# BETTER BATCHES, BETTER PRIVACY: RANDOM ALLOCATION FOR DP-BANDMF

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The current state of the art for batch selection and noise addition in differentially private training is a combination of DP-BandMF (informally, DP-SGD where the noise added in two different steps can be correlated if the steps are within $b$ of each other) and privacy amplification by sampling via Monte Carlo accounting. First, whereas prior batch selection strategies for DP-BandMF were inspired by Poisson sampling, random allocation is known to outperform Poisson sampling for DP-SGD. We therefore propose and analyze new batch selection strategies for DP-BandMF inspired by random allocation, improving RMSE by up to $1.85\times$ over previous work. Second, Monte Carlo accounting typically requires drawing $\Omega(1/\delta)$ samples from a privacy loss distribution, making accounting costly. We increase sampling and scoring throughput by two orders of magnitude through a carefully optimized GPU implementation. Finally, previous work chose the DP-BandMF strategy matrix and the batch selection strategy independently. We give a simple framework for choosing these parameters jointly, reducing RMSE by up to $13.4\%$ over fixed strategy matrices.

## 1 INTRODUCTION

Differential privacy (DP) (Dwork et al., 2006) guarantees that an algorithm's output on two datasets $D, D'$ differing in one example satisfy some indistinguishability guarantee. The de facto DP ML algorithm is differentially private stochastic gradient descent (`DP-SGD`) (Abadi et al., 2016), which clips each per-example gradient to a maximum norm, and adds Gaussian noise to their sum.

A key technique for `DP-SGD`'s success is *privacy amplification by sampling* (Kasiviswanathan et al., 2008). Privacy amplification by sampling is the observation that by randomizing batches in `DP-SGD`, the problem of guessing whether $D$ or $D'$ was used becomes strictly harder, allowing us to reduce the noise while retaining the same privacy guarantee. For most of the time since the popularization of `DP-SGD` by the seminal work of (Abadi et al., 2016), the default method for randomizing batches (which we refer to as a *batch selection strategy*) has been *Poisson sampling*: each example in the dataset is included in each batch independently with probability $p$.

Recent implementations of data processing pipelines (Chua et al., 2024a; Ritter et al., 2023) allow for random access to data elements and arbitrary batch sizes, making almost any batch selection strategy readily implementable. In turn, choosing batch selection strategies which admit lower noise multipliers is a low-barrier, high-yield path to improving practical DP training. In light of this, several recent works have proposed alternatives to Poisson sampling, improving on it in two directions:

- **Controlling participation variance (Shenfeld and Feldman, 2026; Dong et al., 2025; Dong and Özgür, 2026)**: A downside of Poisson sampling is that an example can participate many times, meaning in some extreme settings Poisson sampling is actually worse than deterministic batching for privacy (Chua et al., 2024b). A natural correction is *random allocation*[1]*:* fix the number of participations, and only randomize their position.
- **Matrix factorization (Choquette-Choo et al., 2025; Dong and Ganesh, 2026)**: `DP-SGD` adds independent noise in each step. `DP-MF` is a generalization that allows one to add noise that is

[1]As named by Shenfeld and Feldman (2026); it is also called balanced iteration subsampling by the concurrent work of Dong et al. (2025).



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

correlated in different steps. DP-MF breaks a standard assumption in privacy analysis that each step's randomness is independent. Choquette-Choo et al. (2025) bypassed this assumption by analyzing privacy amplification by sampling for DP-MF via Monte Carlo accounting, where a sufficient primitive is to compute the likelihood ratio for an output of the algorithm on adjacent datasets. For Poisson sampling this primitive is inefficient, but several works have proposed alternative batch selection strategies for which this primitive is efficient.

While for DP-SGD Dong and Özgür (2026) provides evidence that random allocation is near-optimal for DP-SGD, for DP-MF there is still a rich yet largely unexplored space of batch selection strategies. As DP-MF can substantially improve utility in practice compared to DP-SGD (Xu et al., 2023), identifying the best combination of privacy amplification and DP-MF is an important practical question. In this paper we attempt to enumerate and compare some of these strategies.

**Contributions** We connect several ideas presented in recent prior work: (1) random allocation for DP-SGD (Shenfeld and Feldman, 2026; Dong et al., 2025), (2) Monte Carlo accounting for DP-MF (Choquette-Choo et al., 2023; Dong and Ganesh, 2026), and (3) parametric DP-MF optimization under Monte Carlo accounting (Kalinin et al., 2026; McKenna, 2025).

- **Random Allocation for DP-BandMF:** We first attempt to identify the best batch selection strategy for DP-BandMF, a subclass of DP-MF that is particularly amenable to privacy amplification. We introduce three new randomized batch selection strategies for DP-BandMF that guarantee exact per-example participation counts while enforcing a useful "$b$-separation" property in Section 3.2. We derive exact recurrence relations for their likelihood ratios to enable Monte Carlo accounting in Section 3.3. Across all evaluated settings our proposed strategies consistently improve utility, achieving a median $1.54\times$ (up to $1.85\times$) improvement in RMSE (the standard utility metric for DP-MF) over the baseline, and a median $1.31\times$ (up to $1.36\times$) improvement over the prior state of the art.
- **Efficient Monte Carlo Accounting:** We next aim to make Monte Carlo accounting more efficient. We show that the structure of our batch selection strategies and Toeplitz matrices make Monte Carlo accounting particularly well-suited to GPU acceleration in Section 3.4. Through a carefully-optimized JAX implementation, we demonstrate throughput exceeding $10^6$ privacy loss samples per second on an H100 GPU in Fig. 1b, representing a $50\times$ to $100\times$ speedup over existing CPU baselines and making the technique more practical and accessible.
- **Strategy Optimization under Monte Carlo Accounting:** The above contributions assume the DP-BandMF strategy matrix is fixed in advance. We propose a simple and efficient method to optimize over a single-parameter class of strategy matrices under Monte Carlo accounting, and demonstrate that this can reduce RMSE over fixed strategies by up to $13.4\%$.

