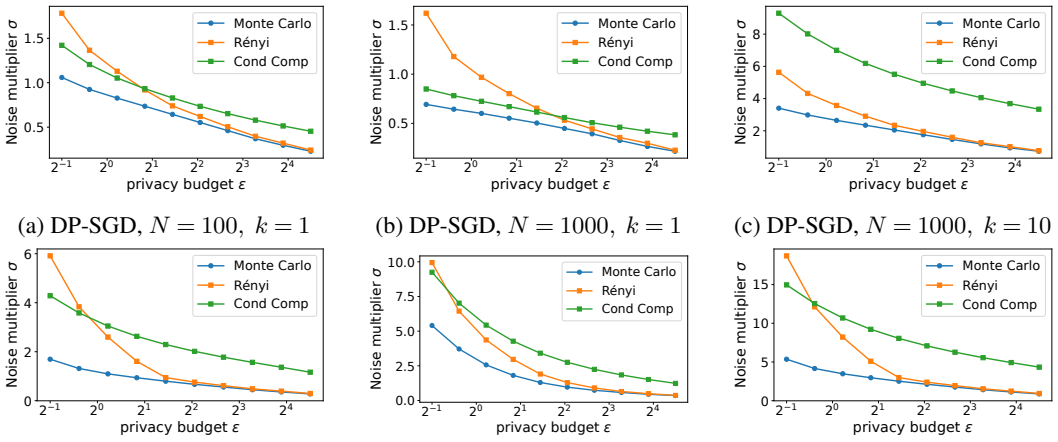
(see proofs in Appendix F). That is, every categorical distribution ψ supported on $\{1, \ldots, i\}$ yields a different bound on the CDF of ternary Gaussian mixture privacy loss random variable $L_{\tilde{P},\tilde{Q},\tilde{R}}$ where R has i components. As demonstrated in Appendix G, choosing the best bound from a variational family Ψ of distributions can enable tighter privacy accounting.

6 Experiments

In this section, we evaluate the Rényi and conditional composition (Cond Comp) based accountants against the sampling based Monte Carlo accountant [12]. We emphasize that privacy guarantees derived via a Monte Carlo accountant are qualitatively different from the (ε, δ)-DP guarantees of our proposed accountants: they hold only with high probability, or require the mechanism to abstain (see Section 2.2). In favor of the baseline, we show raw Monte Carlo estimates (for probabilistic upper bounds, see Appendix A.2). We perform privacy accounting across different modern matrix mechanisms, including DP-SGD ($\boldsymbol{C} = \boldsymbol{I}$) and BSR [23] in Figure 2, as well as BISR, BandInvMF [24], BLT [17] and BandMF [30] in Figure 3 of Appendix A. While DP-SGD, BSR, and BISR are defined analytically, BandMF, BandInvMF, and BLT define the entries of $\boldsymbol{C}$ through an optimization problem (see details in Appendix B). With the Rényi accountant, we set bandwidth truncation $p = 4$, which is exact for DP-SGD and BSR(4). For $\varepsilon \leq 2$, we additionally evaluate $p = 2$ to accommodate larger R'enyi orders (up to $\alpha = 25$) and thereby tighten our bounds at comparable computational cost. For consistency with the experiments in [12], we set $\delta = 10^{-5}$ and plot the required noise multiplier σ for different $\varepsilon > 0$. This matches the practical use of DP machine learning in which we first define a desired level (ε, δ) of privacy and then calibrate the mechanism to attain it. An implementation will be made available at github.com/morganstanley/MSML/tree/main/papers/Matrix_Mechanism_Accounting.

Across all experiments, our Rényi accountant closely matches the Monte Carlo reference in the low-privacy regime (roughly $\varepsilon \geq 2$–4), consistent with its strength in capturing large deviations. In the high-privacy regime ($\varepsilon \leq 1$), our conditional composition accountant yields tighter bounds and enables smaller calibrated noise multipliers, especially under stronger subsampling (larger b, smaller batch size). In multi-epoch settings, the same qualitative pattern holds for non-trivial matrix mechanisms (BSR/BISR), whereas for multi-epoch DP-SGD the conditional composition bound can degrade and provides little benefit over Rényi accounting (see Appendix G.5 for a detailed explanation). Finally, varying correlation bandwidth (e.g., $p = 4$ vs. $p = 64$) does not change these conclusions: the observed low- vs. high-privacy behavior is robust to the strength of noise correlation.

Model utility. The previously cited works already evaluate model utility for different factorizations across various noise multipliers σ. Nevertheless, the utility with our specific σ may be of interest. We



(a) DP-SGD, $N = 100,\ k = 1$ (b) DP-SGD, $N = 1000,\ k = 1$ (c) DP-SGD, $N = 1000,\ k = 10$

(d) BSR, $N = 100,\ k = 1,\ p = 4$ (e) BSR, $N = 100,\ k = 1,\ p = 64$ (f) BSR, $N = 1000,\ k = 10,\ p = 4$

Figure 2: Comparison of the Rényi based and conditional composition based accountants with the Monte Carlo (MC) accountant for DP-SGD and Banded Square Root (BSR) at $\delta = 10^{-5}$, with calibrated noise multipliers σ shown as a function of the privacy budget ε. Each plot corresponds to a different choice of the number of iterations n, bandwidth p and the number of epochs k.

thus conduct experiments on CIFAR-10 in Appendix A.3. In particular, we find that our guarantees are strong enough to capture the utility benefit of using correlated instead of uncorrelated noise.

DP-SGD specific bounds. Our matrix mechanism accountant can also be applied to the special case of uncorrelated noise (DP-SGD). For completeness, we compare to existing accountants for DP-SGD under random allocation in Appendix A.6. The method from Feldman & Shenfeld [20] is slightly tighter, but qualitatively similar. As we explain in more detail in Appendix A.7, our Rényi accountant is exact and our conditional composition accountant can be understood as a strict generalization of their "Poisson decomposition". The latter just uses more general bounds that are slightly less tight when restricted to the special case of DP-SGD. In contrast, the PLD method from concurrent work of Feldman & Shenfeld [21] matches the Monte Carlo accountant almost exactly. This makes its generalization to correlated noise mechanisms a promising but non-trivial direction for future work.

Empirical runtime. One motivation for our methods was efficiency, i.e., overcoming the $\mathcal{O}(\delta^{-1})$ dependence of Monte Carlo sampling. In Appendices A.4 and A.9, we complement our asymptotic analyses by measuring runtime across parameter ranges. The results confirm that different accountants are preferable in different regimes: Monte Carlo accounting is ill-suited for small δ, and, for example, takes longer than our accountants for BSR with bandwidth 4, $N = 1000$ steps and $b = 100$ batches per epoch when $\delta < 10^{-5}$. The conditional composition runtime is quadratic in N and thus takes longer than Monte Carlo accounting for $\delta = 10^{-7}$ when $N \geq 10^4$ on first execution. However, subsequent runs for different σ are significantly cheaper and take less time than Monte Carlo accounting with $\delta = 10^{-6}$ even at $N = 10^4$. The Rényi accountant exhibits (potentially optimal) exponential runtime in bandwidth p, but can trade it off for tightness via truncation. Finally, it is important to remember that, regardless of whether the Monte Carlo accountant is faster or slower in a given setting, its privacy guarantees are *always* qualitatively weaker than our deterministic bounds.

7 Conclusion

The only available accountant for matrix mechanisms under balls-in-bins sampling relies on Monte Carlo sampling [12], which yields guarantees that hold only with high probability or require random abstention. In this work, we provide a way to perform accounting deterministically, that is, without relying on sampling. Our Rényi accountant computes the guarantees efficiently via dynamic programming, improving the time complexity for DP-SGD under random allocation [20] from exponential to polynomial, and generalizing the analysis to subsampled matrix mechanisms. The resulting accountant offers tight privacy guarantees in the low privacy regime. Our conditional composition accountant upper-bounds the privacy profile via a sequence of independent Gaussian mixture mechanisms, which lets us apply tight accountants and offers better guarantees in the high privacy regime.

Broader impact. Our work is primarily targeted at mitigating negative societal impact by contributing towards improving model training with formal privacy guarantees. However, there are risks in misinterpretation or misrepresentation of these guarantees, which we discuss in in Appendix J.

Limitations and future work. There are inherent limitations to the Rényi accountant, as its complexity in the bandwidth is exponential, leading to a trade-off between runtime and tightness. With a limited compute budget, one either has to truncate the bandwidth or the range of Rényi orders, both of which will result in pessimistic bounds unless strategy $\boldsymbol{C}$ is inherently banded and privacy budget ε is large. Beyond that, the conversion from Rényi to hockey-stick divergences is inherently lossy. With a more efficient implementation, the accountant could be improved; however, the exponential dependence on bandwidth is provably unavoidable. An limitation of conditional composition is that it uses a coarse partition into "good" and "bad" outcomes, which may result in overly pessimistic dominating pairs. Future work should explore more fine-grained partitions. However, note that the limit case of infinitely fine-grained partitions is just standard privacy loss distribution accounting, which appears challenging for reasons outlined in Section 3 and Appendix I. A limitation of our instantiation of conditional composition is that it relies on inherently conservative analytical bounds on posterior subsampling probabilities.. An avenue for improvement is to expand upon the variational bounds from Appendix F through computational variational inference. Alternatively, one could use a tighter privacy accountant to improve the tail bounds on the ternary privacy loss underlying the posterior. Overall, we expect future work to develop even tighter sampling-free accountants to fully leverage the benefits of combining correlated subsampling and correlated noise for more practical differentially private machine learning with strong privacy–utility trade-offs.

Acknowledgments

We thank Arun Ganesh for insightful discussions and for sharing the code for the Monte Carlo accountant. We are also grateful to Moshe Shenfeld for helping in evaluating their PLD accountant and providing feedback on our manuscript. Nikita Kalinin's research was funded in part by the Austrian Science Fund (FWF) [10.55776/COE12].

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Conference on Computer and Communications Security (CCS)*, 2016.

[2] Alghamdi, W., Gomez, J. F., Asoodeh, S., Calmon, F., Kosut, O., and Sankar, L. The saddle-point method in differential privacy. In *International Conference on Machine Learning (ICML)*, 2023.

[3] Balle, B. and Wang, Y.-X. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018.

[4] Balle, B., Berrada, L., Charles, Z., Choquette-Choo, C. A., De, S., Doroshenko, V., Dvijotham, D., Galen, A., Ganesh, A., Ghalebikesabi, S., Hayes, J., Kairouz, P., McKenna, R., McMahan, B., Pappu, A., Ponomareva, N., Pravilov, M., Rush, K., Smith, S. L., and Stanforth, R. JAX-Privacy: Algorithms for privacy-preserving machine learning in JAX, 2025. URL `http://github.com/google-deepmind/jax_privacy`.

[5] Barthe, G. and Olmedo, F. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In *International Conference on Automata, Languages, and Programming*, 2013.

[6] Beltran, S. R., Tobaben, M., Jälkö, J., Loppi, N., and Honkela, A. Efficient and scalable implementation of differentially private deep learning without shortcuts. In *Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2026.

[7] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[8] Canonne, C. L., Kamath, G., and Steinke, T. The discrete Gaussian for differential privacy. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[9] Choquette-Choo, C. A., Ganesh, A., McKenna, R., McMahan, H. B., Rush, J. K., Thakurta, A. G., and Xu, Z. (amplified) banded matrix factorization: A unified approach to private training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[10] Choquette-Choo, C. A., McMahan, H. B., Rush, J. K., and Thakurta, A. G. Multi epoch matrix factorization mechanisms for private machine learning. In *International Conference on Machine Learning (ICML)*, 2023.

[11] Choquette-Choo, C. A., Ganesh, A., Steinke, T., and Thakurta, A. Privacy amplification for matrix mechanisms. In *International Conference on Learning Representations (ICLR)*, 2024.

[12] Choquette-Choo, C. A., Ganesh, A., Haque, S., Steinke, T., and Thakurta, A. Near exact privacy amplification for matrix mechanisms. In *International Conference on Learning Representations (ICLR)*, 2025.

[13] Chua, L., Ghazi, B., Harrison, C., Leeman, E., Kamath, P., Kumar, R., Manurangsi, P., Sinha, A., and Zhang, C. Balls-and-bins sampling for DP-SGD. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

[14] Denisov, S., McMahan, H. B., Rush, J., Smith, A., and Guha Thakurta, A. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[15] Dong, A., Chen, W.-N., and Ozgur, A. Leveraging randomness in model and data partitioning for privacy amplification. In *International Conference on Machine Learning (ICML)*, 2025.

[16] Doroshenko, V., Ghazi, B., Kamath, P., Kumar, R., and Manurangsi, P. Connect the dots: Tighter discrete approximations of privacy loss distributions. *Privacy Enhancing Technologies Symposium (PETS)*, 2022.

[17] Dvijotham, K. D., McMahan, H. B., Pillutla, K., Steinke, T., and Thakurta, A. Efficient and near-optimal noise generation for streaming differential privacy. In *Symposium on Foundations of Computer Science (FOCS)*, 2024.

[18] Dwork, C. Differential privacy. In *International colloquium on automata, languages, and programming*, 2006.

[19] Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: from local to central differential privacy via anonymity. In *Symposium on Discrete Algorithms*, 2019.

[20] Feldman, V. and Shenfeld, M. Privacy amplification by random allocation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[21] Feldman, V. and Shenfeld, M. Efficient privacy loss accounting for subsampling and random allocation. In *International Conference on Machine Learning (ICML)*, 2026.

[22] Google DP Team. Privacy loss distributions. `https://raw.githubusercontent.com/google/differential-privacy/main/common_docs/Privacy_Loss_Distributions.pdf`, 2025. Accessed January 29, 2026.

[23] Kalinin, N. P. and Lampert, C. Banded square root matrix factorization for differentially private model training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[24] Kalinin, N. P., McKenna, R., Upadhyay, J., and Lampert, C. H. Back to square roots: An optimal bound on the matrix factorization error for multi-epoch differentially private SGD. In *International Conference on Learning Representations (ICLR)*, 2026.

[25] Koskela, A., Jälkö, J., and Honkela, A. Computing tight differential privacy guarantees using FFT. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[26] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.

[27] Lebeda, C. J., Regehr, M., Kamath, G., and Steinke, T. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition. In *Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

[28] Li, C., Miklau, G., Hay, M., McGregor, A., and Rastogi, V. The matrix mechanism: Optimizing linear counting queries under Differential Privacy. *International Conference on Very Large Data Bases (VLDB)*, 2015.

[29] Li, N., Qardaji, W., and Su, D. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Symposium on Information, Computer and Communications Security*, 2012.

[30] McKenna, R. Scaling up the banded matrix factorization mechanism for differentially private ML. In *International Conference on Learning Representations (ICLR)*, 2025.

[31] McMahan, H. B., Xu, Z., and Zhang, Y. A hassle-free algorithm for strong differential privacy in federated learning systems. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

[32] Mironov, I. Rényi differential privacy. In *Computer Security Foundations Symposium (CSF)*, 2017.

[33] Namatevs, I., Sudars, K., Nikulins, A., and Ozols, K. Privacy auditing in differential private machine learning: The current trends. *Applied Sciences*, 15(2):647, 2025.

[34] Pillutla, K., Upadhyay, J., Choquette-Choo, C. A., Dvijotham, K., Ganesh, A., Henzinger, M., Katz, J., McKenna, R., McMahan, H. B., Rush, K., et al. Correlated noise mechanisms for differentially private learning, 2025. arXiv preprint arXiv:2506.08201.

[35] Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. G. How to DP-fy ML: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 2023.

[36] Sommer, D., Meiser, S., and Mohammadi, E. Privacy loss classes: The central limit theorem in differential privacy. In *Privacy Enhancing Technologies Symposium (PETS)*, 2019.

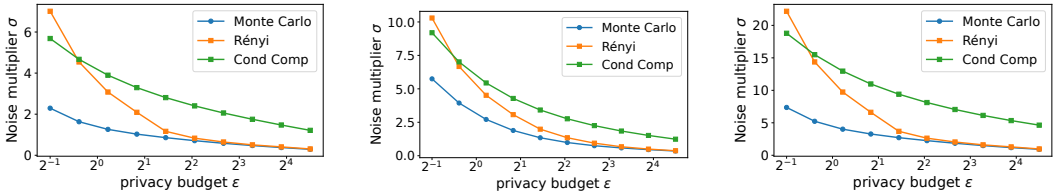
[37] Wang, J. T., Mahloujifar, S., Wu, T., Jia, R., and Mittal, P. A randomized approach to tight privacy accounting. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[38] Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. Subsampled rényi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

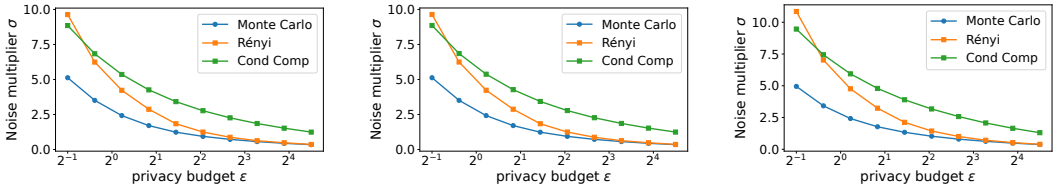
[39] Zhu, Y., Dong, J., and Wang, Y.-X. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

A Additional Experimental Results

A.1 Additional matrix mechanisms



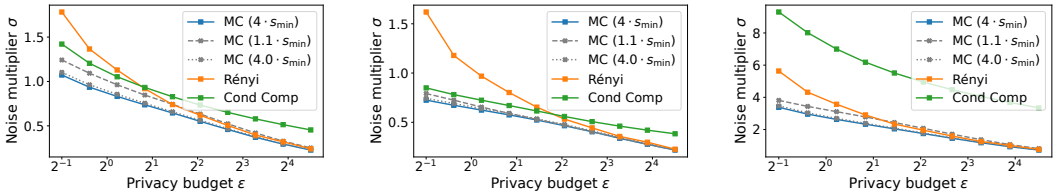
(a) BISR, $N = 100,\ k = 1,\ p = 4$ (b) BISR, $N = 100,\ k = 1,\ p = 64$ (c) BISR, $N = 1000,\ k = 10,\ p = 4$



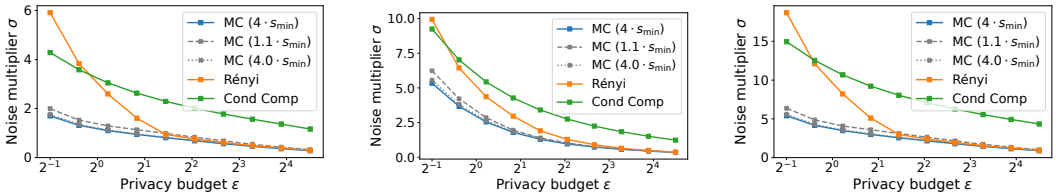
(d) BLT, $N = 100,\ k = 1$ (e) BandMF, $N = 100,\ k = 1,\ p = 64$ (f) BandInvMF, $N = 100,\ k = 1,\ p = 4$

Figure 3: Comparison of the RDP based and conditional composition based accountants with the MC accountant for BISR, BandInvMF, BLT and BandMF at $\delta = 10^{-5}$, with calibrated noise multipliers σ shown as a function of the privacy budget ε. Each plot corresponds to a different choice of the number of iterations n, the number of epochs k, and parameter p.