## 2 Background

### 2.1 Differential Privacy, DP-SGD, and DP-MF

Differential privacy (Dwork et al., 2006) is the gold standard for data privacy. Given a data universe $\mathcal{D}$, we say two datasets $D, D' \in \mathcal{D}^m$ are adjacent, denoted $D \simeq D'$, if $D'$ is the same as $D$ except with one example removed or vice-versa[2]. Given two distributions $P, Q$ over an output space $\mathcal{Y}$, the $\alpha$-hockey stick divergence is $H_\alpha(P, Q) := \mathbb{E}_{y\sim P}\left[\max\left\{1 - \alpha\frac{Q(y)}{P(y)}, 0\right\}\right]$. A mechanism $\mathcal{M} : \mathcal{D}^* \to \mathcal{Y}$ is $(\varepsilon, \delta)$-DP if for all $D \simeq D'$, we have $H_{e^\varepsilon}(\mathcal{M}(D), \mathcal{M}(D')) \leq \delta$.

DP-SGD (Abadi et al., 2016) is the standard approach for DP machine learning. In iteration $i \in [n]$, letting $\mathsf{clip}(\theta, C) := \theta \cdot \frac{C}{\max\{\|\theta\|_2, C\}}$, given batch $B_i$, clipping norm $C$, and noise multiplier $\sigma$, we compute the noisy gradient $\tilde{g}_i = \sum_{d\in B_i} \mathsf{clip}(\nabla\ell(\theta_{i-1}; d), C) + \mathbf{z}_i$, where $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 C^2 \mathbf{I})$. The noisy gradient can then be used to perform a model update (from $\theta_{i-1}$ to $\theta_i$) by any first-order optimizer. Without loss of generality, we assume $C = 1$ throughout the paper.

DP-MF (Denisov et al., 2022) is a generalization of DP-SGD which uses noise correlated across steps. For simplicity we present it in the one-dimensional optimization setting. Let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_T)$.

[2]This is the *add-remove adjacency*. All results in the paper also hold for the *zero-out adjacency*.

DP-MF is parameterized by a (lower-triangular) matrix $\mathbf{C} \in \mathbb{R}^{T \times T}$. DP-MF adds noise $(\mathbf{C}^{-1}\mathbf{z})_i$ in iteration $i$, whereas DP-SGD would add noise $\mathbf{z}_i$. The setting $\mathbf{C} = \mathbf{I}$ retrieves DP-SGD. DP-MF has been observed to achieve substantial improvements in utility over DP-SGD Denisov et al. (2022) including in production training runs Xu et al. (2023).

The main variant of DP-MF we are interested in is ($b$-)DP-BandMF (Choquette-Choo et al., 2023), in which $\mathbf{C}$ is $b$-banded, i.e. $\mathbf{C}[i, j] = 0$ when $i \notin [j, j + b - 1]$. A line of works (Choquette-Choo et al., 2023; 2025; Dong and Ganesh, 2026) have observed DP-BandMF is particularly amenable to privacy amplification by sampling. For convenience we also throughout the paper assume Toeplitz $\mathbf{C}$, i.e. those with constant diagonals, equivalently $\mathbf{C}_{i,j} = \mathbf{C}_{i+k,j+k}$ for all $i, j, k$.

### 2.2 Privacy Analysis and Monte Carlo Accounting

Privacy amplification by sampling (Kasiviswanathan et al., 2008) is the observation that randomizing the batches used in DP-SGD and DP-MF can improve the privacy-utility tradeoff "for free," i.e. without modifying training parameters like the number of iterations or (average) batch size. Hence, the problem of analyzing the privacy guarantees of DP-SGD or DP-MF with randomized batch selection strategies is fundamental for optimizing utility in practice. Lemma 4.5 of Choquette-Choo et al. (2024b), restated below for convenience, simplifies this task:

**Lemma 1.** *Suppose we run $T$-step DP-MF using a batch selection strategy such that each example participates independently from each other. For a given example, let $\mathbf{x} \in \mathbb{R}^T$ be the random vector where $\mathbf{x}_i = 1$ if that example is included in batch $B_i$ and $0$ otherwise, and we assume $\mathbf{x}$ is distributed identically for all examples. Then DP-MF using this batch selection strategy is $(\varepsilon, \delta)$-DP iff $H_{e^\varepsilon}(P, Q) \leq \delta$ and $H_{e^\varepsilon}(Q, P) \leq \delta$, where $P = \mathbf{C}\mathbf{x} + \mathbf{z}$ and $Q = \mathbf{z}$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_T)$.*

Analyzing $H_{e^\varepsilon}(P, Q)$ is challenging because $P$, $Q$ are high-dimensional distributions and $P$ does not necessarily have a simple structure. This is in contrast to e.g. DP-SGD with Poisson sampling, where the coordinates of $P$ are independent. This enables techniques like composition theorems which reduce to easy-to-analyze one-dimensional distributions. There are two predominant techniques in the literature for handling privacy amplification of DP-MF:

- Choquette-Choo et al. (2023) propose "cyclic" batch selection strategies for DP-BandMF. Each cyclic strategy splits the dataset (arbitrarily) into $b$ subsets. The $i$th subset is only eligible to participate in steps $j$ where $j \equiv i \pmod b$. Within these steps, they use another "inner" batch selection strategy such as Poisson sampling to subsample from the eligible examples. They prove the privacy guarantees reduce to those of $T/b$-step DP-SGD using the inner batch selection strategy.
- Monte Carlo accounting (Wang et al., 2023) uses Monte Carlo estimation of $H_{e^\varepsilon}(P, Q)$. In order to perform Monte Carlo accounting, it suffices to have primitives for (i) sampling $y \sim P$, (ii) evaluating $P(y)/Q(y)$ for any given $y$. (i) is straightforward since sampling $\mathbf{x}$ is equivalent to implementing the batch selection strategy, but (ii) can be more challenging since $\mathbf{x}$ can take on up to $2^T$ values. Dong and Ganesh (2026) gave a procedure which uses Monte Carlo accounting, and allows one to run end-to-end accounting and training with formal $(\varepsilon, \delta)$-DP, including calibration of the $\sigma$ parameter. We restate their procedure and its guarantees in Appendix A for convenience. Notably, Monte Carlo accounting typically requires $> 1/\delta$ samples, making it much more expensive than standard accounting. While this adds significant overhead, Dong and Ganesh (2026) demonstrated that Monte Carlo accounting is still feasible at production scales.

### 2.3 Prior Batch Selection Strategies

We briefly review the batch selection strategies from prior work that are most relevant to our work:

**Poisson sampling (DP-SGD):** Each example is included in each batch independently with probability $p$, yielding an expected participation count of $\mathbb{E}[N] = pT$ (Abadi et al., 2016). While analytically convenient, an example's total participation count follows a Binomial distribution whose heavy right tail can degrade privacy (Chua et al., 2024b).

**Random Allocation (DP-SGD):** Each example independently picks a uniformly random subset of $K$ out of the $T$ batches to be included in. This batch selection strategy was proposed in several

concurrent works (Shenfeld and Feldman, 2026; Dong et al., 2025). Dong and Özgür (2026) gave a principled theoretical explanation and strong empirical evidence for why DP-SGD with random allocation seems to strictly improve on DP-SGD with Poisson sampling.

**Deterministic (DP-BandMF):** Each example participates at fixed, evenly spaced iterations with gap $T/K \geq b$ (we call the property of having size $\geq b$ gaps $b$-*separation*). Because noise in DP-BandMF is correlated across windows of length $b$, $b$-separation ensures that an example's participations do not overlap in correlated noise, enabling a simpler privacy analysis.

**Balls-In-Bins (DP-BandMF):** Each example participates at evenly spaced iterations again, but the step of the first participation is randomized. Choquette-Choo et al. (2025) showed this is amenable to Monte Carlo accounting as $\mathbf{x}$ only takes on a few values.

**Cyclic Poisson (DP-BandMF):** Each example is assigned to one of $b$ subsets $i \in [b]$ arbitrarily and participates independently with probability $p$ only in iterations $t \equiv i \pmod b$ (Choquette-Choo et al., 2023). This provides $b$-separation, but still suffers from participation variance.

**Min Sep Poisson (DP-BandMF):** The same as Poisson sampling, but rejecting any example that participated in the previous $b-1$ iterations (Dong and Ganesh, 2026). This maintains $b$-separation while achieving better utility than cyclic Poisson, but again suffers from participation variance.

As summarized in Table 2 in Appendix C, each existing strategy satisfies at most two of three desirable properties: fixed participation counts, $b$-separation, and maximum benefit from amplification. We formalize these design principles in Section 3.1.

## 3 Batch Selection Strategies for DP-BandMF

In this section, we propose and analyze a family of batch selection strategies for DP-BandMF based on random allocation. Below, we formalize the design principles, present our proposed batch selection strategies, derive their associated recurrence relations, and provide practical implementation details. For simplicity, we assume the number of bands $b$ evenly divides the number of iterations $T$ throughout the paper. We also assume $\mathbf{C}$ is $b$-banded Toeplitz, and the first column of $\mathbf{C}$ has norm 1.

### 3.1 Design Principles

Throughout this work we attempt to answer several fundamental questions about the combination of DP-BandMF and privacy amplification. We start with arguably the most immediate one: *What is the batch selection strategy which maximally reduces the noise used by DP-BandMF?* Below we enumerate a set of desiderata that reasonable batch selection strategies should satisfy.

- **Fixed-K**: Each example should participate a fixed number of times $K$. As Dong and Özgür (2026) demonstrate, this is the "first-order" predictor of the privacy-utility tradeoff.
- **$b$-Separation:** Each participation from the same example should be separated by $\geq b$ steps. As observed in Choquette-Choo et al. (2023; 2025), this simplifies privacy analysis for DP-BandMF.
- **Entropy**: All else equal, assignments of examples to iterations that are "more random" are generally preferred. As Dong and Özgür (2026) observed, more randomization helps, but is a "second-order" predictor of the privacy utility tradeoff compared to fixed-$K$.
- **Structure:** The following properties greatly simplify privacy analysis: (1) The batch assignment should be independent across examples (as required by Lemma 1), and (2) the sampling should have a simple recursive structure to make Monte Carlo accounting tractable.

### 3.2 Proposed Strategies

To address the desiderata outlined in Section 3.1, we propose three new batch selection strategies for DP-BandMF that generalize random allocation (Shenfeld and Feldman, 2026; Dong et al., 2025). Below, we formalize each strategy from the perspective of a single example's participation schedule.

**Cyclic Random Allocation (`Cyclic RA`).** Our first batch selection strategy, *cyclic random allocation* (`Cyclic RA`), is an instantiation of the cyclic framework of Choquette-Choo et al. (2023), replacing the inner Poisson sampling with random allocation. Specifically, the dataset is partitioned into $b$ subsets (arbitrarily, but consistently across neighboring datasets $D \simeq D'$), and each example in subset $i \in [b]$ is eligible to participate only in rounds $t \equiv i \pmod b$. Within these eligible rounds, each example selects exactly $K$ rounds uniformly at random. Just as random allocation improves upon Poisson sampling for `DP-SGD`, we expect `Cyclic RA` to improve upon `Cyclic Poisson`. As we show in Section 3.3, `Cyclic RA` inherits the reduction of Choquette-Choo et al. (2023), allowing near-exact privacy analysis via standard privacy loss distributions (PLD).

**Nested Random Allocation (`Nested RA`).** A limitation of `Cyclic Poisson` and `Cyclic RA` is that the choice of modulus $i \in [b]$ for each example is arbitrary, when it could be randomized for better privacy. *Nested random allocation* (`Nested RA`) randomizes the modulus by assigning each example uniformly at random to one of the $b$ subsets. It is otherwise identical to `Cyclic RA`. A formal specification of `Nested RA` is given in Algorithm 3 in Appendix C, written as a sampler for a set of steps for a single example. Because participations under different moduli can affect the same output coordinates, the reduction of Choquette-Choo et al. (2023) cannot account for this added randomness; we instead analyze `Nested RA` via Monte Carlo accounting in Section 3.3.

**Min Sep Random Allocation (`Min Sep RA`).** Just as `Min Sep Poisson` improves upon `Cyclic Poisson` by enforcing $b$-separation only on *actual* participations rather than *potential* participations, we do the same but for random allocation. In *min sep random allocation* (`Min Sep RA`), each example independently selects a uniformly random subset of $K$ rounds from $[T]$ satisfying the $b$-separation property ($|s_i - s_j| \geq b$ for all $i \neq j$). A formalization of `Min Sep RA` is given in Algorithm 4 in Appendix C. The following proposition enables an efficient implementation:

**Proposition 2** (Schedule Bijection)**.** *Let $T' = T - (K-1)(b-1)$. The map $\phi(s_1, \ldots, s_K) = (s_1, s_2 + (b-1), \ldots, s_K + (K-1)(b-1))$ for sorted $s_1 < \cdots < s_K \in [T']$ is a bijection from $\binom{[T']}{K}$ to the set of valid $b$-separated subsets of $[T]$ of size $K$.*

We analyze the privacy of `Min Sep RA` via Monte Carlo accounting in Section 3.3. While `Min Sep RA` has higher entropy than e.g. `Nested RA`, it has one notable drawback:

**Remark 3** (Expected Batch Sizes)**.** *For `Min Sep RA`, the expected batch size varies across steps: near the beginning and end of training, the expected batch size is larger. Similar to the warm-start correction used for `Min Sep Poisson` (Dong and Ganesh, 2026), there is a natural correction to the batch selection strategy which fixes this behavior. See Appendix F for more details.*

### 3.3 Privacy Analyses

We now provide formal statements for privacy analysis of our proposed batch selection strategies.