A.2 Monte Carlo high-probability bounds.



(a) DP-SGD, $N = 100,\ k = 1$ (b) DP-SGD, $N = 1000,\ k = 1$ (c) DP-SGD, $N = 1000,\ k = 10$



(d) BSR, $N = 100,\ k = 1,\ p = 4$ (e) BSR, $N = 100,\ k = 1,\ p = 64$ (f) BSR, $N = 1000,\ k = 10,\ p = 4$

Figure 4: Main experimental Figure 2, re-drawn to evaluate both the raw Monte Carlo estimate (blue) in addition to high probability upper bounds (gray) for varying number of samples.

In our main experiments, we directly used the Monte Carlo estimates of the privacy profile for calibration. In practice, we instead have to compute probabilistic upper bounds, which we show in Figure 4. The number of samples is set as a multiple of the minimum number of samples $s_{\min}$ needed to obtain a bound that holds with high probability $1 - p$. Here, we choose $p = \delta = 10^{-5}$, which is needed to construct an abstention-based mechanism (see Theorem 2.1 in [12]). As can be seen, the high probability bound is almost identical to the raw sample estimate when using enough samples. In other words, the limitation is not the tightness of the tail bounds, but (1) that the resulting guarantees are qualitatively weaker than deterministic, sampling-free privacy parameters and (2) the cost of sampling (see runtime experiments in Figure 6a).

A.3 Model utility on CIFAR-10

In Figure 5 we compare the model utility attained via our noise multipliers to those attained using the Monte Carlo accounting noise multipliers. We replicate the CIFAR-10 experimental setup from [23]. We use 5 random seeds and show mean and standard deviation of test accuracy. Training is performed for $N = 970$ steps divided into $k = 10$ epochs, which corresponds to a batch size of 512. Gradients are clipped to an ℓ_2-norm of 8 and we calibrate noise multipliers to $\delta = 10^{-5}$ (for further details on the experimental setup, see Appendix B.

As can be seen, there is a good match in utility for $\varepsilon \geq 4$, which is consistent with our observations about the computed noise multipliers σ. Furthermore, our sampling-free guarantees are strong enough to retain the benefit of using correlated (BSR, BISR) instead of uncorrelated noise (DP-SGD).

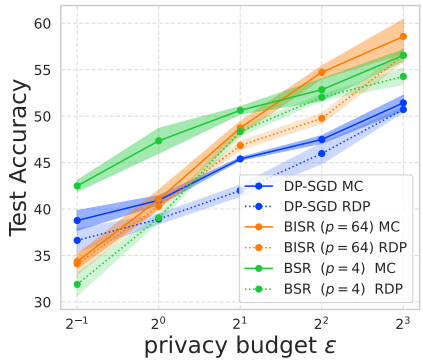
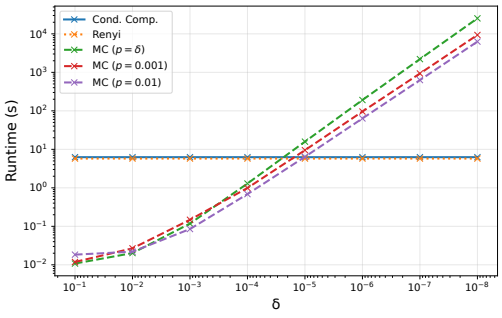
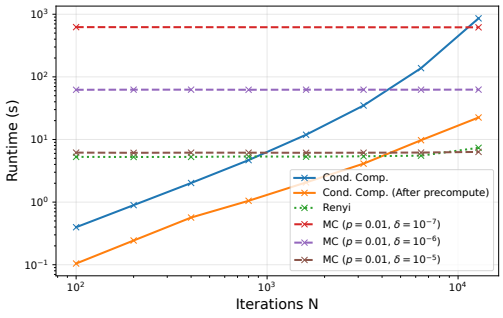


Figure 5: ConvNet model utility on CIFAR-10

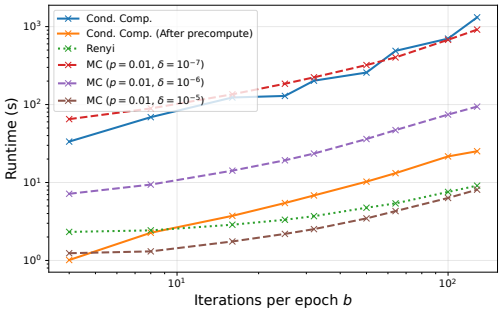
A.4 Empirical runtime



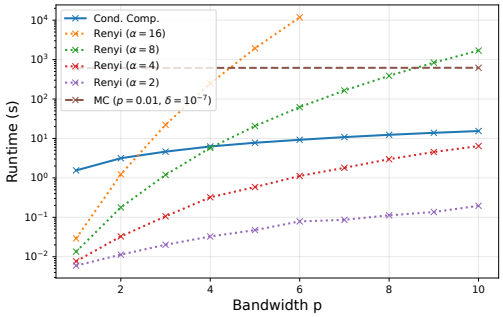
(a) Varying privacy budget δ



(b) Varying number of iterations N



(c) Varying batches per epoch b



(d) Varying bandwidth p

Figure 6: Wall-clock runtime under varying matrix mechanism and privacy accounting parameters. All observations are consistent with the claimed asymptotic runtimes.

In the main text, we already analyzed the asymptotic complexity of our proposed methods, as well the Monte Carlo baseline. Here, we complement this analysis by measuring wall-clock runtime while varying all involved parameters, namely privacy budget δ, number of iterations N, batches per epoch b and bandwidth p. All experiments were performed on the same Intel Xeon Gold 6148 CPU (2.4 GHz, 768 GHz RAM). We measure the cost for a single run of each algorithm (or: run for

all training steps in the case of conditional composition), i.e., we do not perform repeated runs to calibrate noise multiplier σ.

Default parameters. While varying the previously mentioned parameters, we use the following default values for our accountants, which apply unless explicitly stated otherwise. By default, we consider privacy parameters $\varepsilon = 8, \delta = 10^{-5}$ and noise multiplier $\sigma = 10$ (note that none of the accountants have an asymptotic runtime dependence on these parameters). For the Rényi accountant, we use $\alpha = 8$ by default and do not employ bandwidth truncation. For the conditional composition accountant, we use the default AM-GM ternary privacy loss bound, i.e., we do not use additional variational distributions that would increase computational cost by a constant factor. We further set "bad event" probability $\delta = \delta_E \,/\, 2$. With Monte Carlo accounting, we use the minimum number of samples $s_{\min}$ to attain a moderate significance / failure probability of $p = 0.01$ while being able to verify $\delta < 1$.

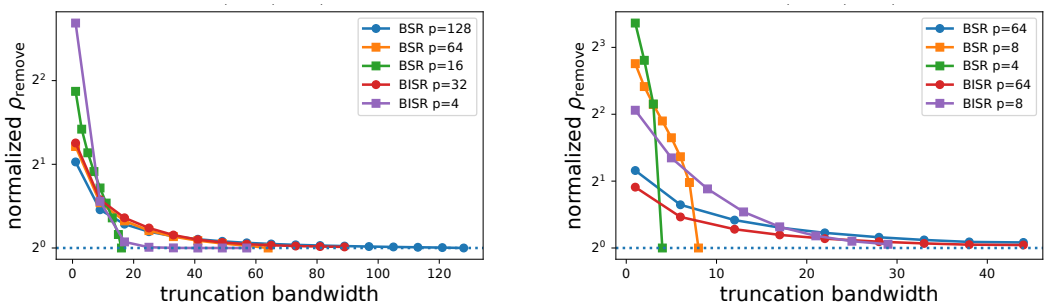
Varying privacy budget. In Figure 6a, we vary the privacy budget δ for BSR factorization ($N = 1000$ steps, $b = 100$ batches per epoch, bandwidth $p = 4$) using the minimum required number of samples for the Monte Carlo baseline's guarantees to hold with significance p. Except for weak guarantees $\delta \geq 10^3$ which require less than 20000 samples, the runtime of the Monte Carlo method increases inversely proportional to δ. This confirms the importance of developing sampling-free privacy accounting methods for matrix mechanisms, especially when considering strict privacy budgets or large datasets, where one typically uses $\delta < \frac{1}{|D|}$.

Varying number of iterations. In Figure 6b, we vary the number of iterations N for BSR factorization ($b = 100$ batches per epoch, bandwidth $p = 4$). For large $N > 10000$, the cost of evaluating the conditional composition accountant $\mathcal{O}(N^2b^2)$ from scratch is similar to Monte Carlo accounting with $10^{-7} \leq \delta \leq 10^{-8}$. However, subsequent evaluations (orange) for different noise multipliers σ are significantly cheaper, which allows for amortizing the runtime when calibrating σ.

Varying batches per epoch. In Figure 6c, we vary the other parameter that conditional composition has a quadratic dependence on, i.e., batches per epoch b here shwon for BSR factorization ($N = 12800$ iterations, bandwidth $p = 4$). As before, for this large $N = 12800$ the cost of evaluating the conditional composition accountant $\mathcal{O}(N^2b^2)$ from scratch is similar to Monte Carlo accounting with $10^{-7} \leq \delta \leq 10^{-8}$. However, after an initial evaluation, subsequent evaluations are less expensive than Monte Carlo accounting with $\delta = 10^{-6}$.

Varying bandwidth. Finally, in Figure 6d we vary bandwidth p and Rényi order α for BSR factorization ($N = 1000$ iterations, 100 batches per epoch $p = 4$). For Rényi accounting, we truncate the bandwidth of the Gram matrix to be identical to the bandwidth of the BSR noise correlation matrix. As anticipated, at a fixed compute budget we need to either reduce the bandwidth to evaluate the Rényi accountant for large α, or limit α, which will result in weaker guarantees in high privacy / small ε settings.

A.4.1 Effect of bandwidth truncation on Rényi bounds



(a) $N = 1000, k = 1, \alpha = 2, \sigma = 0.5$ (b) $N = 1000, k = 10, \alpha = 3, \sigma = 1$

Figure 7: Comparison of the "remove" direction Rényi guarantee in the single-epoch ($k = 1$) and multi-epoch ($k = 10$) setting, after truncation relative to the true guarantee without truncation. When the matrix is banded with a small bandwidth, it is better not to truncate it at all, whereas for a large bandwidth, one obtains an accurate approximation with a relatively small p.

Effect on Rényi divergence. In Figure 1, we visually demonstrated that matrix mechanisms, including those with high bandwidth, often have small effective bandwidth in the Gram matrix. Our dynamic program exploits this by truncating the Gram matrix to speed up computation at the cost of an easy-to-compute additive term. In Figure 7, we compare the computed Rényi divergence across different factorizations, bandwidths, and Gram matrix bandwidth truncations. We specifically focus on small values of α, which correspond to the low-privacy / large-ε regime in which Rényi accounting performs best. In both the single- and multi-epoch setting, we observe the following: For mechanisms with small bandwidth, truncating too aggressively may introduce a large error. For mechanisms with large bandwidth (e.g., BSR with $p = 64$ or BISR with arbitrary p), even truncating to a small fraction can still offer a good upper bound due to their limited effective bandwidth.

Effect on correction term. Recall from our discussion from Equation (4) that truncation itself reduces Rényi divergence, which has to be compensated by an additive correction term $\frac{\alpha\tau}{2\sigma^2}$. For completeness, we evaluate the dependence of τ on truncation bandwidth p for different mechanisms and $N = 100$ steps in Table 1.

Table 1: Truncation correction term τ of the Rényi accountant for $N = 100$, $k = 1$ and varying truncation bandwidth p. Note that $\tau = 0$ for $p > N / 2$.

Mechanism	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$	$p = 50$
BSR(4)	0.805	0.531	0	0	0	0	0
BSR(8)	1.060	0.824	0.547	0	0	0	0
BSR(64)	1.750	1.535	1.312	1.081	0.837	0.559	0.320
BISR(4)	0.981	0.755	0.456	0.159	0.019	$2.87 \cdot 10^{-4}$	$2.51 \cdot 10^{-6}$
BLT	1.654	1.440	1.216	0.982	0.740	0.465	0.274
BandMF	1.657	1.441	1.214	0.976	0.717	0.426	0.207

Effect on noise multiplier. The figure in Figure 1 shows the effect of truncation on our Rényi divergence bounds, but these are only an intermediate step. In Table 2, we additionally show the final noise multipliers σ after calibrating for a specific privacy budget (ε, δ). For a given ε, we choose $\alpha_{\max}$ large enough so that the only looseness is due to truncation p and conversion to (ε, δ)-DP. Table 2a shows different mechanisms for small $\varepsilon \in \{2, 3, 4\}$ and $p \in \{1, 2, 4\}$ under $\alpha_{\max} = 16$ while Table 2b shows mechanisms for larger $\varepsilon \in \{4, 8, 16\}$ and $p \in \{2, 4, 8\}$ under $\alpha_{\max} = 8$.

Table 2: Calibrated noise multipliers σ for $\delta = 10^{-5}$ attained by Rényi accounting under varying truncation bandwidth p. The maximal Rényi order $\alpha_{\max}$ is chosen large enough that non-tightness is only due to truncation and conversion to (ε, δ)-DP.

(a) Small ε and p under $\alpha_{\max} = 8$.

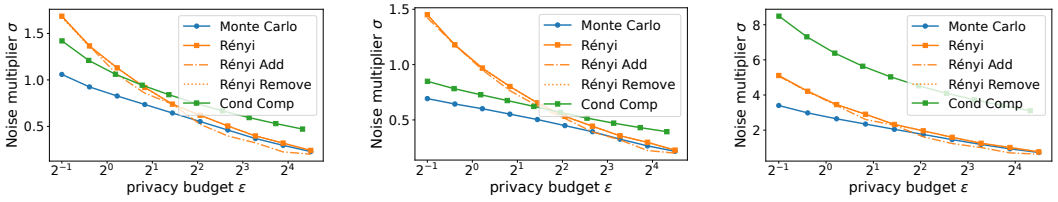
Mechanism	p	$\varepsilon = 2$	$\varepsilon = 3$	$\varepsilon = 4$
BSR(4)	1	1.937	1.349	1.045
BSR(4)	2	1.592	1.112	0.866
BSR(4)	4	1.073	0.878	0.768
BSR(64)	1	2.847	1.982	1.534
BSR(64)	2	2.672	1.860	1.439
BSR(64)	4	2.480	1.727	1.337
BLT	1	2.769	1.928	1.491
BLT	2	2.588	1.802	1.395
BLT	4	2.390	1.665	1.289

(b) Larger ε and p under $\alpha_{\max} = 8$.

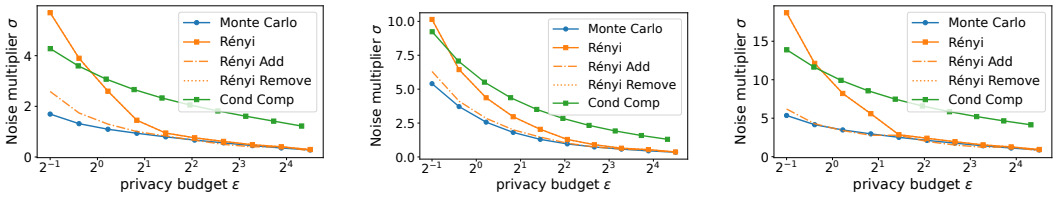
Mechanism	p	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
BSR(8)	2	1.063	0.609	0.404
BSR(8)	4	0.903	0.587	0.404
BSR(8)	8	0.829	0.577	0.404
BSR(64)	2	1.439	0.795	0.474
BSR(64)	4	1.337	0.743	0.471
BSR(64)	8	1.230	0.712	0.471
BLT	2	1.395	0.770	0.463
BLT	4	1.289	0.719	0.461
BLT	8	1.177	0.693	0.461

A.4.2 Comparison of Rényi "remove" and "add" direction.