**Cyclic Random Allocation** Theorem 5 of Choquette-Choo et al. (2023) gives the privacy analysis:

**Theorem 4.** *$b$-`DP-BandMF` using $K$-out-of-$T$ cyclic random allocation satisfies any privacy guarantee satisfied by `DP-SGD` using $K$-out-of-$T/b$ random allocation.*

The privacy guarantees for `DP-SGD` with random allocation can be computed using the results of Feldman and Shenfeld (2026); Dong and Özgür (2026).

**Nested Random Allocation** The analysis of Choquette-Choo et al. (2023) assumes the modulus is public and doesn't benefit directly from randomization of it. Instead of using their analysis, we provide the necessary primitives for Monte Carlo accounting: (i) being able to sample $\mathbf{C}\mathbf{x} + \mathbf{z}$ efficiently, where $\mathbf{x}, \mathbf{z}$ are defined as in Lemma 1, and (ii) being able to evaluate $P(\mathbf{y})/Q(\mathbf{y})$ for an arbitrary vector $\mathbf{y}$, where $P = \mathbf{C}\mathbf{x} + \mathbf{z}$ and $Q = \mathbf{z}$. (i) is straightforward, so we focus on (ii).

We cannot evaluate $P/Q$ via naive summation as there are $b \cdot \binom{T/b}{K}$ possible values for $\mathbf{x}$. We instead follow the framework of Dong and Ganesh (2026): we determine a recursive structure that allows us to evaluate $P/Q$ via a dynamic program. Let $\mathbf{c} = \mathbf{C}[1:b, 1]$, and let $\ell_t := \frac{\mathcal{N}(\mathbf{c}, \sigma^2\mathbf{I})(\mathbf{y}[t:t+b-1])}{\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})(\mathbf{y}[t:t+b-1])}$[3].

[3]We abuse notation here for readability; if $t > T - b + 1$, instead we are looking at the last $T - t + 1 < b$ coordinates of $\mathbf{y}$, and the first $T - t + 1$ coordinates of $\mathbf{c}$.

That is, $\ell_t$ is the likelihood ratio of $\mathbf{y}[t : t+b-1]$, assuming participation in step $t$. Let $\mathrm{LR}(k,t)$ denote the likelihood ratio of the length $T-t+1$ suffix $\mathbf{y}[t:T]$, conditioned on (i) the example not having participated in iterations $t-b+1$ to $t-1$, (ii) the modulus of the example being $\equiv t \pmod b$, and (iii) the example participates $k$ times in iterations $t$ to $T$. Our base cases are $\mathrm{LR}(k,t)=1$ if either $t>T$ or $k=0$. Letting $p_{k,t} := \frac{k}{\lceil (T+1-t)/b \rceil}$ our recurrence is:

$$\mathrm{LR}(k,t) = p_{k,t} \cdot \ell_t \cdot \mathrm{LR}(k-1, t+b) + (1-p_{k,t}) \cdot \mathrm{LR}(k, t+b). \tag{1}$$

**Theorem 5.** *Using Eq.* (1), $\frac{1}{b}\sum_{i=1}^{b} \mathrm{LR}(K,i)$ *is equal to* $P(\mathbf{y})/Q(\mathbf{y})$ *for b-*`DP-BandMF` *with* `Nested RA` *and can be computed in time* $O(T(K+\log T))$.[4]

The proof is given in Appendix D. Intuitively, the first (resp. second) term captures the probability of participating (resp. not participating) in step $t$, the likelihood ratio $\ell_t$ (resp. 1) of $\mathbf{y}[t:t+b-1]$, and the likelihood ratio of the remaining suffix with $k-1$ (resp. $k$) remaining participations.

**Min Sep Random Allocation** Due to structural similarities between `Min Sep RA` and `Nested RA`'s recursive structure, Monte Carlo accounting for `Min Sep RA` is analogous to Monte Carlo accounting for `Nested RA`. Define $\mathbf{c}, \ell_t$ as in the setup for Eq. (1). Let $\mathrm{LR}(k,t)$ be the likelihood ratio for $\mathbf{y}[t:T]$, conditioned on $k$ participations happening in iterations $t$ to $T$, and the example not participating in iterations $t-b+1,\ldots,t-1$. We let $p_{k,t} := \frac{k}{T-t+1-(k-1)(b-1)}$. We have, similarly to Eq. (1), the base case $\mathrm{LR}(k,t)=1$ if $k=0$ or $t>T$ and:

$$\mathrm{LR}(k,t) = p_{k,t} \cdot \ell_t \cdot \mathrm{LR}(k-1, t+b) + (1-p_{k,t}) \cdot \mathrm{LR}(k, t+1). \tag{2}$$

**Theorem 6.** *Using Eq.* (2)*,* $LR(K,1)$ *equals* $P(\mathbf{y})/Q(\mathbf{y})$ *for b-*`DP-BandMF` *with* `Min Sep RA` *and can be computed in time* $O(T(K+\log T))$.

The proof is again given in Appendix D. The intuition is analogous to Eq. (1). Besides the definition of $p_{k,t}$ the only difference is that the second term includes $\mathrm{LR}(k,t+1)$ instead of $\mathrm{LR}(k,t+b)$, as we no longer restrict participation based on an example's sampled modulus. The first term still uses $\mathrm{LR}(k-1,t+b)$ due to enforcement of the $b$-separation constraint.