As describe Algorithm 1, our overall Rényi accountant uses a dynamic program for the "remove" direction and an analytical bound for the "add" direction. If the "add" bound were consistently larger, then the "remove" bound would never be active and there would be no benefit to running the dynamic program. However, as shown in Figure 8, this is not the case. In fact, we consistently found the "add"



(a) DP-SGD, $N = 100,\ k = 1$ (b) DP-SGD, $N = 1000,\ k = 1$ (c) DP-SGD, $N = 1000,\ k = 10$



(d) BSR, $N = 100,\ k = 1,\ p = 4$ (e) BSR, $N = 100,\ k = 1,\ p = 64$ (f) BSR, $N = 1000,\ k = 10,\ p = 4$

Figure 8: Main experimental Figure 2, re-drawn with separate "add" and "remove" directions for the Rényi accountant. The "add" bound is consistently smaller, i.e., the overall privacy guarantee is identical to the "remove" direction and the "add" slack has no effect on the overall privacy guarantee.

direction bound to be smaller across parameters and matrix factorizations. This matches up with prior work on Rényi accounting. For example, Feldman & Shenfeld [20] claim for the special case of uncorrelated noise (DP-SGD) that "numerical analysis seems to indicate the bound on the remove direction always dominates the one for the add direction". Mironov [32] formally prove for Poisson subsampling that the "remove" direction always dominates.

A.5 Effect of bad event probability on conditional composition bounds

In all other experiments, we choose bad event probability $\delta_E = \frac{\delta}{2}$ for consistency with [11]. However, one can in principle use any $\delta_E \in (0, \delta)$. In Table 3, we compare our previous results to the the calibrated noise multiplier σ for an optimized $\delta \in \{0.9, 0.5, \text{logspace}(-1, -3, 5\}$ for DP-SGD and BISR(64). We find that the grid search over δ_E can offer slightly better σ, but that $\delta\ /\ 2$ from prior work is a reasonable default.

Table 3: Comparison of calibrated noise multiplier σ for conditional composition accountant with "bad" event probability $\delta_E = \delta\ /\ 2$ compared to a grid search over $\delta_E \in \{0.9, 0.5, \text{logspace}(-1, -3, 5\}$. Optimization can offer better noise multipliers.

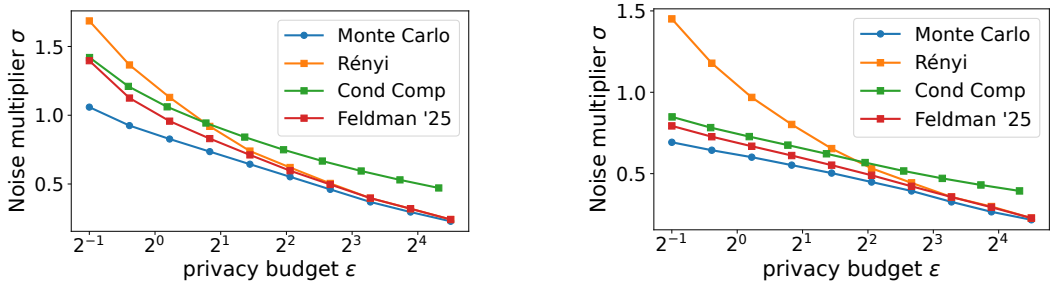
Mechanism	Optimized	$\varepsilon = 0.50$	$\varepsilon = 0.75$	$\varepsilon = 1.14$	$\varepsilon = 1.71$	$\varepsilon = 2.58$	$\varepsilon = 3.88$
DP-SGD	N	1.420	1.210	1.061	0.943	0.841	0.749
DP-SGD	Y	1.396	1.192	1.047	0.931	0.831	0.741
BISR(64)	N	9.179	7.065	5.523	4.379	3.516	2.856
BISR(64)	Y	9.047	6.995	5.490	4.367	3.516	2.856

A.6 Comparison to DP-SGD-specific random allocation bounds

The focus of our work is on privacy accounting for correlated noise, but are accountants are also applicable to the special case of uncorrelated noise (DP-SGD). For completeness, we thus also compare to Feldman & Shenfeld [20, 21].

A.7 Comparison to Feldman & Shenfeld (2025) method for uncorrelated noise

Similar to us, Feldman & Shenfeld [20] combine an exact Rényi accountant with other bounds that are tighter for small ε. However, as shown in Figure 9, their procedure offers tight bounds for small to moderate privacy budgets ε. This raises the question of what prevents our accountants from achieving similar tightness.



(a) DP-SGD, $N = 100,\ k = 1$ (b) DP-SGD, $N = 1000,\ k = 1$

Figure 9: Main experimental Figures 2a and 2b, re-drawn to also include the DP-SGD specific guarantees of [20] Their bounds are qualitatively similar (slack for small ε, tight for large ε), but tighter because they combine a wider range of methods specific to uncorrelated noise.

By definition, our exact Rényi accountant yields the same results as the "direct analysis" accountant from their Theorems 4.8 and 4.9 when applied to DP-SGD. However, our novel dynamic program does so in a computationally much more efficient manner (see Appendix A.9 for empirical runtimes).

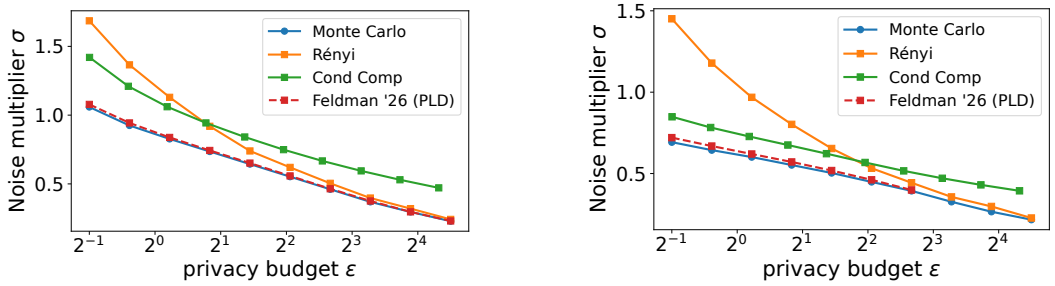
Similarly, our conditional composition accountant from Algorithm 3 could be made identical to their "truncated Poisson" accountant for DP-SGD (their Theorem 4.1) by letting $\lambda_1 = \cdots = \lambda_{b-1}$ and using their maximal Azuma Hoeffding bound to bound λ_b. However, as can be seen from their Appendix B.4, this bound relies on the same decomposition assumption as [21]. As we already discussed in Section 3 and Appendix I, this decomposition into per-component PLRVs does not generalize to matrix mechanisms.

Furthermore, the same PLRV decomposition lets them recursively refine their bound (their Theorem 4.2), which again would not generalize to matrix mechanisms.

Finally, they are able to apply a complete separate class of bounds ("Poisson decomposition"), which exploits that they only ever need to distinguish two cases per step, just like in Poisson subsampling: Either the private record participates or not. In contrast, our general method has to account for the fact that earlier participation may also have an indirect effect through correlation.

To summarize, since we address a more general problem, we must use more general bounds that are less tight when restricted to the special case of DP-SGD.

A.8 Comparison to Feldman & Shenfeld (2026) method for uncorrelated noise



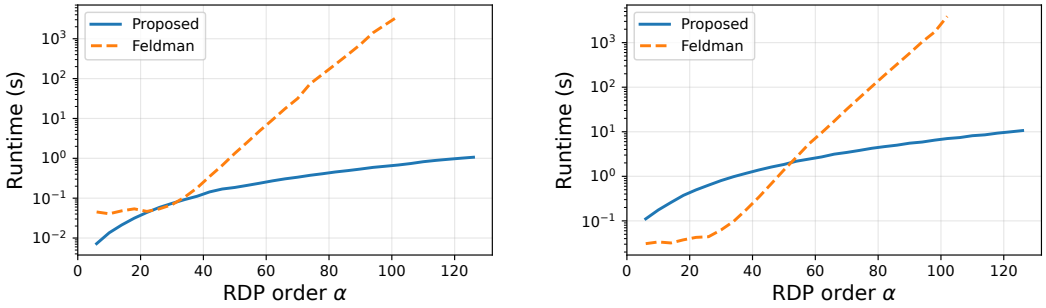
(a) DP-SGD, $N = 100,\ k = 1$ (b) DP-SGD, $N = 1000,\ k = 1$

Figure 10: Main experimental Figures 2a and 2b, re-drawn to also include the DP-SGD specific guarantees of Feldman & Shenfeld [21]. Their bounds more almost exactly the Monte Carlo accountant.

In concurrent work, Feldman & Shenfeld [21] improve upon their previous paper through a method that foregoes bounding privacy profile $\delta(\varepsilon)$ or Rényi profile $\rho(\alpha)$ and instead operates in privacy loss distribution space. As shown in Figure 10, this leads to improved bounds that almost exactly match those attained via Monte Carlo accounting (see Figure 10).

We believe that future work should work towards privacy loss distribution accounting for subsampled matrix mechanisms. However, for reasons outlines in Section 3 and Appendix I, we do not believe that this can be done through a simple generalization from their DP-SGD method.

A.9 Comparison to Feldman & Shenfeld (2025) Rényi accountant runtime



(a) DP-SGD ($N = 100, b = 100$)

(b) DP-SGD ($N = 1000, b = 1000$)

Figure 11: Wall-clock runtime comparison to Feldman et al. for the special case of uncorrelated noise / DP-SGD. Except for $\alpha \leq 30$, which benefits from vectorization, hasing, and just-in-time compilation, the baseline's runtime explodes explodes exponentially ($\mathcal{O}(2^\alpha)$) while our polynomial-time dynamic program ($\mathcal{O}(b\alpha^2)$) scales to much larger α

In Section 4 we claimed an improvement in runtime for exact Rényi accounting of uncorrelated-noise random allocation (DP-SGD) in the "remove direction". Specifically, we claimed an improvement from exponential $\mathcal{O}(2^\alpha)$ to quadratic $\mathcal{O}(b\alpha^2)$ compared to the best known method from [20]. To confirm that this bound does not only hold in asymptotic limits, we evaluate both accountants on the the same Intel Xeon Gold 6148 CPU (2.4 GHz, 768 GHz RAM). The empirical runtime measurements in Figure 11 confirm that the baseline's cost explodes exponentially, whereas our method scales to large Rényi orders α.

B Experimental details

B.1 Conditional composition accountant

Parameters for Algorithm 2. In principle, given a target $\delta \in [0, 1]$, we can choose an arbitrary "bad" event probability $\delta_E < \delta$ or even optimize over many different choices of $\delta_E < \delta$. For simplicity, we follow [11] and simply set $\delta_E = \delta \,/\, 2$. In addition to the default AM-GM bounds from Theorem 5.5 we evaluate the variational bound from Appendix G.3 which retains the same computational complexity. We set its temperatures to $T = \{10^{-1}, 10^{-0.5}, 1, 10^{0.5}, 10^{1}\}$ based on our ablation from Figure 13.

Evaluation of conditional composition bound. After determining per-step dominating pairs $(P^{(1)}, Q^{(1)}), \ldots, (P^{(N)}, Q^{(N)})$ via Algorithm 3, we compute their N-fold composition using the Google `dp_accounting` library [22] specifically its PLD accountant applied to the `MixtureGaussianPrivacyLoss`. We quantize the privacy loss distribution using the "connect the dots" algorithm from [16], while adjusting the value discretization interval such that each quantized privacy loss pmf has a support of size ≤ 1000 and at least one has a support of size exactly 1000. We otherwise use standard parameters, in particular a log-mass truncation bound of -50 for the privacy loss mechanism and a a tail mass truncation of 1×10^{-15} for the quantized privacy loss pmfs.

B.2 Rényi accountant

In our experiments, we have numerical trade-offs arising in privacy accounting. The Rényi accountant takes the bandwidth as an input. If the matrix $\boldsymbol{C}$ is p-banded for a small p, we can compute the divergence exactly. Otherwise, when the bandwidth is large or $\boldsymbol{C}$ is not banded, we must use an *effective* bandwidth: we pass a smaller p and upper bound the contributions outside the p-cyclic band (see Algorithm 4). This creates a trade-off between computational cost and the tightness of the bound.

This situation occurs, for example, for banded-inverse factorizations such as BISR and BandInvMF [24], where $\boldsymbol{C}^{-1}$ is banded but $\boldsymbol{C}$ is not, and for Buffered Linear Toeplitz (BLT) [17], which is essentially dense in both $\boldsymbol{C}$ and $\boldsymbol{C}^{-1}$. Moreover, smaller privacy budgets ε generally require larger Rényi orders α, meaning we must use smaller effective bandwidth. For our experiments, we thus report the optimal bound that can be attained for combinations of $(p, \alpha) \in \{2\} \times \{1, \ldots, 25\} \cup \{4\} \times \{1, \ldots, 7\} \cup \{2\} \times \{1, \ldots, 4\}$. The exception are ancillary experiments in Appendix A in which we explicitly vary α across different ranges.

B.3 Monte Carlo accountant.

By default, we use 400000 samples to compute raw Monte Carlo estimates for our experiments. For our ancillary experiments in Appendix A.2, where we evaluate high probability upper bounds using the method from [12], we set the number of samples to different multiples of the minimum number of samples $s_{\min}$ required to obtain a bound better than $\delta = 1$ (see details in Appendix A.2).

B.4 DP-SGD random allocation accountants.

While the focus of our work is on correlated-noise mechanisms, we still conduct a comparison to Feldman & Shenfeld [20] and Feldman & Shenfeld [21] for completeness (see Appendix A.6) using their respective reference implementations.

Feldman & Shenfeld [20]: We use the official `random-allocation` library (v1.0) available via pip. We take the minimum over all available sub-accountants using the `allocation_delta_combined` method. For comparability with our other Rényi accounting results, we set the maximum α to 25. We otherwise use default parameters. In particular, we use a discretization of 10^{-4} and a δ-tolerance of 10^{-15}.

Feldman & Shenfeld [21]: We use the official `PLD_subsampling` library (commit 952e91a36230e81bfa7c072e74dc89bb89750b0f). We leave all parameters at their default values. In particular, we use a loss discretization of 0.1, a tail truncation of 10^{-12}, a maximum FFT grid of 10^6, set `max_grid_mult` $= -1$ and employ the geometric convolution method.

B.5 Optimized matrix factorizations

The DP-SGD, BSR, and BISR factorization have closed-form expressions. In contrast, BandMF, BLT, and BandInvMF are minimized to best approximate the non-private workload matrix. For BandMF, BLT, and BandInvMF, we use the Google `jax-privacy` implementation [4] for this optimization.

B.6 CIFAR-10 experiments

For our CIFAR-10 [26] experiments, which are a standard task for evaluating the privacy–utility trade-off of matrix mechanisms (see, e.g., [10–12]) we replicate the experimental setup from [23].

Training mechanism. We use the standard train–test split, i.e., 50000 training samples and 10000 test samples. Training is performed for $N = 970$ steps divided into $k = 10$ epochs, which corresponds to a batch size of 512. Gradients are clipped to an ℓ_2-norm of 8 and we calibrate noise multipliers to $\delta = 10^{-5}$. We evaluate three different factorizations: DP-SGD, BSR ($p = 4$), and BISR ($p = 4$).

Model architecture. We use the following ConvNet architecture:

- $2\times$ Conv2D(channels=32, kernel=(3, 3), strides=(1, 1), padding='SAME', activation='relu')
- MaxPool(kernel=(2, 2), strides=(2, 2))
- $2\times$ Conv2D(channels=64, kernel=(3, 3), strides=(1, 1), padding='SAME'), activation='relu')
- MaxPool(kernel=(2, 2), strides=(2, 2))
- $2\times$ Conv2D(channels=128, kernel=(3, 3), strides=(1, 1), padding='SAME'), activation='relu')
- $2\times$ MaxPool(kernel=(2, 2), strides=(2, 2))
- Flatten()
- Dense(outputs=10)

Other details. We use 5 random seeds and show mean and standard deviation of test accuracy.

B.7 Compute resources.

All experiments for Rényi accounting and CIFAR-10 training were performed on a basic Google collab instance. Experiments involving conditional composition were performed using an Intel Xeon Gold 6148 CPU ($2.4\,\mathrm{GHz}$, $768\,\mathrm{GHz}$ RAM). All runtime measurements for Appendix A.4 were performed using this Intel Xeon CPU. The experiments did not involve multi-worker orchestration beyond python process tools for parallelizing PLd accounting with the `dp_accounting` library (see provided implementation of the conditional composition accountant.).

B.8 Licenses of used assets

CIFAR-10 is available under MIT license. The `dp_accounting` and `jax_privacy` libraries are available under Apache-2.0 license. The `random_allocation` and `PLD_accounting` libarries are available under MIT license.

C Full details of the Rényi accounting algorithm

Algorithm 4 Rényi accountant "remove" direction

Require: Correlation matrix $\mathbf{C}$, bandwidth p, separation b, parameter $\alpha \in \mathbb{N}_+$.

$\mathbf{m}_i \leftarrow \sum_{j=0}^{k-1} |\mathbf{C}|_{:,\, i+jb}$

$\mathbf{G}_{i,j} \leftarrow \langle \mathbf{m}_i, \mathbf{m}_j \rangle \quad \forall i,j \in \{1,\dots,b\}$

$\tau \leftarrow \max_{\min(|i-j|, b-|i-j|) \geq p} \mathbf{G}_{i,j}$

$\mathbf{G}^{(p)}_{i,j} \leftarrow \max(\mathbf{G}_{i,j} - \tau, 0) \quad \forall \min(|i-j|, b-|i-j|) < p$

Loop over all possible counts for the first $p-1$ values

$\log S \leftarrow -\infty$

for all $l = (l_0, \dots, l_{p-2})$ s.t. $\sum_{i=0}^{p-2} l_i = m_l \leq \alpha$ **do**

  # Precompute the prefix dynamics value

  $\log w_l \leftarrow \sum_{i<j}^{p-2} \mathbf{G}^{(p)}_{i,j} \frac{l_i l_j}{\sigma^2} + \sum_{i=0}^{p-2} [\mathbf{G}^{(p)}_{i,i} \frac{l_i(l_i-1)}{2\sigma^2} - \log(l_i!)]$

  $\texttt{states} \leftarrow \{(m_l, l) \mapsto \log w_l\}$

  for $k = p-1$ to $b-1$ **do**

    $\texttt{new_states} \leftarrow \emptyset$

    for all $(m, r) \in \texttt{states}$ **do**

      $\log w \leftarrow \texttt{states}(m, r)$

      $L \leftarrow \text{length}(r)$

      $S_1 \leftarrow 2 \sum_{i=0}^{L-1} \mathbf{G}^{(p)}_{k,\, k-L+i} \cdot r_i$

      for $t = 0$ to $\alpha - m$ **do**

        $\Delta \leftarrow \dfrac{\mathbf{G}^{(p)}_{k,k}\, t(t-1) + S_1 t}{2\sigma^2} - \log(t!)$

        $m' \leftarrow m + t$

        if $p = 1$ **then**

          $r' = ()$

        else if $L < p-1$ **then**

          $r' \leftarrow (r_0, \dots, r_{L-1}, t)$

        else

          $r' \leftarrow (r_1, \dots, r_{L-1}, t)$

        end if

        $\log w_{\text{old}} \leftarrow \texttt{new_states}(m', r')$

        $\log w_{\text{new}} \leftarrow \text{LSE}(\log w, \log w_{\text{old}} + \Delta)$

        $\texttt{new_states}(m', r') \leftarrow \log w_{\text{new}}$

      end for

    end for

    $\texttt{states} \leftarrow \texttt{new_states}$

  end for

  $\log S_l \leftarrow -\infty$

  for all $(m, r) \in \texttt{states}$ **do**

    if $m = \alpha$ **then**

      $\log S_{l,r} \leftarrow \sum_{i,j=0}^{p-2} \frac{\mathbf{G}^{(p)}_{i,b-1-j} l_i r_{L-1-j}}{\sigma^2}$

      $\log S_l \leftarrow \text{LSE}\left(\log S_l,\ \texttt{states}(m, r), \log S_{l,r} \right)$

    end if

  end for

  $\log S \leftarrow \text{LSE}(\log S,\ \log S_l)$

end for

$\rho \leftarrow \dfrac{\log S + \log(\alpha!) - \alpha \log b}{\alpha - 1} + \dfrac{\tau \alpha}{2\sigma^2}$

return ρ

D Proofs for Section 3

Lemma D.1. *The Rényi divergence in the "remove" direction $R_\alpha(\hat{P} \,\|\, \hat{Q})$ is given by*

$$\frac{1}{\alpha - 1} \log \sum_{(r_1,\ldots,r_\alpha)\in[1,b]^\alpha} \exp\left(\sum_{\substack{j_1,j_2=1 \\ j_1\neq j_2}}^{\alpha} \frac{\mathbf{G}_{r_{j_1},r_{j_2}}}{2\sigma^2} \right) - \frac{\alpha \log b}{1-\alpha}, \tag{3}$$

where G is the Gram matrix with entries $\mathbf{G}_{i,j} = \langle \mathbf{m}_i, \mathbf{m}_j \rangle$.

Proof. By Lemma 2.8 $(\hat{P}, \hat{Q})$ is a dominating pair for the "add" adjacency, where the density of $\hat{P}$ is given by

$$p(\mathbf{x}) = \frac{1}{b} \sum_{j=1}^{b} \phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I),$$

and the density of $\hat{Q}$ is

$$q(\mathbf{x}) = \phi(\mathbf{x}; \mathbf{0}, \sigma^2 I),$$

where $\phi(\mathbf{x}; \mu, \Sigma)$ denotes the density of the multivariate Gaussian distribution.

Thus, for $\alpha > 1$, the Rényi divergence in the add direction is

$$R_\alpha(\hat{P}||\hat{Q}) = \frac{1}{\alpha-1} \log \int_{\mathbb{R}^N} \frac{p(\boldsymbol{x})^\alpha}{q(\boldsymbol{x})^{\alpha-1}} \, \mathrm{d}\boldsymbol{x} = \frac{1}{\alpha-1} \log \mathbb{E}_{\boldsymbol{x}\sim Q} \frac{p(\boldsymbol{x})^\alpha}{q(\boldsymbol{x})^\alpha} \tag{11}$$

Via multinomial rule and quadratic expansion, we have

$$\begin{aligned}
p(\boldsymbol{x})^\alpha &= \left(\frac{1}{b} \sum_{j=1}^{b} \phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I)) \right)^\alpha = \frac{1}{b^\alpha} \sum_{r_1+\cdots+r_b=\alpha} \binom{\alpha}{r_1,\ldots,r_b} \left(\prod_{j=1}^{b} \phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I)^{r_j} \right) \\
&= \frac{1}{b^\alpha} \sum_{r_1+\cdots+r_b=\alpha} \binom{\alpha}{r_1,\ldots,r_b} \frac{1}{(2\pi\sigma^2)^{N\alpha/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^{b} r_j (\mathbf{x} - \boldsymbol{m}_j)^T (\mathbf{x} - \boldsymbol{m}_j) \right) \\
&= \frac{1}{b^\alpha} \frac{1}{(2\pi\sigma^2)^{N\alpha/2}} \exp\left(-\frac{\alpha}{2\sigma^2} \boldsymbol{x}^T \boldsymbol{x} \right) \sum_{r_1+\cdots+r_b=\alpha} \binom{\alpha}{r_1,\ldots,r_b} \exp\left(\frac{1}{\sigma^2} \boldsymbol{x}^T \left[\sum_{j=1}^{b} r_j \boldsymbol{m}_j \right] \right) \\
&\qquad \times \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^{b} r_j \boldsymbol{m}_j^T \boldsymbol{m}_j \right)
\end{aligned}$$

The $q(\boldsymbol{x})^\alpha$ is simply given by:

$$q(\boldsymbol{x})^\alpha = \frac{1}{(2\pi\sigma^2)^{N\alpha/2}} \exp\left(-\frac{\alpha}{2\sigma^2} \boldsymbol{x}^T \boldsymbol{x} \right).$$

Then the Rényi divergence is computed as:

$$\begin{aligned}
R_\alpha(\hat{P}||\hat{Q}) = \frac{1}{1-\alpha} \log \Bigg(\frac{1}{b^\alpha} \sum_{r_1+\cdots+r_b=\alpha} \binom{\alpha}{r_1,\ldots,r_b} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^{b} r_j \boldsymbol{m}_j^T \boldsymbol{m}_j \right) \\
\times \mathbb{E}_{\boldsymbol{x}\sim Q} \exp\left(\frac{1}{\sigma^2} \boldsymbol{x}^T \left[\sum_{j=1}^{b} r_j \boldsymbol{m}_j \right] \right) \Bigg)
\end{aligned}$$

Using the moment generating function of a multivariate Gaussian, for $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ and any $\mathbf{t} \in \mathbb{R}^N$,

$$\mathbb{E}\big[\exp(\mathbf{t}^\top \mathbf{x}) \big] = \exp\left(\frac{\sigma^2}{2} \|\mathbf{t}\|_2^2 \right),$$

and taking $\mathbf{t} = \frac{1}{\sigma^2}\sum_{j=1}^{b} r_j \mathbf{m}_j$, we obtain

$$R_\alpha(\hat{P}||\hat{Q}) = \frac{1}{1-\alpha}\log\left(\frac{1}{b^\alpha}\sum_{r_1+\cdots+r_b=\alpha}\binom{\alpha}{r_1,\ldots,r_b}\exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{b} r_j \boldsymbol{m}_j^T \boldsymbol{m}_j\right)\right. \\ \left.\times\exp\left(\frac{1}{2\sigma^2}\left\|\sum_{j=1}^{b} r_j \boldsymbol{m}_j\right\|_2^2\right)\right). \tag{12}$$

We change the summation from partitions of α to all possible b^α tuples of α indices in $[1, b]$:

$$\begin{aligned} R_\alpha(\hat{P}||\hat{Q}) &= \frac{1}{1-\alpha}\log\left(\frac{1}{b^\alpha}\sum_{(r_1,\cdots,r_b)\in[1,b]^\alpha}\exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{\alpha}\boldsymbol{m}_j^T\boldsymbol{m}_j\right)\exp\left(\frac{1}{2\sigma^2}\left\|\sum_{j=1}^{\alpha}\boldsymbol{m}_j\right\|_2^2\right)\right) \\ &= \frac{1}{1-\alpha}\log\left(\frac{1}{b^\alpha}\sum_{(r_1,\cdots,r_b)\in[1,b]^\alpha}\exp\left(\frac{1}{2\sigma^2}\sum_{j_1\neq j_2}^{\alpha}\boldsymbol{m}_{r_{j_1}}^T\boldsymbol{m}_{r_{j_2}}\right)\right) \\ &= \frac{1}{1-\alpha}\log\left(\frac{1}{b^\alpha}\sum_{(r_1,\cdots,r_b)\in[1,b]^\alpha}\exp\left(\frac{1}{2\sigma^2}\sum_{j_1\neq j_2}^{\alpha}\mathbf{G}_{r_{j_1},r_{j_2}}\right)\right), \end{aligned}$$

where $\mathbf{G}$ denotes the Gram matrix of vectors $\boldsymbol{m}_j$, concluding the proof. □

Lemma D.2. *Algorithm 4 computes the Rényi divergence based value in eq. 3 exactly for a p-banded strategy matrix $\mathbf{C}$, and returns an upper bound otherwise. The algorithm runs in time $O\big(bp\alpha^{2p}\big)$ and requires $O(\alpha^p + bp)$ memory.*

Proof. To compute the divergence efficiently using dynamic programming, we use the expression for $R_\alpha(\hat{P}||\hat{Q})$ from equation equation 12, substituting the Gram matrix $\mathbf{G}$:

$$\begin{aligned} R_\alpha(\hat{P}||\hat{Q}) &= \log\sum_{r_1+\cdots+r_b=\alpha}\binom{\alpha}{r_1, r_2\ldots r_b}\exp\left(\frac{1}{2\sigma^2}\sum_{j=1}^{b} r_j(r_j-1)\mathbf{G}_{j,j} + \frac{1}{\sigma^2}\sum_{j_1<j_2}^{b}\mathbf{G}_{j_1,j_2}r_{j_1}r_{j_2}\right) \\ &= \log\sum_{r_1+\cdots+r_b=\alpha}\frac{1}{r_1!\ldots r_b!}\exp\left(\frac{1}{2\sigma^2}\sum_{j=1}^{b} r_j(r_j-1)\mathbf{G}_{j,j} + \frac{1}{\sigma^2}\sum_{j_1<j_2}^{b}\mathbf{G}_{j_1,j_2}r_{j_1}r_{j_2}\right) \\ &\quad + \log(\alpha!). \end{aligned} \tag{13}$$

Next, we factor out the dependence on r_b, using that the matrix $\mathbf{G}$ is p cyclic-banded:

$$\begin{aligned} &\log\sum_{r_b=0}^{\alpha}\left[\frac{1}{r_b!}\exp\left(\frac{1}{2\sigma^2}(r_b-1)r_b\mathbf{G}_{b,b} + \frac{1}{\sigma^2}\sum_{j=b-p+1}^{b-1} r_b r_j \mathbf{G}_{j,b} + \frac{1}{\sigma^2}\sum_{j=1}^{\min(p-1,b-p)} r_b r_j \mathbf{G}_{j,b}\right)\right. \\ &\left.\times\sum_{r_1+\cdots+r_{b-1}=\alpha-r_b}\frac{1}{r_1!\ldots r_{b-1}!}\exp\left(\frac{1}{2\sigma^2}\sum_{j=1}^{b-1} r_j(r_j-1)\mathbf{G}_{j,j} + \frac{1}{\sigma^2}\sum_{j_1<j_2}^{b-1}\mathbf{G}_{j_1,j_2}r_{j_1}r_{j_2}\right)\right] \\ &+ \log(\alpha!) \end{aligned} \tag{14}$$

This suggests a dynamic programming formulation. Define

$$F(k,m,\mathbf{l},\mathbf{r}) := \log \sum_{\substack{r_1+\cdots+r_k=m \\ r_1,\ldots r_k\geq 0 \\ (r_{k-p+1},\ldots,r_k)=\mathbf{r} \\ (r_1,\ldots,r_{p-1})=\mathbf{l}}} \frac{1}{r_1!\ldots r_k!} \exp\left(\frac{1}{2\sigma^2}\sum_{j=1}^{k} r_j(r_j-1)\mathbf{G}_{j,j} + \frac{1}{\sigma^2}\sum_{j_1<j_2}^{k} \mathbf{G}_{j_1,j_2} r_{j_1} r_{j_2}\right), \tag{15}$$

where $\mathbf{l}$ is the prefix of length $p-1$ (empty if we run DP-SGD with $p=1$), and $\mathbf{r}$ is the suffix of length $p-1$ starting at position $k-p+1$.

If we compute all $O(b\alpha^{2p-1})$ values of F, then the divergence is:

$$R_\alpha(\hat{P}||\hat{Q}) = \frac{1}{\alpha-1}\log \sum_{\|\mathbf{l}\cup\mathbf{r}\|_1\leq\alpha} \exp(F(B,\alpha,\mathbf{l},\mathbf{r})) \exp\left(\frac{1}{\sigma^2}\sum_{j_1=1}^{p-1}\sum_{j_2=b-p+2}^{b} \mathbf{G}_{j_1,j_2} l_{j_1} r_{j_2}\right) + \frac{\log(\alpha!)}{\alpha-1} - \frac{\alpha\log b}{\alpha-1}. \tag{16}$$

The interaction for the indices $1 \leq j_1 < j_2 \leq b$ are accounted in the dynamic function F, and to compute the whole sum we would have to account for the cyclic interaction between the last $p-1$ indices and the first $p-1$ indices. Note that we must fix the prefix $\mathbf{l}$; otherwise, when computing the cyclic interaction, we would obtain conflicting values for the first $p-1$ entries. Although this increases the computational complexity, we view it as an unavoidable cost of cyclic interaction.

To compute the dynamic values F, we use dynamic programming in a forward direction. We go in a cycle over all possible prefix values $\mathbf{l} = (r_1,\ldots,r_{p-1})$. Given a state $(k,m,\mathbf{l},\mathbf{r})$, we update all successor states $(k+1, m+r_{k+1}, \mathbf{l}, (r_{k-p+3},\ldots,r_{k+1}))$ via

$$\begin{aligned} &F(k+1, m+r_{k+1}, \mathbf{l}, (r_{k-p+3},\ldots,r_{k+1})) \\ &\quad \mathrel{+}= F(k,m,\mathbf{l},(r_{k-p+2},\ldots r_k)) - \log(r_{k+1}!) + \frac{1}{2\sigma^2}(r_{k+1}-1)r_{k+1}\mathbf{G}_{k+1,k+1} \\ &\qquad + \frac{1}{\sigma^2}\sum_{j_1=k-p+2}^{k} r_{k+1} r_{j_1} \mathbf{G}_{j_1,k+1}. \end{aligned} \tag{17}$$

Each of $O(b\cdot\alpha\cdot\alpha^{p-1}\cdot\alpha^{p-1})$ states can be computed in $O(p)$ time, with $O(\alpha)$ forward transitions, resulting in overall runtime $O(bp\alpha^{2p})$. Regarding the memory requirements, we loop over the prefix values $\mathbf{l}$ and therefore do not need to store values for previously processed prefixes. Likewise, since we increase k iteratively, we do not need to retain values from earlier k. Instead, we store all current suffixes (there are $O(\alpha^{p-1})$ of them), each with at most $O(\alpha)$ distinct partial sums, for a total of $O(\alpha^p)$ memory. In addition, we must store the p-cyclic banded matrix $\mathbf{G}$, which requires $O(bp)$ memory, resulting in an overall requirement of $O(\alpha^p + bp)$ memory.

Algorithm 4 computes the sum in equation 13 via dynamic programming under the assumption that the Gram matrix $\mathbf{G}$ is p cyclically banded, and otherwise adds the nonzero term $\frac{\tau\alpha}{2\sigma^2}$ to obtain an upper bound. Concretely, given a matrix $\mathbf{C}$, it first computes the Gram matrix $\mathbf{G}$, then forms the truncated p cyclically banded matrix $\mathbf{G}^{(p)}$ together with the error term τ.

The indices in the sum form a cyclic interaction. To compute the sum dynamically, we first break the cycle by looping over all prefixes $\mathbf{l} = (l_0,\ldots,l_{p-2})$ whose total sum is less than the target sum α. For each fixed prefix, we precompute the log sum of exponentials corresponding to the interactions among the first $p-1$ indices. We then expand the range of indices up to $b-1$ iteratively by running a forward dynamic program over $k = p-1,\ldots,b-1$. At step k, the set of dynamic states is stored in a dictionary, where each state is identified by (i) the current suffix $\mathbf{r}$ of the previous $p-1$ values and (ii) the current partial sum m of indices up to position $k-1$. From a state $(\mathbf{r},m)$ we propose a next value $t \in \{0,\ldots,\alpha-m\}$, compute the contribution corresponding to interactions involving the new index k, and update the suffix, the partial sum, and the accumulated log value. For numerical stability, we aggregate contributions using the log-sum-exp (LSE) operation. The resulting states are stored in a temporary dictionary $\mathrm{new_states}$, and states is updated after all admissible forward values have been considered. This completes the computation of all interactions between indices

$0 \le i < j \le b-1$ except for the cyclic terms. Finally, we close the cycle by adding the interactions between the final suffix and the fixed prefix in a last loop. Summing the resulting log values over all valid choices of the prefix $\mathbf{l}$ completes the computation.