### 3.4 Implementation Details

**Algorithm 1** Monte Carlo Sampling and Scoring (Wang et al., 2023; Dong and Ganesh, 2026)

**Require:** $b$-banded strategy matrix $\mathbf{C} \in \mathbb{R}^{T\times T}$, noise multiplier $\sigma$, batch distribution $\mathcal{D}_K$
1: Sample sparse participation indicator, $\mathbf{x} \sim \mathcal{D}_K, \mathbf{x} \in \{0,1\}^T$
2: Sample standard Gaussian noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_T)$
3: Form observations: $\mathbf{y}^+ = \mathbf{C}\mathbf{x} + \sigma\mathbf{z}$ and $\mathbf{y}^- = \sigma\mathbf{z}$
4: Compute scores: $\Lambda^+ \leftarrow \mathrm{ScoreDP}(\mathbf{y}^+, \mathbf{C}, \sigma, K)$ and $\Lambda^- \leftarrow \mathrm{ScoreDP}(\mathbf{y}^-, \mathbf{C}, \sigma, K)$
5: **return** $\{\Lambda^+, \Lambda^-\}$

A high-level algorithm for Monte Carlo Sampling and Scoring for `DP-BandMF` is shown in Algorithm 1. In our actual implementation, we exploit the structure of `DP-BandMF` and our proposed batch selection strategies to greatly improve efficiency over a baseline implementation:

- **Exploiting $b$-bandedness and $b$-separation** By leveraging the fact that $\mathbf{C}$ is a *$b$-banded Toeplitz* matrix and and $\mathcal{D}_K$ samples $b$-separated participations, we can efficiently compute $\mathbf{C}\mathbf{x}$ in linear time using simple array indexing tricks. Specifically, the $b$-separation guarantees that each participation's contribution is completely disjoint, so $\mathbf{C}\mathbf{x}$ is simply a zero vector with the $b$ Toeplitz coefficients $\mathbf{c} := \mathbf{C}[1:b,1]$ placed at each index $\{j : \mathbf{x}_j = 1\}$ without any collisions.
- **Exploiting Fixed-$K$ Structure** Since $|S| = K$ is static under our batch selection strategies, all intermediate variables in Algorithm 1 have static shapes, enabling JIT compilation and efficient GPU-based processing in JAX, which are more efficient than CPU-based numpy implementations.

[4] We remark that a "screening" technique was introduced by Dong and Özgür (2026), that makes the runtime per sample closer to $O(T\log T)$. A similar technique can be used to speed up Monte Carlo accounting using Theorem 5. Since the changes are analogous to the work of Dong and Özgür (2026), we omit the details here.

- **Batching and Accumulation** Monte Carlo accounting requires generating $\Omega(\frac{1}{\delta})$ samples for formal verification. To do this efficiently, we leverage `jax.vmap` and `jax.lax.scan` to vectorize the computation of many samples and accumulate them into a size $50K$ histogram indexed by privacy loss values linearly spaced out in the range $[-20, 30]$. To ensure the reported privacy guarantee remains formally valid, we round privacy loss values up to the nearest histogram bin value. This combination of higher-order functions allows us to fully saturate the hardware while keeping memory manageable even for very large sample sizes, all while keeping the primary sampling and scoring code specialized to a single sample.
- **Carefully Optimized Dynamic Program** Evaluating Eqs. (1) and (2) requires computing per-step Gaussian log-likelihood ratios $\ell_t$, which in turn requires computing $\langle \mathbf{c}, \mathbf{y}[t : t + b - 1] \rangle$ and $\langle \mathbf{c}, \mathbf{c} \rangle$, and sweeping a backward recurrence over $k \in \{0, \ldots, K\}$. We compute all $T$ inner products simultaneously in $O(T \log T)$ time using `jax.scipy.signal.fftconvolve`. Computing $\langle \mathbf{c}, \mathbf{c} \rangle$ is straightforward and fast. Finally, our recurrences at step $t$ only rely on future steps $t + 1$ and $t + b$. Rather than allocating a full $T \times (K + 1)$ table in device memory, we implement the backward sweep using a circular rolling buffer of size $b \times (K + 1)$.

Together, these optimizations enable us to generate $\approx 10^6$ privacy loss samples per second (Fig. 1b), a 50-100$\times$ improvement over prior work, making Monte Carlo accounting practical for smaller $\delta$.

## 4 Optimizing DP-BandMF with MC Accounting

Our goal is to optimize the DP-BandMF strategy matrix $\mathbf{C} \in \mathbb{R}^{T \times T}$ while accounting for amplification. The standard objective for DP-MF, as in Denisov et al. (2022), is the *expected total squared error* on a workload matrix $\mathbf{A}$ (typically the "prefix sum" all-ones lower-triangular matrix):

$$\text{TSE}(\mathbf{C}) \;=\; \|\mathbf{A}\mathbf{C}^{-1}\|_F^2 \cdot \sigma_{\varepsilon,\delta}^2(\mathbf{C}) \tag{3}$$

where $\|\mathbf{A}\mathbf{C}^{-1}\|_F$ computes the matrix mechanism's $\ell_2$-error for $\sigma = 1$, and $\sigma_{\varepsilon,\delta}(\mathbf{C})$ is the minimum noise multiplier required for the resulting DP-BandMF mechanism to satisfy $(\varepsilon, \delta)$-DP.

For cyclic strategies privacy depends on $\mathbf{C}$ only through its maximum column norm and bandwidth $b$. As a result, $\mathbf{C}$ can be optimized independently for different values of $b$, and $\sigma$ can be calculated efficiently using standard numerical methods. This decoupling does not apply for non-cyclic strategies analyzed under Monte Carlo accounting, where the privacy depends on the specific contents of $\mathbf{C}$. Hence, $b$ and $\mathbf{C}$ must be optimized jointly with $\sigma$.

Freely optimizing the $b$ parameters of banded Toeplitz $\mathbf{C}$ is non-trivial under Monte Carlo accounting. For simplicity, we adopt the following single-parameter family of Toeplitz strategies proposed in McKenna (2025): $\mathbf{C}[i, 1] \;\propto\; r + 1/i$, for $i = 1, \ldots, b$. In previous work, this parameterization was observed to be near-optimal for the prefix workload. While this near-optimality may not hold under Monte Carlo accounting, we adopt it as a convenient heuristic to reduce the search space.

The joint optimization over bandwidth and Toeplitz coefficients can be done as a discrete outer search over candidate bandwidths $b \;\leq T/K$, and a continuous 1D search over $r \in (-1/b, 1/2)$ for each candidate $b$. Because the objective is well-behaved and unimodal in $r$, we solve the 1D subproblem using standard scalar bounded optimization (Press et al., 1992). Evaluating the objective at $(b, r)$ requires estimating $\sigma_{\varepsilon,\delta}(\mathbf{C})$. Since only an estimate is required, we use a much smaller Monte Carlo sample budget ($N \approx \frac{1}{\delta}$) than what we need for *formal calibration*. This gives an approximately optimal candidate $(b^*, r^*)$, for which we perform a final formal calibration of $\sigma$ prior to training.