□

Lemma D.3. *The Rényi divergence in the "add" direction is bounded as*

$$\mathrm{R}_\alpha(\hat{Q}\|\hat{P}) \le \frac{1}{2b\sigma^2}\sum_{j=1}^{b}\mathbf{G}_{j,j} \;+\; \frac{\alpha-1}{2b^2\sigma^2}\sum_{i=1}^{b}\sum_{j=1}^{b}\mathbf{G}_{i,j}, \tag{5}$$

where $\mathbf{G}$ *is the Gram matrix with entries* $\mathbf{G}_{i,j} = \langle \mathbf{m}_i, \mathbf{m}_j \rangle$.

Proof. If $(\hat{P}, \hat{Q})$ is a dominating pair for the "remove" adjacency, then Lemma 29 of Zhu et al. [39] implies that $(\hat{Q}, \hat{P})$ is a dominating pair for the "add" adjacency. The density of $\hat{P}$ is

$$p(\mathbf{x}) = \frac{1}{b}\sum_{j=1}^{b}\phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I),$$

and the density of $\hat{Q}$ is

$$q(\mathbf{x}) = \phi(\mathbf{x}; \mathbf{0}, \sigma^2 I),$$

where $\phi(\mathbf{x}; \mu, \Sigma)$ denotes the density of the multivariate Gaussian distribution.

Thus, for $\alpha > 1$, the Rényi divergence in the remove direction is

$$\mathrm{R}_\alpha(Q\|P) = \frac{1}{\alpha-1}\log\int_{\mathbb{R}^b}\frac{q(\mathbf{x})^\alpha}{p(\mathbf{x})^{\alpha-1}}\,d\mathbf{x}. \tag{18}$$

The mixture density $p(\mathbf{x})$ appears in the denominator, which makes the integral difficult to compute in closed form. We therefore upper bound it.

By the arithmetic mean–geometric mean inequality (equivalently, Jensen's inequality for the exponential function),

$$p(\mathbf{x}) = \frac{1}{b}\sum_{j=1}^{b}\phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I) \;\ge\; \left(\prod_{j=1}^{b}\phi(\mathbf{x}; \mathbf{m}_j, \sigma^2 I)\right)^{1/b} \tag{19}$$

$$= \frac{1}{(2\pi\sigma^2)^{N/2}}\exp\left(-\frac{1}{2b\sigma^2}\sum_{j=1}^{b}(\mathbf{x}-\mathbf{m}_j)^\top(\mathbf{x}-\mathbf{m}_j)\right), \tag{20}$$

and

$$q(\mathbf{x}) = \phi(\mathbf{x}; \mathbf{0}, \sigma^2 I) = \frac{1}{(2\pi\sigma^2)^{N/2}}\exp\!\left(-\frac{1}{2\sigma^2}\mathbf{x}^\top\mathbf{x}\right). \tag{21}$$

Substituting the lower bound on $p(\mathbf{x})$ into the expression for $\mathrm{R}_\alpha(\hat{Q}\|\hat{P})$ yields

$$\mathrm{R}_\alpha(\hat{Q}\|\hat{P}) \le \frac{1}{\alpha-1}\log\int_{\mathbb{R}^N} q(\mathbf{x})\exp\left(\frac{\alpha-1}{2b\sigma^2}\sum_{j=1}^{b}\left(-2\mathbf{x}^\top\mathbf{m}_j + \mathbf{m}_j^\top\mathbf{m}_j\right)\right)d\mathbf{x} \tag{22}$$

$$= \frac{1}{2b\sigma^2}\sum_{j=1}^{b}\mathbf{m}_j^\top\mathbf{m}_j + \frac{1}{\alpha-1}\log\mathbb{E}_{\mathbf{x}\sim Q}\exp\left\{-\frac{\alpha-1}{b\sigma^2}\left(\sum_{j=1}^{b}\mathbf{m}_j\right)^\top\mathbf{x}\right\}. \tag{23}$$

Using the moment generating function of a multivariate Gaussian, for $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ and any $\mathbf{t} \in \mathbb{R}^N$,

$$\mathbb{E}\big[\exp(\mathbf{t}^\top\mathbf{x})\big] = \exp\!\left(\frac{\sigma^2}{2}\|\mathbf{t}\|_2^2\right),$$

and taking $\mathbf{t} = -\frac{\alpha-1}{b\sigma^2}\sum_{j=1}^{b}\mathbf{m}_j$, we obtain

$$\mathrm{R}_\alpha(\hat{Q}\|\hat{P}) \le \frac{1}{2b\sigma^2}\sum_{j=1}^{b}\mathbf{m}_j^\top\mathbf{m}_j + \frac{1}{\alpha-1}\cdot\frac{\sigma^2}{2}\left\|\frac{\alpha-1}{b\sigma^2}\sum_{j=1}^{b}\mathbf{m}_j\right\|_2^2 \tag{24}$$

$$= \frac{1}{2b\sigma^2}\sum_{j=1}^{b}\mathbf{m}_j^\top\mathbf{m}_j + \frac{\alpha-1}{2b^2\sigma^2}\left\|\sum_{j=1}^{b}\mathbf{m}_j\right\|_2^2. \tag{25}$$

Finally, let G be the Gram matrix with entries $\mathbf{G}_{i,j} = \langle\mathbf{m}_i, \mathbf{m}_j\rangle$. Then

$$\sum_{j=1}^{b}\mathbf{m}_j^\top\mathbf{m}_j = \sum_{j=1}^{b}\mathbf{G}_{j,j}, \qquad \left\|\sum_{j=1}^{b}\mathbf{m}_j\right\|_2^2 = \sum_{i=1}^{b}\sum_{j=1}^{b}\mathbf{G}_{i,j},$$

which proves the claim. □

Lemma D.4. *The sum*

$$\sum_{(r_1,\ldots,r_\alpha)\in[b]^\alpha}\exp\left(\sum_{\substack{j_1,j_2=1\\ j_1\neq j_2}}^{\alpha}\frac{\mathbf{G}_{r_{j_1},r_{j_2}}}{2\sigma^2}\right) \tag{26}$$

appearing in Lemma 4.2 is $\#P$-complete to compute for a general matrix $\mathbf{G}\in\mathbb{R}^{b\times b}$.

Proof. Suppose, for the sake of contradiction, that the sum in equation 26 can be computed in polynomial time for an arbitrary matrix $\mathbf{G}$.

Let H be a graph on b vertices. We construct a matrix $\mathbf{G}$ from H as follows:

$$\mathbf{G}_{i,j} = \begin{cases} 0, & \text{if } i\neq j \text{ and } (i,j)\in E(H),\\ -\infty, & \text{otherwise.}\end{cases}$$

Then, for any tuple $(r_1,\ldots,r_\alpha)\in[b]^\alpha$, the corresponding term in equation 26 is equal to 1 if the vertices $r_1,\ldots,r_\alpha$ are distinct and form an α-clique in H, and is equal to 0 otherwise. Hence the sum in equation 26 counts the number of ordered α-cliques in H.

Since each unordered α-clique contributes exactly $\alpha!$ ordered tuples, computing equation 26 would allow us to count the number of α-cliques in H in polynomial time. This is a $\#P$-complete problem. Therefore, computing the sum in equation 26 is $\#P$-complete. □

E Proofs for Section 4

In the following, we first confirm that the conditional composition Lemma 5.1 immediately follows as a special case of the general conditional composition framework from [11]. In the next sections, we then provide the omitted proofs for our instantiation of the conditional composition framework.

The general conditional composition result is:

Theorem E.1 (Theorem 3.1 in [11])**.** *Let $\mathcal{M}_1 : \mathcal{D} \to \mathcal{X}_1, \mathcal{M}_2 : \mathcal{X}_1 \times \mathcal{D} \to \mathcal{X}_2, \mathcal{M}_3 : \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{D} \to \mathcal{X}_3, \dots \mathcal{M}_N$ be a sequence of adaptive mechanisms, where each $\mathcal{M}_n$ takes a dataset in $\mathcal{D}$ and the output of mechanisms $\mathcal{M}_1, \dots, \mathcal{M}_{n-1}$ as input. Let $\mathcal{M}$ be the mechanism that outputs $(x_1 = \mathcal{M}_1(D), x_2 = \mathcal{M}_2(x_1, D), \dots, x_N = \mathcal{M}_N(x_1, \dots, x_{N-1}, D))$. Fix any two adjacent datasets D, D'.*

Suppose there exists "bad events" $E_1 \subseteq \mathcal{X}_1, E_2 \subseteq \mathcal{X}_1 \times \mathcal{X}_2, \dots ...E_{N-1} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_{N-1}$ such that

$$\Pr_{x \sim \mathcal{M}(D)} [\exists n : (x_1, x_2, \dots x_n) \in E_n] \leq \delta$$

and pairs of distributions $(P_1, Q_1), (P_2, Q_2), \dots (P_N, Q_N)$ such that the PLD of $\mathcal{M}_1(D)$ and $\mathcal{M}_1(D')$ is dominated by the PLD of $P^{(1)}, Q^{(1)}$ and for any $n \geq 1$ and "good" output $(x_1, x_2, \dots x_n) \notin E_n$, the PLD of $\mathcal{M}_{n+1}(x_1, \dots, x_n, D)$ and $\mathcal{M}_{n+1}(x_1, \dots, x_n, D')$ is dominated by the PLD of $P^{(n+1)}, Q^{(n+1)}$. Then for all ε:

$$H_\varepsilon(\mathcal{M}(D), \mathcal{M}(D')) \leq H_\varepsilon \left(P^{(1)} \times P^{(2)} \times \dots \times P^{(N)}, Q^{(1)} \times Q^{(2)} \times \dots \times Q^{(N)} \right) + \delta.$$

In our case, we do not have a mechanism, but only its dominating pair (P, Q) from Lemma 2.8. However, for any D, D', we can define the dummy mechanism $\mathcal{M}(D) = P$ and $\mathcal{D}' = Q$ and $M_n(x_1, \dots, x_n, D) = P_{x_n | x_{1:n-1}}$ and $M_n(x_1, \dots, x_n, D') = Q_{x_n | x_{1:n-1}}$. The special case below then follows immediately.

Lemma E.2. *[Special case of Theorem 3.1 in [11]] Let P, Q be distributions on $\mathbb{R}^N$. For every step $n \in N$, choose a pair of distributions $P^{(n)}, Q^{(n)}$. Let $A^{(n)} \subseteq \mathbb{R}^N$ refer to all "good" outcomes $\tilde{y}$ such that the conditional distributions $P_{y_n | \tilde{y}_{1:n-1}}$ and $Q_{y_n | \tilde{y}_{1:n-1}}$ are dominated by $P^{(n)}, Q^{(n)}$. Define "bad" outcomes $E = \bigcup_{n=1}^N \overline{A^{(n)}}$ with probability $\delta_E = P(E)$. Then,*

$$H_\gamma(P || Q) \leq H_\gamma \left(\bigtimes_{n=1}^N P^{(n)} || \bigtimes_{n=1}^N Q^{(N)} \right) + \delta_E. \tag{6}$$

E.1 Dominance criterion

Our dominance criterion that relates stochastic dominance between mixture-of-Gaussians mechanisms to a pointwise comparison between reverse hazard functions is derived from the following result:

Lemma E.3. *[Corollary 4.4 in [11]] Define distributions $P = \sum_{i=1}^b \mathcal{N}(\mu_i, \sigma) p(z = i)$ and $Q = \mathcal{N}(0, \sigma)$ with $0 \leq \mu_b \leq \dots \leq \mu_b$. Let $P' = \sum_{i=1}^b \mathcal{N}(\mu_i, \sigma) p'(z = i)$. Then $(P, Q) \prec (P', Q)$ and $(Q, P) \prec (Q, P')$ whenever $p(z)$ is stochastically dominated by $p'(z)$, i.e., $P(z \geq i) \leq P'(z \geq i)$ for all $i \in [b]$.*

Next, we show that if the reverse hazard function associated with mixture weights $p(z)$ is pointwise smaller than that associated with $p(z')$, then we have stochastic dominance, i.e., the criterion in Lemma E.3 is fulfilled:

Lemma E.4. *Given $p, p' : [b] \to [0, 1]$, define the corresponding reverse hazard functions at i as $\lambda_i = P(z = i \mid z \leq i)$ and $\lambda'_i = P'(z = i \mid z \leq i)$. If $\lambda_i \leq \lambda'_i$ for all $i \in [b]$, then $p(z)$ is stochastically dominated by p'.*

Proof. Let $F_i = P(z \leq i)$ denote the CDF of z. By definition, $\lambda_i = P(z = i \mid z \leq i) = P(z = i)/F_i$. Furthermore, $F_i = F_{i+1} - P(z = i + 1)$. It thus follows that $F_i = F_{i+1} - P(z = i + 1) = F_{i+1}(1 - \lambda_{i+1})$. Since $F_i = 1$ for any $i \geq b$, recursive application of this equality yields $F_i = \prod_{j=i+1}^b (1 - \lambda_j)$. The condition $\lambda_j \leq \lambda'_j$ implies $(1 - \lambda_j) \geq (1 - \lambda'_j)$ for all j. Consequently, $F_i \geq F'_i$ for all i, i.e., $p(z)$ is stochastically dominated by $p'(z)$. □

Our result then immediately follows from combining the previous two lemmata:

Lemma E.5. *Assume non-decreasing mixture means $0 \leq m_{1,n} \leq \cdots \leq m_{b,n}$ in Equations* (7) *and* (8)*. If the posterior reverse hazard function is pointwise less or equal, i.e.,*

$$P(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) \leq P^{(n)}(z = i \mid z \leq i) \quad \textit{for all } 1 \leq i \leq b, \tag{9}$$

then the distributions from Equation (8) *dominate those from Equation* (7)*, i.e., $(P_{y_n|y_{1:n-1}}, Q_{y_n|y_{1:n-1}}) \prec (P^{(n)}_{y_n}, Q^{(n)}_{y_n})$ and $(Q_{y_n|y_{1:n-1}}, P_{y_n|y_{1:n-1}}) \prec (Q^{(n)}_{y_n}, P^{(n)}_{y_n})$.*

Proof. Instantiate Lemma E.4 with $\lambda_i = P(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1})$ and $\lambda'_i = P^{(n)}(z = i \mid z \leq i)$. Having $\forall i[b] : \lambda_i \leq \lambda'_i$ then implies stochastic dominance via Lemma E.4, i.e. $\forall i \in [b] : P(z \geq i \mid P(z = i \mid \boldsymbol{y}_{1:n-1}) \leq P(z \geq i)$. Thus, the conditions from Lemma E.3 are met, i.e., this stochastic dominance translated to dominance in the differential privacy sense. □

E.2 Analytical bounds

First, we prove that the reverse hazard function underlying our dominance criterion is equivalent to a ternary form of privacy loss between Gaussian mixtures. Then, we prove our analytical tail bounds on this privacy loss.

Lemma E.6. *Let $s(x) = (1 + e^{-x})^{-1}$ be the logistic sigmoid function. For any $i \in [b]$, the conditional reverse hazard function is $P(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) = s(-\log(i-1) - L_i(\boldsymbol{y}_{1:n-1}))$ with*

$$L_i(\boldsymbol{y}_{1:n-1}) = \log\left(\frac{p(\boldsymbol{y}_{1:n-1} \mid z < i)}{p(\boldsymbol{y}_{1:n-1} \mid z = i)}\right) = \log\left(\frac{\sum_{j=1}^{i-1} \mathcal{N}(\boldsymbol{y}_{1:n-1} \mid \boldsymbol{m}_{j,:n-1}, \sigma^2 \boldsymbol{I}) \frac{1}{i-1}}{\mathcal{N}(\boldsymbol{y}_{1:n-1} \mid \boldsymbol{m}_{i,:n-1}, \sigma^2 \boldsymbol{I})}\right). \tag{10}$$

Proof. Let $p(\boldsymbol{y}_{1:n-1} \mid z = j) = \mathcal{N}(\boldsymbol{y}_{1:n-1} \mid \boldsymbol{m}_{j,:n-1}, \sigma^2 \boldsymbol{I})$ denote the likelihood of the earlier outcome $\boldsymbol{y}_{1:n-1}$ conditioned on mixture component j. Let $p(z = j) = \frac{1}{b}$ denote the uniform prior that arises from random allocation with b batches per epoch. Using the definition of conditional probability and Bayes' rule, the conditional reverse hazard function is:

$$\begin{aligned}
&\Pr(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) \\
=&\frac{p(\boldsymbol{y}_{1:n-1} \mid z = i) p(z = i)}{\sum_{j=1}^{i} p(\boldsymbol{y}_{1:n-1} \mid z = j) p(z = j)} \\
=&\frac{p(\boldsymbol{y}_{1:n-1} \mid z = i)}{\sum_{j=1}^{i} p(\boldsymbol{y}_{1:n-1} \mid z = j)} \\
=&\left(1 + \frac{\sum_{j=1}^{i-1} p(\boldsymbol{y}_{1:n-1} \mid z = j)}{p(\boldsymbol{y}_{1:n-1} \mid z = i)}\right)^{-1}.
\end{aligned}$$

Recall that $L_i(\boldsymbol{y}_{1:n-1}) = \log\left(\frac{\frac{1}{i-1}\sum_{j=1}^{i-1} p(\boldsymbol{y}_{1:n-1}|z=j)}{p(\boldsymbol{y}_{1:n-1}|z=i)}\right)$. Multiplying and dividing the second summand by $i - 1$ yields:

$$\frac{\sum_{j=1}^{i-1} p(\boldsymbol{y}_{1:n-1} \mid z = j)}{p(\boldsymbol{y}_{1:n-1} \mid z = i)} = (i-1)\exp(L_i(\boldsymbol{y}_{1:n-1})) = \exp(\log(i-1) + L_i(\boldsymbol{y}_{1:n-1})). \tag{27}$$

Applying the definition of the sigmoid function $s(x) = (1 + e^{-x})^{-1}$, we obtain:

$$\Pr(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) = s\left(-\big(\log(i-1) + L_i(\boldsymbol{y}_{1:n-1})\big)\right), \tag{28}$$

which completes the proof. □

Theorem 5.5. *Let $\boldsymbol{y} \sim R$ with $R = Q$ ("add') as defined in Lemma 2.8. Define standard normal CDF $\Phi : \mathbb{R} \to [0, 1]$ and $\psi = \text{Uniform}([i-1])$ Then $\Pr[L_i(\boldsymbol{y}_{1:n-1}) \leq \tau_i] \leq \Phi\left(\frac{\tau_i - \nu}{\xi}\right)$ with*

$$\nu = (||\boldsymbol{m}_{i,:n-1}||_2^2 - \mathbb{E}_{j\sim\psi}[||\boldsymbol{m}_{j,:n-1}||_2^2])] \,/\, (2\sigma^2), \qquad \xi = ||\boldsymbol{m}_{i:n-1} - \mathbb{E}_{j\sim\psi}[\boldsymbol{m}_{j,:n-1}]||_2 \,/\, \sigma.$$

Proof. This result directly follows as a special case of the general variational tail bound Theorem F.5 by defining $\boldsymbol{\mu}_i = \boldsymbol{m}_{i,1:n-1}$ for all $i \in [b]$ and using variational distribution $\psi \sim \mathrm{Uniform}(\{1, \ldots, i-1\})$. In particular, this causes the term $\mathrm{KL}(\psi||\pi)$ to vanish.