## 5 Experiments

We evaluate seven batch selection strategies: four established baselines `Deterministic` (`No-Amp`), `Cyclic Poisson`, `Balls-in-Bins`, and `Min Sep Poisson` and our `Cyclic RA`, `Nested RA`, and `Min Sep RA`. To ensure a fair comparison, all methods are normalized to $K$ expected (or exact) participations per user over $T$ iterations. We use a different banded Toeplitz strategy tailored for each batch selection strategy. For `Deterministic`, we optimize RMSE for $b = \frac{T}{K}$, and heuristically apply this same Toeplitz strategy to `Balls-in-Bins`. For `Cyclic Poisson` and `Cyclic RA`, we search $b \in \{1, 2, \ldots, \frac{T}{K}\}$ to find the optimal $(b^*, \mathbf{C}^*)$ minimizing

Table 1: Table of results showing the RMSE of each batch selection strategy normalized by `Cyclic Poisson` (values $< 1.00$ indicate improvement). The raw RMSE for `Cyclic Poisson` is shown in parenthesis for reference.

| | | **Prior Work** | | | | **Proposed (Ours)** | | |
|---|---|---|---|---|---|---|---|---|
| | | PLD | | Monte Carlo | | PLD | Monte Carlo (Std / Opt) | |
| | Param | `No-Amp` | `Cyclic Poisson` | `Balls in Bins` | `Min Sep Poisson` | `Cyclic RA` | `Nested RA` | `Min Sep RA` |
| Vary $T$ | 256 | 1.14 | 1.00 (1.32) | 0.85 | 0.91 | 0.88 | 0.76 / 0.72 | 0.75 / **0.68** |
| | 512 | 1.18 | 1.00 (1.38) | 0.85 | 0.90 | 0.89 | 0.75 / 0.71 | 0.73 / **0.65** |
| | 1024 | 1.21 | 1.00 (1.44) | 0.84 | 0.90 | 0.89 | 0.76 / 0.69 | 0.75 / **0.63** |
| | 2048 | 1.24 | 1.00 (1.50) | 0.83 | 0.90 | 0.89 | 0.74 / 0.67 | 0.73 / **0.61** |
| Vary $K$ | 1 | 1.05 | 1.00 (3.14) | 0.55[5] | 0.90 | 0.89 | 0.72 / **0.54** | 0.71 / **0.54** |
| | 2 | 1.16 | 1.00 (2.04) | 0.69 | 0.90 | 0.88 | 0.69 / 0.61 | 0.68 / **0.55** |
| | 4 | 1.21 | 1.00 (1.44) | 0.84 | 0.90 | 0.89 | 0.76 / 0.69 | 0.75 / **0.63** |
| | 8 | 1.21 | 1.00 (1.11) | 0.96 | 0.93 | 0.92 | 0.85 / 0.79 | 0.84 / **0.77** |
| | 16 | 1.17 | 1.00 (0.94) | 1.03 | 0.98 | 0.95 | 0.92 / 0.90 | 0.92 / **0.87** |
| Vary $\varepsilon$ | 1.0 | 1.75 | 1.00 (3.43) | 1.22 | 0.98 | 0.93 | 0.90 / 0.85 | 0.90 / **0.80** |
| | 2.0 | 1.48 | 1.00 (2.17) | 1.03 | 0.94 | 0.91 | 0.82 / 0.77 | 0.81 / **0.70** |
| | 4.0 | 1.21 | 1.00 (1.44) | 0.84 | 0.90 | 0.89 | 0.76 / 0.69 | 0.75 / **0.63** |
| | 8.0 | 1.00 | 1.00 (0.97) | 1.00 | 0.97 | 0.87 | 0.81 / 0.76 | 0.80 / **0.72** |

TSE (noting that $b^*$ can differ between the two), and heuristically adopt these strategies for `Min Sep Poisson` and the standard variants of `Nested RA` and `Min Sep RA`, respectively. We use the approach described in Section 4 for the optimized variants of these strategies. Details on accounting for the experiments are deferred to Appendix E.

**Evaluation** We compare the batch selection strategies, while varying number of iterations ($T$), number of participations per example ($K$) and privacy budgets ($\varepsilon$). We center the experiments around the *anchor* config $T = 1024$, $K = 4$, and $\varepsilon = 4$, and vary each parameter one at a time. We chose this config because it represents a standard practical setting for large-scale DP training, where single-digit privacy budgets are common and high-epoch training is impractical. Our primary evaluation metric is the (normalized) RMSE, defined as RMSE $= \frac{1}{K}\sqrt{\text{TSE}/T}$.

### 5.1 RMSE Experiments

Our main experimental results are shown in Table 1. Below we quantify improvements in terms of the *relative RMSE* between two methods, i.e., the ratio between two RMSEs.

- Among PLD-based methods (which are cheaper to calibrate), `Cyclic RA` consistently provides the best RMSE, beating `Cyclic Poisson` by between 5-14%, and about 11% on average.
- Prior to our work, the best batch selection strategy varied across settings. Our proposed `Min Sep RA` with optimized batch selection strategy consistently performs the best, followed by `Nested RA` with an optimized strategy. The magnitude of improvement over `Cyclic Poisson` ranges from 15% to 85%, with improvement growing with larger $T$ and smaller $K$.
- When using the standard strategy (optimized for `Cyclic RA`), the RMSE between `Min Sep RA` and `Nested RA` is consistently less than 3% in all settings we considered. When using the optimized strategies, the difference is up to 10% in some settings.
- The gain between `Cyclic RA` and `Nested RA`/ `Min Sep RA` using the same Toeplitz coefficients averages 12%. When optimizing the Toeplitz coefficients, the relative RMSE compared to the standard Toeplitz coefficients averages 8.5% for `Nested RA` and 13.4% for `Min Sep RA`.

In Fig. 1a, we show the RMSE as a function of the bandwidth $b$ under the anchor configuration.

[5] At ($K = 1, b = T$), `Balls-in-Bins`, `Nested RA`, `Min Sep RA` are equivalent. `Balls-in-Bins` ($b = T$) outperforms the non-optimized alternatives because they use $b = 64$ (optimal for `Cyclic RA`).