Alternatively, this result could be proven by (1) applying the arithmetic-mean geometric-mean inequality to the logarithm in Equation (10) to obtain a pointwise lower bound, (2) noticing that the privacy loss between any two Gaussian mixture components is an affine function of $\boldsymbol{y}_{:n-1}$, and finally (3) use the standard formula for the distribution of affine transformations of a multivariate Gaussians. □

Theorem E.7. *Let $\boldsymbol{y} \sim R$ with $R = Q$ ("add') as defined in Lemma 2.8. Define standard normal CDF $\Phi : \mathbb{R} \to [0, 1]$ and $\psi = \mathrm{Uniform}([i-1])$ Then $\Pr[L_i(\boldsymbol{y}_{1:n-1}) \leq \tau_i] \leq \frac{1}{b} \sum_{k=1}^{b} \Phi\left(\frac{\tau_i - \nu_k}{\xi_k}\right)$ with*

$$\nu_k = \frac{1}{\sigma^2} \boldsymbol{m}_{k,:n-1}^T (\mathbb{E}_{j\sim\psi}[\boldsymbol{m}_{j:n-1}] - \boldsymbol{m}_{i,:n-1}) + \frac{1}{2\sigma^2} \left(\|\boldsymbol{m}_{i,:n-1}\|_2^2 - \mathbb{E}_{j\sim\psi}[\|\boldsymbol{m}_{j,:n-1}\|_2^2] \right),$$
$$\xi_k = \frac{1}{\sigma} \|\boldsymbol{m}_{i,:n-1} - \mathbb{E}_{j\sim\psi}[\boldsymbol{m}_{j,:n-1}]\|_2.$$

Proof. This result directly follows as a special case of the general variational tail bound Theorem F.3 by defining $\boldsymbol{\mu}_i = \boldsymbol{m}_{i,1:n-1}$ for all $i \in [b]$ and using variational distribution $\psi \sim \mathrm{Uniform}(\{1, \ldots, i-1\})$. In particular, this causes the term $\mathrm{KL}(\psi||\pi)$ to vanish. □

Remark E.8. Note that since the "remove" bound is a weighted sum of Gaussian CDFs, we can no longer find τ_i for a desired significance analytically via the inverse CDF. Due to monotonicity of standard normal CDF Ψ, we can nevertheless find τ_i with arbitrary precision via bisection.

E.3 Correctness of Algorithm 2

While Section 5.1 in its entirety already served as a proof sketch, we now provide a formal proof for the correctness of Algorithm 3:

Theorem 5.6. *For the "remove" relation, the distributions returned by Algorithm 3 dominate $P_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ and $Q_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ with probability $1-\beta$ under P. For the "add" relation, the returned distributions dominate $Q_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ and $P_{y_n|\tilde{\boldsymbol{y}}_{1:n-1}}$ with probability $1-\beta$ under Q.*

Proof. Algorithm 3 allocates a failure probability β_i to each component i such that $\sum_i \beta_i = \beta$. For each i, the subroutine `whp_lower` returns a threshold τ_i such that $\Pr[L_i(\boldsymbol{y}_{1:n-1}) < \tau_i] \leq \beta_i$, where the probability is over the randomness of the privacy loss random variable $L_{\tilde{P},\tilde{Q},\tilde{R}}$ generated by the algorithm's reference distribution $\tilde{R}$. In the "remove" case, $\tilde{R}$ is the joint distribution $P_{\boldsymbol{y}_{1:n-1}}$ of the first $n-1$ outcomes of $\boldsymbol{y} \sim P$. In the "add" case, $\tilde{R}$ is the joint distribution $Q_{\boldsymbol{y}_{1:n-1}}$ of the first $n-1$ outcomes of $\boldsymbol{y} \sim Q$.

Define the failure event $\overline{A} = \bigcup_i \{L_i(\boldsymbol{y}_{1:n-1}) < \tau_i\}$ with L_i defined in Lemma 5.3. By a union bound, $\Pr(\overline{A}) \leq \sum \beta_i = \beta$.

Conditioned on the success event $A = \overline{\overline{A}}$, we have $L_i(\boldsymbol{y}_{1:n-1}) \geq \tau_i$ for all i. Consequently, by the monotonicity of the sigmoid function and the derivation in Lemma 5.3, the true conditional reverse hazard satisfies:

$$\Pr(z = i \mid z \leq i, \boldsymbol{y}_{1:n-1}) = s(-\log(i-1) - L_i) \leq s(-\log(i-1) - \tau_i) = \overline{\lambda_i}. \tag{29}$$

Since the means are sorted non-decreasingly (Line 3 of Algorithm 3), the conditions of Lemma E.4 (larger reverse hazard implies dominance in the differential privacy sense) are satisfied. Thus, in the "remove" case the returned pair $(P^{(n)}, Q^{(n)})$ dominates the conditional distributions $P_{y_n|\tilde{y}_{1:n-1}}$ and $Q_{y_n|\tilde{y}_{1:n-1}}$ with probability at least $1-\beta$ under P. In the "add" case, the returned pair $(Q^{(n)}, P^{(n)})$ dominates the conditional distributions $Q_{y_n|\tilde{y}_{1:n-1}}$ and $P_{y_n|\tilde{y}_{1:n-1}}$ with probability at least $1-\beta$ under Q. □

E.4 Runtime complexity of Algorithm 2

In the following, we explain in more detail why executing Algorithm 3 once for every iteration $n \in \{1, \dots, N\}$ leads to an overall complexity of $\mathcal{O}(N^2 b^2)$. Then, we explain why subsequent evaluations for different noise multipliers $\sigma \in sR_+$ can be done in $\mathcal{O}(Nb^2)$ ("remove" direction) or $\mathcal{O}(Nb)$ ("add" direction).

Initial call complexity. When executing Appendix E.1 for a specific iteration n, we first sort the entries of $\boldsymbol{m}_{:,n} \in \mathbb{R}^b$, i.e., we have an initial cost of $\mathcal{O}(b \log b)$. Then, we calculate one privacy loss tail bound per mixture component $i \in \{1, \dots, b\}$ using bisection with κ steps on the analytical bound from Theorems 5.5 and E.7. The main computational cost of evaluating these bounds are the two expectations $\mathbb{E}_{j \sim \psi^{(i)}}[\boldsymbol{m}_{j:n-1}]$ and $\mathbb{E}_{j \sim \psi^{(i)}}[\|\boldsymbol{m}_{j,:n-1}\|_2^2]$ with $\psi^{(i)} = \mathrm{Uniform}([i-1])$. Since they are weighted prefix sums of each other, all of them can be calculated jointly for all $i \in [b]$ at an overall cost of $\mathcal{O}(Nb)$. In the "add" case, we can then evaluate ν and ξ once at a cost of $\mathcal{O}(N)$ and then do bisection on $\Phi\left(\frac{\tau_i - \nu}{\xi}\right)$ at a constant cost $\mathcal{O}(\kappa)$. In the "remove" case, we can evaluate all $(\nu_k)_{k=1}^b$ and $(\xi_k)_{k=1}^b$ once at a cost of $\mathcal{O}(N \cdot b)$ and then di bisection $\frac{1}{b}\sum_{k=1}^b \Phi\left(\frac{\tau_i - \nu_k}{\xi_k}\right)$ at a cost of $\mathcal{O}(b\kappa)$. Thus, the cost for a specific iteration n is:

- Add: $\mathcal{O}(b \log b) + Nb + Nb + b\kappa) = \mathcal{O}(Nb)$,
- Remove: $\mathcal{O}(b \log b) + Nb + Nb^2 + b^2\kappa) = \mathcal{O}(Nb^2)$.

since the number of participation patterns b is never larger than the number of training steps N. Multipliying by the overall number of iterations than yields $\mathcal{O}(N^2 b)$ and $\mathcal{O}(N^2 b^2)$ for the "add" and "remove" direction.

Subsequent call complexity As can be seen from our previous discussion, the computational cost for each iteration $n \in [N]$ and component $i \in [b]$ is dominated by the cost of calculating ν and ξ ("add" case) or $(\nu_k)_{k=1}^b$ and $(\xi_k)_{k=1}^b$ ("remove direction") because it involves vector operations along the n-dimension. However, these terms only depend on noise multiplier σ via multiplication. Thus, after we have computed them once, we can simply scale the existing values for any new σ'. This reduces the runtime for a specific iteration n to:

- Add: $\mathcal{O}(b \log b + b\kappa) = \mathcal{O}(b \log b)$
- Remove: $\mathcal{O}(b \log b + b^2\kappa) = \mathcal{O}(b^2)$

Thus, the cost for evaluation across all iterations is $\mathcal{O}(Nb \log b)$ and $\mathcal{O}(Nb^2)$, respectively.

F Variational tail bounds on ternary privacy loss distribution

In the following, we show that we can attain a wider (and, as shown in Appendix G potentially tighter) range of privacy loss distribution tail bounds than the AM-GM that can be used to derive our previous reuslts. The proof strategy is as follows: We apply Jensen's inequality / a pointwise variational bound to the privacy loss, observe that the resultant function is an affine transformation of $\boldsymbol{y} \sim P$ or $\boldsymbol{y} \sim Q$, and then determine the distribution of this Gaussian or Gaussian mixture after transformation.

Lemma F.1. *Let $\tilde{P} = \sum_{j=1}^{i-1} \mathcal{N}(\boldsymbol{\mu}_j, \sigma^2 \boldsymbol{I}) \cdot \pi_i$ with weights $\pi_i = \frac{1}{i-1}$ and $\tilde{Q} = \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 \boldsymbol{I})$. Let $L_i(\boldsymbol{x}) = \log(\frac{\mathrm{d}\tilde{P}}{\mathrm{d}\tilde{Q}})(\boldsymbol{x})$ be their privacy loss at $\boldsymbol{x}$. Define per-component privacy losses $L_{i,j}(\boldsymbol{x}) = \log\left(\frac{\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \sigma^2\boldsymbol{I})}{\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \sigma^2\boldsymbol{I})}\right)$. Let ψ be any categorical distribution on $[i-1]$. Then, $L_i(\boldsymbol{x}) \geq \underline{L_i}(\boldsymbol{x})$ with*

$$\underline{L_i}(\boldsymbol{x}) = \mathbb{E}_{j\sim\psi}[L_{i,j}] - \mathrm{KL}(\psi||\pi). \tag{30}$$

Proof. Let p be the density of P and q be the density of Q. By definition of the privacy loss L_i and the product rule for logarithms, we have

$$L_i(\boldsymbol{x}) = \log\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) = \log\left(\sum_{j=1}^{i-1} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \sigma^2\boldsymbol{I})\pi_i\right) - \log(\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \sigma^2\boldsymbol{I}).$$

The first term is simply the log-likelihood of a Gaussian mixture model. It is, quasi by definition, lower-bounded by the evidence lower bound (for the formal proof, see, e.g., Section 2.2 of [7]). We thus have

$$\begin{aligned}
L_i(\boldsymbol{x}) &\geq \mathbb{E}_{j\sim\psi}\left[\log\left(\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \sigma^2\boldsymbol{I})\right)\right] - \mathrm{KL}(\psi||\pi) - \log(\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \sigma^2\boldsymbol{I}) \\
&= \mathbb{E}_{j\sim\psi}\left[\log\left(\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \sigma^2\boldsymbol{I})\right) - \log(\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_i, \sigma^2\boldsymbol{I})\right] - \mathrm{KL}(\psi||\pi) \\
&= \mathbb{E}_{j\sim\psi}\left[L_{i,j}(\boldsymbol{x})\right] - \mathrm{KL}(\psi||\pi) \\
&= \underline{L_i}(\boldsymbol{x}),
\end{aligned}$$

where the first equality is due to linearity of expectation. □

Next, we use this bound on the privacy loss L_i to derive a lower tail bound on $L_i(\boldsymbol{x})$ with $\boldsymbol{x} \sim \tilde{R}$ where $\tilde{R}$ is an arbitrary Gaussian mixture. The concrete instantiations for the "remove" and "add" $\tilde{R}$ from Algorithm 3 will then follow as special cases.

Lemma F.2. *Let $\boldsymbol{x} \sim \tilde{R}$ with $\tilde{R} = \sum_{k=1}^{K} \mathcal{N}(\boldsymbol{\rho}_k, \sigma^2\boldsymbol{I})\frac{1}{K}$. Let $\Psi = \{\psi^{(1)}, \ldots, \psi^{(K)}\}$ be a collection of variational distributions on $[i-1]$. Consider the random variable Y generated by sampling a component index $k \sim \mathrm{Uniform}([K])$, sampling $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\rho}_k, \sigma^2\boldsymbol{I})$, and computing the component-specific lower bound $\underline{L_i}(\boldsymbol{x}; \psi^{(k)})$ as defined in Equation* (30)*. Then $Y \sim \sum_{k=1}^{K} \mathcal{N}(\nu_k, \xi_k)\frac{1}{K}$ with component-specific means $\nu_k \in \mathbb{R}$ and standard deviations $\xi_k \in \mathbb{R}_+$ defined by:*

$$\begin{aligned}
\nu_k &= \frac{1}{\sigma^2}\boldsymbol{\rho}_k^T(\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i) + \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j\sim\psi^{(k)}}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi^{(k)} \,||\, \pi), \\
\xi_k &= \frac{1}{\sigma}\|\boldsymbol{\mu}_i - \mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j]\|_2.
\end{aligned}$$

Proof. We first derive a bound for each component k. Recall that the per-component privacy loss is $L_{i,j}(\boldsymbol{x}) = \frac{1}{\sigma^2}\boldsymbol{x}^T(\boldsymbol{\mu}_j - \boldsymbol{\mu}_i) + \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \|\boldsymbol{\mu}_j\|_2^2\right)$. For a specific mixture component k, we use the variational distribution $\psi^{(k)}$. By linearity of expectation, the lower bound $\underline{L_i}(\boldsymbol{x}; \psi^{(k)}) = \mathbb{E}_{j\sim\psi^{(k)}}[L_{i,j}(\boldsymbol{x})] - \mathrm{KL}(\psi^{(k)} \,||\, \pi)$ from Lemma F.1 can be expressed as an affine transformation:

$$\underline{L_i}(\boldsymbol{x}; \psi^{(k)}) = \boldsymbol{x}^T \underbrace{\left[\frac{1}{\sigma^2}(\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i)\right]}_{\mathbf{a}_k} + \underbrace{\frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j\sim\psi^{(k)}}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi^{(k)} \,||\, \pi)}_{c_k}.$$

Conditioned on the latent component $z = k$, the vector $\boldsymbol{x}$ follows $\mathcal{N}(\boldsymbol{\rho}_k, \sigma^2\boldsymbol{I})$. Thus, the conditional distribution of the lower bound is univariate Gaussian with mean ν_k and variance ξ_k^2. The mean can

be calculated via:

$$\begin{aligned}\nu_k &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\rho}_k,\sigma^2\boldsymbol{I})}[\boldsymbol{x}^T\mathbf{a}_k + c_k] = \boldsymbol{\rho}_k^T\mathbf{a}_k + c_k \\ &= \frac{1}{\sigma^2}\boldsymbol{\rho}_k^T(\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i) + \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j\sim\psi^{(k)}}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi^{(k)} \mid\mid \pi).\end{aligned}$$

The variance ξ_k^2 can be calculated via:

$$\begin{aligned}\xi_k^2 &= \mathrm{Var}(\boldsymbol{x}^T\mathbf{a}_k + c_k \mid \boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\rho}_k,\sigma^2\boldsymbol{I})) = \mathbf{a}_k^T(\sigma^2\boldsymbol{I})\mathbf{a}_k \\ &= \sigma^2\left\|\frac{1}{\sigma^2}(\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i)\right\|_2^2 \\ &= \frac{1}{\sigma^2}\|\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i\|_2^2.\end{aligned}$$

Taking the square root yields $\xi_k = \frac{1}{\sigma}\|\boldsymbol{\mu}_i - \mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j]\|_2$. Since the mixture component k is chosen uniformly at random with probability $1/K$, the marginal distribution is the mixture $\frac{1}{K}\sum_{k=1}^K\mathcal{N}(\nu_k,\xi_k^2)$. □

Applying this result to the "remove" case, we have:

Theorem F.3. *Let $\boldsymbol{x}\sim\tilde{R}$ with $\tilde{R} = \frac{1}{b}\sum_{k=1}^b\mathcal{N}(\boldsymbol{\mu}_k,\sigma^2\boldsymbol{I})$. Let $\Psi = \{\psi^{(1)},\ldots,\psi^{(b)}\}$ be a collection of variational distributions on $[i-1]$. For any threshold $\tau\in\mathbb{R}$, the left-tail probability of the privacy loss random variable $L_i(\boldsymbol{x})$ with L_i defined in Lemma 5.3 is bounded by:*

$$\Pr_{\boldsymbol{x}\sim\tilde{R}}[L_i(\boldsymbol{x})\le\tau] \le \frac{1}{b}\sum_{k=1}^b\Phi\left(\frac{\tau-\nu_k}{\xi_k}\right), \tag{31}$$

where Φ is the cumulative distribution function of the standard normal distribution, and per-component parameters are:

$$\begin{aligned}\nu_k &= \frac{1}{\sigma^2}\boldsymbol{\mu}_k^T(\mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i) + \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j\sim\psi^{(k)}}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi^{(k)} \mid\mid \pi), \\ \xi_k &= \frac{1}{\sigma}\|\boldsymbol{\mu}_i - \mathbb{E}_{j\sim\psi^{(k)}}[\boldsymbol{\mu}_j]\|_2.\end{aligned}$$

Proof. By the Law of Total Probability, $\Pr_{\boldsymbol{x}\sim\tilde{R}}[L_i(\boldsymbol{x})\le\tau] = \frac{1}{b}\sum_{k=1}^b\Pr_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}_k,\sigma^2\boldsymbol{I})}[L_i(\boldsymbol{x})\le\tau]$. Since $L_i(\boldsymbol{x})\ge\underline{L_i}(\boldsymbol{x};\psi^{(k)})$, we have the bound $\Pr[L_i(\boldsymbol{x})\le\tau]\le\Pr[\underline{L_i}(\boldsymbol{x};\psi^{(k)})\le\tau]$ for each component k. Applying Lemma F.2 with $\boldsymbol{\rho}_k = \boldsymbol{\mu}_k$, the variable $\underline{L_i}(\boldsymbol{x};\psi^{(k)})$ conditioned on component k follows $\mathcal{N}(\nu_k,\xi_k)$. Consequently, each term in the sum is exactly $\Phi\left(\frac{\tau-\nu_k}{\xi_k}\right)$. □

For the "add" case, we observe that the lower bound on our privacy loss is a single, univariate Gaussian:

Lemma F.4. *Let $\boldsymbol{x}\sim\tilde{R}$ with $\tilde{R} = \mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})$. Then $\underline{L_i}(\boldsymbol{x})\sim\mathcal{N}(\nu,\xi)$ with mean and standard deviation*

$$\begin{aligned}\nu &= \left(||\boldsymbol{\mu}_i||_2^2 - \mathbb{E}_{j\sim\psi}[||\boldsymbol{\mu}_j||_2^2]\right) / (2\sigma^2) - \mathrm{KL}(\psi||\pi), \\ \xi &= ||\boldsymbol{\mu}_i - \mathbb{E}_{j\sim\psi}[\boldsymbol{\mu}_j]||_2 / \sigma.\end{aligned}$$

Proof. This immediately follows as a special case of Lemma F.2 with $\boldsymbol{\rho}_k = \mathbf{0}$ and $K = 1$. □

Naturally, this result directly implies the following analytical tail bound for the "add" case:

Theorem F.5. *Let $\boldsymbol{x}\sim\tilde{R}$ with $\tilde{R} = \mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})$. Let ψ be an arbitrary variational distribution on $[i-1]$. For any threshold $\tau\in\mathbb{R}$, the left-tail probability of the privacy loss random variable $L_i(\boldsymbol{x})$ with L_i defined in Lemma 5.3 is bounded by:*

$$\Pr_{\boldsymbol{x}\sim\tilde{R}}[L_i(\boldsymbol{x})\le\tau] \le \Phi\left(\frac{\tau-\nu}{\xi}\right), \tag{32}$$

where Φ *is the cumulative distribution function of the standard normal distribution, with parameters*

$$\nu = \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j\sim\psi}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi \mid\mid \pi),$$
$$\xi = \frac{1}{\sigma}\|\boldsymbol{\mu}_i - \mathbb{E}_{j\sim\psi}[\boldsymbol{\mu}_j]\|_2.$$

Remark F.6 (Handling Identical Components). In structured matrix mechanisms, multiple batches often share the same mixture mean, i.e., $\boldsymbol{\mu}_j = \boldsymbol{\mu}_i$. For these components, the privacy loss $L_{i,j}(\boldsymbol{x})$ is identically zero for all $\boldsymbol{x}$. Including them in the variational mean ν is suboptimal as it pulls the lower bound toward zero. We handle this by setting the variational weights $\psi_j = 0$ for all such indices. This effectively pulls the zero-loss components out of the expectation and into the KL divergence term, where they contribute a constant offset to the mean ν.

G Evaluation of variational bounds

In the following, we conduct an initial exploration of whether and how the variational privacy loss tail bounds from Appendix F may improve upon the basic AM-GM tail bounds from Appendix E. In particular, we consider different natural variational families and empirically compare them in terms of tightness of their reverse hazard bounds and bounds on the final privacy profile after conditional composition.

Recall that we need to find a lower tail bound on the ternary privacy loss random variable $L_{\tilde{P},\tilde{Q},\tilde{R}}$, which is defined as $L_i(\boldsymbol{x})$ with $L_i(\boldsymbol{x}) = \log\left(\frac{\mathrm{d}\tilde{P}}{\mathrm{d}\tilde{Q}}\right)(\boldsymbol{x})$ and $\boldsymbol{x} \sim \tilde{R}$, where $\tilde{P}$ is a uniform Gaussian mixture with means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1}$, $\tilde{Q}$ is a single Gaussian with mean $\boldsymbol{\mu}_i$, and $\tilde{R}$ is either:

- $\sum_{k=i}^{b} \mathcal{N}(\boldsymbol{\mu}_k, \sigma^2 \boldsymbol{I}_{n-1}) \cdot \frac{1}{b}$ ("remove" case),
- or $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_{n-1})$ ("add" case), which is equivalent to the trivial mixture $\sum_{k=i}^{b} \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_{n-1}) \cdot \frac{1}{b}$.

Let $\boldsymbol{\rho}_k \in \mathbb{R}^{n-1}$ denote an arbitrary component mean of $\tilde{R}$ (either $\boldsymbol{\mu}_k$ or the all-zeros vector $\mathbf{0}$). From Lemma F.2 and Theorems F.3 and F.5, we know that the tail probability conditioned on component k is upper-bounded via

$$\Pr_{\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\rho}_k, \sigma^2 \boldsymbol{I})}[L_i(\boldsymbol{x}) \leq \tau] \leq \Phi\left(\frac{\tau - \nu_k}{\xi_k}\right) \tag{33}$$

with scalar mean $\nu_k \in \mathbb{R}$ and scalar standard deviation $\xi_k \in \mathbb{R}_+$ with

$$\nu_k = \frac{1}{\sigma^2} \boldsymbol{\mu}_k^T (\mathbb{E}_{j \sim \psi^{(k)}}[\boldsymbol{\mu}_j] - \boldsymbol{\mu}_i) + \frac{1}{2\sigma^2}\left(\|\boldsymbol{\mu}_i\|_2^2 - \mathbb{E}_{j \sim \psi^{(k)}}[\|\boldsymbol{\mu}_j\|_2^2]\right) - \mathrm{KL}(\psi^{(k)} \mid\mid \pi), \tag{34}$$

$$\xi_k = \frac{1}{\sigma} \|\boldsymbol{\mu}_i - \mathbb{E}_{j \sim \psi^{(k)}}[\boldsymbol{\mu}_j]\|_2, \tag{35}$$

$$\pi = \mathrm{Uniform}(\{1, \ldots, i-1\}), \tag{36}$$

for any variational distribution $\psi^{(k)}$ over mixture indices $\{1, \ldots, i-1\}$. Therefore, we can define an entire variational family $\Psi^{(k)}$ and choose a $\psi^{(k)} \in \Psi^{(k)}$ that leads to the tightest tail bound.

In principle, we could let $\Psi^{(k)}$ be the set of all distributions supported on $\{1, \ldots, i-1\}$ and explicitly optimize $\psi^{(k)}$ (e.g., via gradient descent, which would be an interesting direction for future work). However, for $k \in \mathbb{N}$ epochs and $b \in \mathbb{N}$ batches per epoch and thus $N = k \cdot b$, we would need to solve $N \cdot b^2$ such optimization problems (one for each step n, each privacy loss L_i with $i \in [b]$, and each component $k \in [b]$ of $\tilde{R}$). We therefore consider the following heuristically chosen variational families instead:

G.1 Option 1: Maximizing the expectation based on KL divergence

A heuristic for tightening the bound in Equation (33) is by maximizing the mean ν_k. In turn, a heuristic for doing so is to eliminate the KL-divergence term in Equation (34) by letting

$$\psi^{(k)} = \pi = \mathrm{Uniform}(\{1, \ldots, i-1\}.$$

This corresponds to taking the geometric mean of privacy losses, which we already used in deriving our Rényi divergence bound from Lemma 4.4.

G.2 Option 2: Maximizing the expectation based on component distance

Another heuristic for maximizing the mean is to apply quadratic expansion, i.e.,

$$\nu_k = -\frac{1}{2\sigma^2} \mathbb{E}_{j \sim \psi^{(k)}}\left[||\boldsymbol{\rho}_k - \boldsymbol{\mu}_j||_2^2\right] + \frac{||\boldsymbol{\rho}_k - \boldsymbol{\mu}_i||_2^2}{2\sigma^2} - \mathrm{KL}(\psi^{(k)} || \pi),$$

and notice that the second term does not depend on variational distribution $\psi^{(k)}$. Instead of minimizing the KL-divergence, we could instead define $\psi^{(k)}$ to minimize the distance in the first

term, i.e., $\psi^{(k)}(z = j) = 1 \iff j = \text{argmin}||\rho_k - \boldsymbol{\mu}_j||_2^2$. To smoothly interpolate between this solution and minimization of the KL divergence, we can define the variational family as $\Psi^{(k)} = \{\psi(t, \boldsymbol{\rho}_k, (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1}\}) \mid t \in T\}$ with

$$\psi(t, \boldsymbol{\rho}_k, (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1})\} = \text{softmax}\left(\left\{-\frac{||\boldsymbol{\rho}_k - \boldsymbol{\mu}_j||_2^2}{t}\middle| 1 \leq j < i\right\}\right)$$

for some fixed set of non-negative temperatures $T \subset \mathbb{R}_+$.

G.3 Option 3: Minimizing the standard deviation based on component distance

Another heuristic for tightening the tail bound in Equation (33) is by minimizing the standard deviation ν_k, which makes the argument of Gaussian CDF Φ smaller for any $\tau \leq \nu_k$, i.e., any tail probability smaller than 0.5. Analogous to the previous approach, we can define $\Psi^{(k)} = \{\psi(t, \boldsymbol{\mu}_i, (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1})\} \mid t \in T\}$ with

$$\psi(t, \boldsymbol{\mu}_i, (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{i-1})\} = \text{softmax}\left(\left\{-\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2^2}{t}\middle| 1 \leq j < i\right\}\right)$$

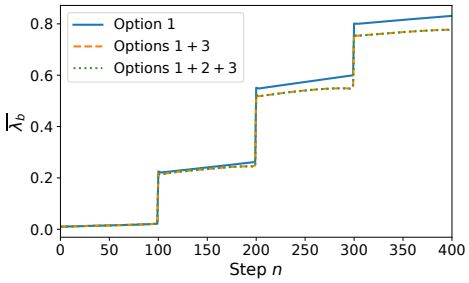
for some fixed set of non-negative temperatures $T \subset \mathbb{R}_+$. Notice that, unlike the previous approach, this variational family is independent of the component mean $\boldsymbol{\rho}_k$, i.e., we use the same variational family across all components of $\tilde{R}$.

G.4 Our choice.

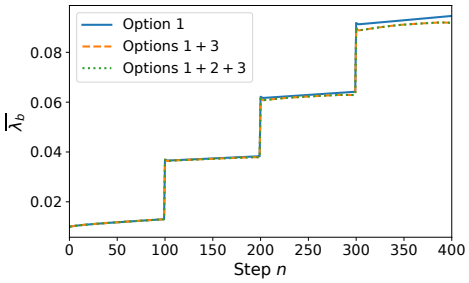
In our experiments, we opt for Option 3, since it only requires evaluation of a linear number of distances $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||$, and this computational cost is shared across all K components of our tail bound. We additionally include $\tau = \infty$, which corresponds to the KL-minimizing uniform distribution from Option 1. For our experiments, we specifically use temperatures $T = \{\infty\} \cup \{10^{-1}, 10^{-0.5}, 1, 10^{0.5}, 10^{1}\}$, which we found to improve tightness of our accountant compared to Option 1 ($T = \{\infty\}$) alone, i.e., the geometric mean. Furthermore, we empirically did not find Option 2 to offer any increased tightness. Figures 12 and 13 demonstrates this through an empirical comparison across different matrix factorizations and noise levels in the multi-epoch setting, visualizing both the reverse hazard function and the resultant privacy profiles. Table 4 additionally shows the effect on calibrated noise multipliers σ, where the variational family results in slightly reduced noise and the relative improvement grows with privacy budget ε.

Table 4: Effect of using the proposed variational family with temperature range $T \in \{\infty\} \cup \text{np.logspace}(-2, 2, 9)$ instead of only using the geometric mean (AM-GM) for calibrating noise multipliers σ under privacy budget $\delta = 10^{-5}$ and varying ε.

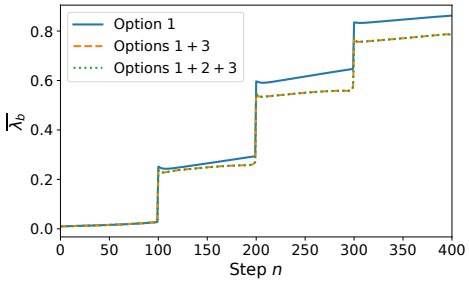
Mechanism	Bound	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 4$	$\varepsilon = 8$	$\varepsilon = 16$
BSR(4)	AM-GM	9.286	7.026	5.526	4.463	3.655	2.991
BSR(4)	VI(-2,2,9)	9.271	7.009	5.507	4.440	3.625	2.960
BISR(4)	AM-GM	11.764	8.650	6.611	5.183	4.113	3.267
BISR(4)	VI(-2,2,9)	11.732	8.614	6.575	5.141	4.057	3.208



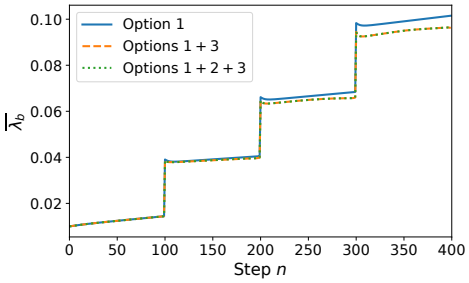
(a) BSR, $N = 400,\ k = 4,\ p = 4,\ \sigma = 2$



(b) BSR, $N = 400,\ k = 4,\ p = 4,\ \sigma = 5$

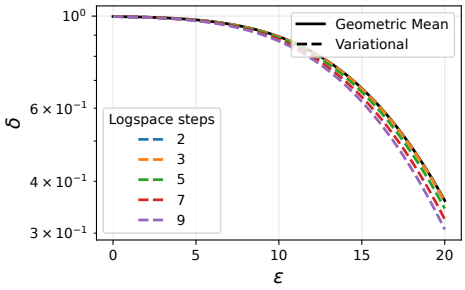


(c) BISR, $N = 400,\ k = 4,\ p = 4,\ \sigma = 2$

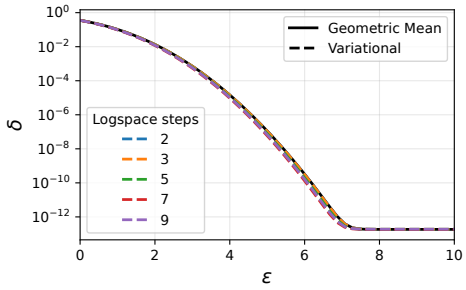


(d) BISR, $N = 400,\ k = 4,\ p = 4,\ \sigma = 5$

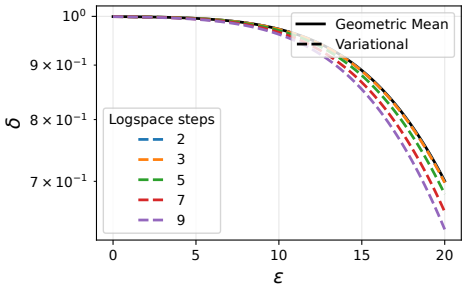
Figure 12: Comparison of the largest mixture component's posterior probability $\overline{\lambda_b}$ (smaller is better) between variational families for different factorizations and noise multipliers σ under $\delta_E = 10^{-5}$ and the "remove" relation. In later epochs, minimizing variance of the tail bound ("Options 1+3") improves upon only taking the geometric mean via a uniform distribution ("Option 1"). The effect is more pronounced in settings where the noise multiplier is small and sensitivity is large ($\sigma = 2$, BISR), compared to settings where the noise multiplier is large and the overall sensitivity is small ($\sigma = 5$, BSR). Maximizing the mean ("Option 2") does not further improve the posterior bound.