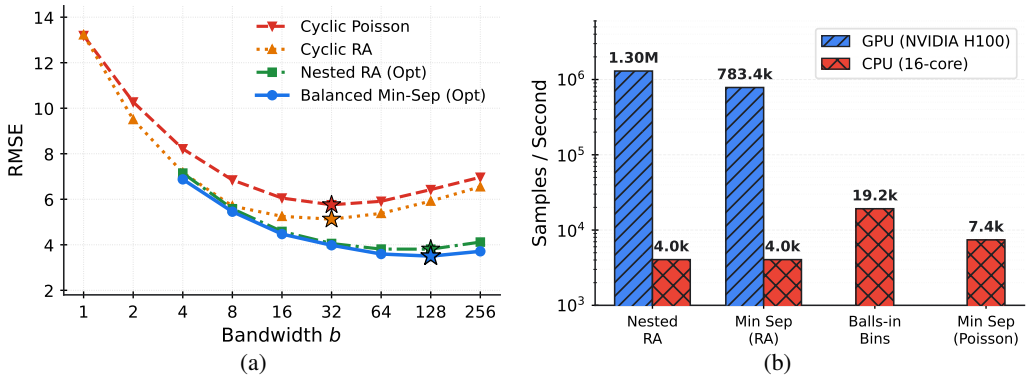


Figure 1: (a) RMSE as a function of number of bands $b$ and (b) Sampling throughput of Monte Carlo accounting on CPU and GPU. Both plots show results for $T = 1024$, $K = 4$, and $\varepsilon = 4.0$.

- While `Cyclic Poisson` and `Cyclic RA` achieve their optimal RMSE at $b = 32$, `Nested RA` and `Min Sep RA` reach their optimum at a $4\times$ larger bandwidth of $b = 128$.
- While `Cyclic Poisson` and `Cyclic RA` require delicate bandwidth tuning to balance privacy amplification against correlated noise, both `Nested RA` and `Min Sep RA` consistently benefit from larger bandwidths without having to make the same trade-off.

### 5.2 Sampling Throughput

In Fig. 1b we measure the throughput for Monte Carlo sampling for both our new batch selection strategies, and existing ones under the anchor config. For existing methods, we use the implementation available in `JAX Privacy`, and for our methods, we use the one described in Section 3.4.

- On CPU, our implementation generates about $4K$ samples per second, while the `JAX Privacy` implementations enjoy $19.2K$ and $7.4K$ samples per second, respectively. This difference is explained by the extra dimension $(K)$ needed in our dynamic programming state. At this throughput, generating 1 million samples takes $\approx 4$ minutes on a single CPU. Recalling that we need $\approx 10M$ samples for a single formal verification at $\delta = 10^{-5}$, and an additional sweep over up to $\approx 50$ different $\sigma$ values, this is feasible but expensive and may require parallelization.
- The throughput increases substantially when moving to an H100 GPU, up to $1.3M$ samples/second for `Nested RA` and $783K$ samples/second for `Min Sep RA`. This makes it feasible to do both strategy optimization and formal calibration on a single machine in a reasonable amount of time. This is enabled by our highly-optimized implementation described in Section 3.4.

## 6 Discussion and Future Work

In this work, we unified two previously separate lines of research in private training: random allocation for `DP-SGD` and privacy amplification for banded matrix factorization. By generalizing the $b$-separation and random allocation families into a cohesive framework, our proposed strategies consistently improve RMSE across all settings evaluated, achieving gains over prior PLD-based approaches as large as $1.85\times$. Furthermore, our carefully-optimized implementation makes Monte Carlo accounting orders of magnitude more efficient, transforming it into a more practical tool.

Our results point to several promising directions for further improvement. First, formal calibration currently draws $\Omega(1/\delta)$ samples for each candidate noise multiplier $\sigma$. Narrowing down the initial candidate $\sigma_0$ could reduce the number of evaluated values from roughly 50 to 2 or 3, providing a substantial speedup. Moreover, developing heuristics to directly predict the optimal $(b, \theta, \sigma_0)$ tuple from a given setting $(\varepsilon, \delta, T, K)$ would bypass our expensive $(b, r)$ grid search. Second, our Toeplitz strategy optimization is somewhat coarse: we consider only powers of 2 for bandwidth $b$ and restrict Toeplitz coefficients to a single-parameter family. Exploring other parameterizations of the strategy matrix could yield additional utility improvements.

## REFERENCES


Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, 2021.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and Rényi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2496–2506, 2020.

Zhiqi Bu, Jialin Mao, and Shiyun Xu. Scalable and efficient training of large convolutional neural networks with differential privacy. *Advances in Neural Information Processing Systems*, 35: 38305–38318, 2022.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In *Advances in Neural Information Processing Systems*, 2023.

Christopher Choquette-Choo, Krishnamurthy Dvijotham, Krishna Pillutla, Arun Ganesh, Thomas Steinke, and Abhradeep Guha Thakurta. Correlated noise provably beats independent noise for differentially private learning. In *International Conference on Learning Representations*, volume 2024, pages 4616–4676, 2024a.

Christopher Choquette-Choo, Arun Ganesh, Thomas Steinke, and Abhradeep Guha Thakurta. Privacy amplification for matrix mechanisms. In *International Conference on Learning Representations*, volume 2024, pages 38418–38447, 2024b.

Christopher A. Choquette-Choo, H. Brendan McMahan, Keith Rush, and Abhradeep Guha Thakurta. Multi-epoch matrix factorization mechanisms for private machine learning. *arXiv preprint arXiv:2211.06530*, 2022.

Christopher A. Choquette-Choo, Arun Ganesh, Ryan McKenna, H. Brendan McMahan, Keith Rush, Abhradeep Guha Thakurta, and Zheng Xu. (Amplified) banded matrix factorization: A unified approach to private training. In *Advances in Neural Information Processing Systems*, 2023.

Christopher A. Choquette-Choo, Arun Ganesh, Saminul Haque, Thomas Steinke, and Abhradeep Guha Thakurta. Near exact privacy amplification for matrix mechanisms. In *International Conference on Learning Representations*, 2025.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Scalable DP-SGD: Shuffling vs. poisson subsampling. In *Advances in Neural Information Processing Systems*, 2024a.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How private are DP-SGD implementations? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8904–8918. PMLR, 21–27 Jul 2024b. URL `https://proceedings.mlr.press/v235/chua24a.html`.

Sergey Denisov, H. Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In *Advances in Neural Information Processing Systems*, 2022.