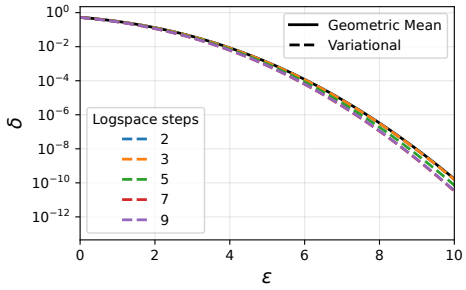
(a) BSR ($N = 100, k = 4, p = 4$), $\sigma = 2$



(b) BSR ($N = 100, k = 4, p = 4$), $\sigma = 4$



(c) BISR ($N = 100, k = 4, p = 4$), $\sigma = 2$



(d) BISR ($N = 100, k = 4, p = 4$), $\sigma = 2$

Figure 13: Effect of using the proposed variational family with temperature range $T \in \{\infty\} \cup \mathrm{np.logspace}(-2, 2, \mathrm{steps})$ instead of only using the geometric mean ($T = \infty$). Using multiple temperatures leads to an improvement in privacy parameter δ, especially for small noise multiplier σ / large privacy parameter ε.

G.5 Remark on Jumps in Reverse Hazard in Multi-Epoch Setting

At first, the jumps in our reverse hazard upper bound, which can be observed in Figure 12 may seem abnormal or like an implementation error. However, this is not the case. Instead, these jumps explain why we see a degradation in bound tightness in the multi-epoch setting, especially for DP-SGD.

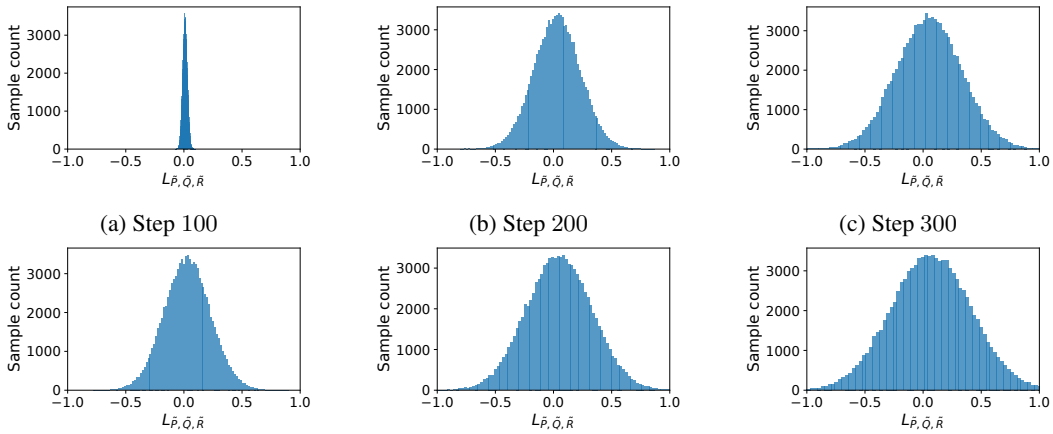
Consider DP-SGD with k epochs, b batches per epoch, and $N = k \cdot b$ iterations overall. From the definition of our dominating pair in Lemma 2.8, we know that the mixture means $\boldsymbol{m} \in \mathbb{R}^{k \cdot N}$ are simply

$$\boldsymbol{m} = \begin{bmatrix} \boldsymbol{I}_b \\ \boldsymbol{I}_b \\ \dots \\ \boldsymbol{I}_b \end{bmatrix},$$

i.e., a vertical stack of identity matrices, each with shape $b \times b$. Consider an arbitrary step $n \in [N]$. Recall from Algorithm 3 that, to find our bound $\overline{\lambda_b}$ on the reverse hazard of the largest mixture component, we derive a lower tail bound on the ternary privacy loss $L_{\tilde{P},\tilde{Q},\tilde{R}} = \log\left(\frac{\mathrm{d}\tilde{P}}{\tilde{Q}}\right)(\boldsymbol{x})$ with $\boldsymbol{x} \sim \tilde{R}$. Here, $\tilde{P}$ is a uniform mixture with means $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{b-1} \in \{0,1\}^{n-1}$ taken from $\boldsymbol{m}_{:,1:n-1}$. The distribution $\tilde{Q}$ is a single Gaussian with mean $\boldsymbol{\mu}_b \in \{0,1\}^{n-1}$. In epoch $e = (n \mod b) + 1$, the single mean $\boldsymbol{\mu}_b$ of $\tilde{Q}$ has $\max\{e-1, 0\}$ non-zero entries. Similarly, each mean $\boldsymbol{\mu}_j$ of $\tilde{P}$ has either e or $\max\{e-1, 0\}$ non-zero entries in distinct dimensions. Thus, the pairwise distance between $\boldsymbol{\mu}_i$ and any $\boldsymbol{\mu}_j$ fulfills

$$||\boldsymbol{\mu}_j - \boldsymbol{\mu}_i||_2 \in \{\sqrt{2e-1}, \sqrt{2e-2}\}.$$

Thus, the privacy loss increases from epoch to epoch. In Figure 14, we empirically verify this effect by taking 100 000 samples from $L_{\tilde{P},\tilde{Q},\tilde{R}}$ for DP-SGD at the end and start of each epoch for $k = 4$, $b = 100$, $\sigma = 5$. We observe that the sample histogram of the privacy loss distribution significantly widens when transitioning from one epoch to the next.



(a) Step 100 (b) Step 200 (c) Step 300

(d) Step 101 (e) Step 201 (f) Step 301

Figure 14: Sample histogram of the ternary privacy loss $L_{\tilde{P},\tilde{Q},\tilde{R}}$, whose lower tail is used to find reverse hazard upper bound $\overline{\lambda_b}$, shown at transitions between epochs. Samples are taken for DP-SGD with $k = 4$ epochs, $b = 100$ batches per epoch, and noise multiplier $\sigma = 5$ for the "remove" relation. The histogram widens between epochs, which explains observed jumps in the reverse hazard bound.

H Add and Remove Direction Dominating Pairs

Recall that we assume the following dominating pair:

Lemma H.1. *[Lemma 3.2 from [12]] Assume lower-triangular, non-negative strategy matrix $\boldsymbol{C}$, gradient clipping norm $\Delta = 1$, and balls-in-bins sampling with $k \in \mathbb{N}$ epochs and $b \in \mathbb{N}$ batches per epoch. Define distributions $P = \frac{1}{b}\sum_{i=1}^{b} \mathcal{N}(\boldsymbol{m}_i, \sigma^2 \boldsymbol{I}_N)$ and $Q = \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_N)$ supported on $\mathbb{R}^N$ with $N = k \cdot b$ and mixture means $\boldsymbol{m}_i = \sum_{j=0}^{k-1} |\mathbf{C}|_{:,\, i+jb}$, Then the subsampled matrix mechanism is dominated by (P, Q) under the "remove" [6] relation and by (Q, P) under the "add" relation.*

That is:

- In the "remove" case, P is a Gaussian mixture and Q is a single multivariate Gaussian.
- In the "add" case P is a single Gaussian and Q is a Gaussian mixture.

In the original Lemma 3.2 of [12], the directions are reversed. However, this appears to be a typing error. Their proof for the "add" case actually argues correctly that the first argument of the hockey-stick divergence should be a zero-mean Gaussian and the second argument should be a Gaussian mixture (note that in their proof $\boldsymbol{z}$ is a zero-mean Gaussian): "We give the proof for the add adjacency [...] Because each user is assigned to their batch independently, we can assume without loss of generality that contributions from all users other than the differing user are always 0. In more detail, by post-processing, we can assume that we release the contributions to the input matrix of all examples except the differing user's. Let $\boldsymbol{x}$ be these contributions, and $\boldsymbol{x}'$ be $\boldsymbol{x}$ plus the contribution of the differing user. Then distinguishing $\boldsymbol{C}\boldsymbol{x} + \boldsymbol{z}$ and $\boldsymbol{C}\boldsymbol{x}' + \boldsymbol{z}$ is equivalent to distinguishing $\boldsymbol{z}$ and $\boldsymbol{C}(\boldsymbol{x}' - \boldsymbol{x}) + \boldsymbol{z}$ [...]".

Further note that the same pattern of P being a mixture under "remove" and Q being a mixture under "add" arises in various other subsampling schemes, such as Poisson subsampling or subsampling without replacement (see, e.g., Theorem 11 in [39]).

[6] The original Lemma 3.2 swaps the "add" and "remove" direction, which appears to be a typing error, see Appendix H.

I Inapplicability of privacy loss distribution accounting

In the following, we explain in slightly more detail why existing privacy loss distribution (PLD) accountants cannot be applied to matrix mechanisms under random allocation. We first consider the single-epoch setting ($k = 1$), before discussing the multi-epoch setting ($k > 1$), which introduces further complications.

I.1 Single-epoch setting

In the single-epoch setting, the number of participation patterns b equals the number of steps N and each record participates to exactly one step. Thus, in Lemma 2.8 the number of mixture components $\mu_i = \mathcal{N}(\boldsymbol{m}_i, \sigma^2 \boldsymbol{I}_N)$ equals the number of steps N and the mixture means are indicator vectors $\boldsymbol{m}_i = \boldsymbol{e}_i$. Note that, in this section, subscripts i of μ denote mixture components, whereas subscripts n of P and Q denote steps.

The *approach* underlying PLD accounting is that, once we know the distribution of privacy loss random variable $L = \log\left(\frac{\mathrm{d}P}{\mathrm{d}Q}(\boldsymbol{y})\right)$ with $\mathbf{y} \sim P$ for dominating pair (P, Q), the privacy profile $\delta(\varepsilon) = H_{e^\varepsilon}(P||Q)$ can be evaluated via a simple univariate integral.

PLD accounting as an *efficient algorithmic procedure* becomes feasible when PLRV $L_{P,Q}$ can be expressed as a simple function of "smaller" PLRVs $L_1, \ldots, L_M$ whose distribution is known and easy to evaluate.

The standard use case [36, 25, 39] is that dominating pairs for N-fold adaptive composition factorize as $P = \bigtimes_{n=1}^{N} P_n$ and $Q = \bigtimes_{n=1}^{N} Q_n$, which implies $L = \sum_{n=1}^{N} L_{P_n,Q_n}$. However, as we already explained in Section 3, our distribution P is not factorizable for random allocation, even with uncorrelated noise ($\boldsymbol{C} = \boldsymbol{I}_N$). This is because P is a mixture distribution $P = \frac{1}{b}\sum_{i=1}^{b} \mu_i$.

For uncorrelated noise, Feldman & Shenfeld [21] show that one can nevertheless break down $L_{P,Q}$ into N simple PLRVs (see their Theorem 4.4). Specifically, they exploit that each μ_i only differs from Q in a single dimension, specifically $\mu_i = \bigtimes_{n=1}^{i-1} S_n \times R \times \bigtimes_{n=i+1}^{N} S_n$ and $Q = \bigtimes_{n=1}^{N} S_n$, where R and all S_n are univariate Gaussians. Thus, the privacy loss simplifies to

$$\log\left(\frac{\mathrm{d}P}{\mathrm{d}Q}(\boldsymbol{y})\right) = \log\left(\frac{1}{b}\sum_{i=1}^{b}\frac{\mathrm{d}\mu_i}{\mathrm{d}Q}(\boldsymbol{y})\right) = \log\left(\frac{1}{N}\sum_{n=1}^{N}\exp\left(\log\left(\frac{\mathrm{d}R}{\mathrm{d}S}(y_n)\right)\right)\right).$$

This is a relatively simple analytical function of N per-step privacy losses. These N privacy losses become independent PLRVs when sampling $\mathbf{y} \sim \mu_i$ conditioned on any mixture component i.

In contrast, with matrix mechanisms, each component μ_i has a different mean $\boldsymbol{m}_i$ (recall Lemma 2.8) with multiple non-zero entries. Thus, the per-component likelihood ratios $\frac{\mathrm{d}\mu_i}{\mathrm{d}Q}(\boldsymbol{y})$ are functions of $\boldsymbol{m}_i^T\boldsymbol{y}$. Thus, and because the means are generally not orthogonal, each summand of $\log\left(\frac{1}{b}\sum_{i=1}^{b}\frac{\mathrm{d}\mu_i}{\mathrm{d}Q}(\boldsymbol{y})\right)$ depends on multiple elements of random vector $\boldsymbol{y}$, which breaks the independence assumption.

To conclude: While random allocation breaks decomposition into per-step privacy losses, noise correlation further breaks decomposition into per-component privacy losses.

I.2 Multi-epoch setting

Going beyond $k = 1$ poses further challenges that make the PLD accountant of Feldman & Shenfeld [21] inapplicable. First, it should be noted that there are actually two variants of balls-and-bins sampling or random allocation: The first one [13, 12], which we use in Definition 2.7, repeats the same participation pattern in each of the k epochs. We adopted this definition for consistency with prior work on matrix mechanism accounting by Choquette-Choo et al. [12]. They likely chose it because the number of mixture components is linear in the number of batches per epoch b, which facilitates Monte Carlo accounting. In contrast, Feldman & Shenfeld [20, 21] and Dong et al. [15] let each record choose k steps to participate in uniformly at random, resulting in a combinatorally growing number of possible participation patterns.

However, even if one were to apply matrix mechanisms to their notion of random allocation, the proof strategy from Feldman & Shenfeld [21] would break down. Concretely, their first step is a reduction from $k > 1$ to $k = 1$ (see Lemma 3.2 in Feldman & Shenfeld [20] and Lemma 2.8 in Feldman & Shenfeld [21]). This reduction relies on randomly splitting the steps $\{1, \dots, N\}$ into $N \,/\, k$ subsets and applying the subsampled mechanism independently to each subset. However, this is fundamentally incompatible with matrix mechanisms: They are specifically designed to map a fixed number of inputs N (e.g., batches) to a fixed number of outputs N (e.g., batch gradients) by sampling from a joint distribution with optimized strategy matrix $\boldsymbol{C} \in \mathbb{R}^{N \times N}$. A similar reduction could potentially be possible if one were to impose a block-diagonal structure on $\boldsymbol{C}$. However, this is not standard practice.

J Broader impact

Our work is primarily targeted at mitigating negative societal impact of machine learning by providing formal privacy protection. However, it shares various potential risks, including risks for misuse, with other works. Firstly, differential privacy is an inherently probabilistic notion of privacy, but may be misrepresented as providing qualitatively identical results to deterministic cryptographic methods (e.g., encryption). Furthermore, differentially private machine learning in practice is usually conducted in settings with $\delta < |S|^{-1}$ but $\varepsilon > 1$ [35]. Here, differential privacy allows for relatively comparison of privacy between models, but does not provide strong privacy protection in absolute sense [35]. Nevertheless, the claim that a model was trained with some form of differential privacy may be used to incorrectly privacy-wash publicly released methods. This is compounded by the fact that numerical privacy parameters (ε, δ) are not easy to interpret by non-experts. Another issue is that, even if an algorithm is formally proven to fulfill certain privacy guarantees, it's practical implementations may violate them due to implementation errors or numerical issues. This may affect matrix mechanisms in particular, since they add additional complication compared to uncorrelated noise (DP-SGD) model training. A potential way to mitigate this risk is through privacy auditing methods (see [33] for an overview). But these may themselves be affected by implementation issues.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The claim that our methods are sampling-free is inherent to their design. The claim that our Rényi accountant uses dynamic programming for efficient computation is supported by Algorithm 4 and the runtime discussion in Section 4. The claim that conditional composition complements it by offering stronger guarantees in the low-ε regime is supported by experimental results in Section 6 and Appendix A. The claim that we support arbitrary banded and non-banded matrices is inherent to the design of our algorithms. The claim that we conduct experiments across a wide range of matrix mechanisms is supported by experiments in Section 6 and Appendix A, where we consider DP-SGD, BSR, BISR, BLT, BandMF, and BandInvMF factorizations. Our claim that we improve upon the computational complexity of the best known Rényi accountant for random allocation and uncorrelated noise is formally proven as part of Lemma 4.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See "limitations" paragraph at end of Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We formally prove all claims while explicitly stating all involved assumptions in Appendices D to F.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The used algorithms are clearly described in Algorithms 1 to 3 and 4. The involved parameters for our experiments are specified in Appendix B. Code was shared via the supplementary material during the review period, and will be made publicly available at github.com/morganstanley/MSML/tree/main/papers/Matrix_Mechanism_Accounting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code was shared via the supplementary material during the review period, and will be made publicly available at github.com/morganstanley/MSML/tree/main/papers/Matrix_Mechanism_Accounting. The experimental setup for our main experimental results is fully specified in Appendix B. Further details for reproducing ancillary results, in which we vary various parameters, are provided with the respective experiments in Appendix A. The only used data is the CIFAR-10 dataset, which is publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Appendix B which specifies all experimental parameters. Further details for reproducing ancillary results, in which we vary various parameters, are provided with the respective experiments in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our main contribution is that the proposed algorithms are sampling-free, i.e., there is no randomness involved. For the Monte Carlo baseline, we show raw Monte Carlo estimates in the main experimental figure, but we additionally calculate high probability confidence intervals in Appendix A.2. For the CIFAR model training experiment in Appendix A.3, we show standard deviations across random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the worker type in Appendix B. We did not track compute used across the entire project, but measure wall-clock runtime for our algorithms across a wide range of parameter choices in Appendix A.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We read the ethics guideline and agreed to them on OpenReview.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Yes (see Appendix J, which we reference from Section 7). However, our method is primarily intended to mitigate negative societal impact by providing formal privacy guarantees. There are potentials for misuse, but these are shared with any other work on differentially private machine learning.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any new data or models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, we cite the authors/owners in Appendix B and specify the respective licenses in Appendix B.8.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We did not conduct studies with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We did not conduct studies with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLMS as an important, original, or non-standard component of the core methods.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.