Andy Dong and Arun Ganesh. Privacy amplification for BandMF via $b$-min-sep subsampling. *arXiv preprint arXiv:2602.09338*, 2026.

Andy Dong and Ayfer Özgür. Less random, more private: What is the optimal subsampling scheme for DP-SGD? *arXiv preprint arXiv:2605.07072*, 2026.

Andy Dong, Wei-Ning Chen, and Ayfer Ozgur. Leveraging randomness in model and data partitioning for privacy amplification. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 13938–13962. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/dong25a.html.

Krishnamurthy Dj Dvijotham, H Brendan McMahan, Krishna Pillutla, Thomas Steinke, and Abhradeep Thakurta. Efficient and near-optimal noise generation for streaming differential privacy. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2306–2317. IEEE, 2024.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284, 2006.

Vitaly Feldman and Moshe Shenfeld. Efficient privacy loss accounting for subsampling and random allocation, 2026. URL https://arxiv.org/abs/2602.17284.

Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *IEEE Symposium on Foundations of Computer Science*, 2018.

Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained error bound on differentially private continual observation. In *International Conference on Machine Learning*, pages 10072–10092, 2023.

Arun Ganesh, Ryan Mckenna, Hugh Brendan Mcmahan, Adam Smith, and Fan Wu. It's my data too: Private ML for datasets with multi-user training examples. 267:18189–18205, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/ganesh25a.html.

Arun Ganesh, Brendan McMahan, and Abhradeep Thakurta. On design principles for private adaptive optimizers. *arXiv preprint arXiv:2507.01129*, 2025b.

Atefeh Gilani, Naima Tasnim, Lalitha Sankar, and Oliver Kosut. Geoclip: Geometry-aware clipping for differentially private sgd. *Advances in Neural Information Processing Systems*, 38:6233–6257, 2026.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems*, 2021.

Xin Gu, Yingtai Xiao, Guanlin He, Jiamu Bai, Daniel Kifer, and Kiwan Maeng. Correlating cross-iteration noise for dp-sgd using model curvature. *arXiv preprint arXiv:2510.05416*, 2025.

Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. Exploring the limits of differentially private deep learning with group-wise clipping. *arXiv preprint arXiv:2212.01539*, 2022.

Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. Almost tight error bounds on differentially private continual counting. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5003–5039, 2023.

Peter Kairouz, H. Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, 2021.

Nikita P. Kalinin, Ryan McKenna, Rasmus Pagh, and Christoph H. Lampert. DP-$\lambda$CGD: Efficient noise correlation for differentially private model training. *arXiv preprint arXiv:2601.22334*, 2026.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008. doi: 10.1109/FOCS.2008.27.

Anastasia Koloskova, Ryan McKenna, Zachary Charles, Keith Rush, and H. Brendan McMahan. Gradient descent with linearly correlated noise: Theory and applications to differential privacy. In *Advances in Neural Information Processing Systems*, 2023.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: Optimizing linear counting queries under differential privacy. *The VLDB Journal*, 24 (6):757–781, 2015.

Tian Li, Manzil Zaheer, Ken Ziyu Liu, Sashank J. Reddi, H. B. McMahan, and Virginia Smith. Differentially private adaptive optimization with delayed preconditioners. *ArXiv*, abs/2212.00309, 2022. URL https://api.semanticscholar.org/CorpusID:254125312.

Ryan McKenna. Scaling up the banded matrix factorization mechanism for differentially private ML. In *International Conference on Learning Representations*, 2025. Spotlight.

Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Proc. VLDB Endow.*, 11:1206–1219, 2018. URL https://api.semanticscholar.org/CorpusID:51869463.

Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium*, pages 263–275, 2017.

Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X. Yu, Sashank J. Reddi, and Sanjiv Kumar. AdaCliP: Adaptive clipping for private SGD. *arXiv preprint arXiv:1908.07643*, 2019.

William H Press, Saul A Teukolsky, William T Vetterling, Brian P Flannery, et al. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1992.

Marvin Ritter, Ihor Indyk, Aayush Singh, Andrew Audibert, Anoosha Seelam, Camelia Hanes, Eric Lau, Jacek Olesiak, Jiyang Kang, and Xihui Wu. Grain - feeding jax models, 2023. URL http://github.com/google/grain.

Sebastián Rodríguez Beltrán, Marlon Tobaben, Joonas Jälkö, Niki Loppi, and Antti Honkela. Efficient and scalable implementation of differentially private deep learning without shortcuts. *arXiv preprint arXiv:2406.17298*, 2024.

Jan Schuchardt and Nikita Kalinin. Sampling-free privacy accounting for matrix mechanisms under random allocation. *arXiv preprint arXiv:2601.21636*, 2026.

Moshe Shenfeld and Vitaly Feldman. Privacy amplification by random allocation. *Advances in Neural Information Processing Systems*, 38:96188–96230, 2026.

Jiachen T. Wang, Saeed Mahloujifar, Tong Wu, Ruoxi Jia, and Prateek Mittal. A randomized approach to tight privacy accounting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher Choquette, Peter Kairouz, Brendan Mcmahan, Jesse Rosenstock, and Yuanbo Zhang. Federated learning of gboard language models with differential privacy. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 629–639, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.60. URL https://aclanthology.org/2023.acl-industry.60/.

## Ethics Statement

This work studies algorithmic batch selection and formal privacy accounting for differentially private machine learning using purely analytical evaluations; it does not involve human subjects or sensitive datasets, and raises no ethical concerns or negative societal risks.

## REPRODUCIBILITY STATEMENT

We believe our work is fully reproducible, as all critical details have been included in the paper or appendix. In addition, source code for this work will be made available upon request, which should make it very easy to reproduce any value we report or claim we make in Section 5. Specifically, any cell in Table 1 can be reproduced by running a self-contained binary with inputs $T, K, \varepsilon$ (for PLD approaches and "Std" `Nested RA` and `Min Sep RA`). To reproduce the "Opt" results, the same binary can be used with a custom $(b, r)$ input, with optimized values provided in Tables 4 and 5 for reference.

## AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and proposing the random rotation construction (Appendix F) to resolve the biased batch size issue. We have not used generative AI tools for experimental design, developing our core theoretical framework, or writing the primary proofs, and dataset-related and translation tasks are not applicable to this work. Additionally, we used generative AI tools for writing code, managing code health, generating figures and tables, and editing the manuscript to improve readability. We have reviewed all AI-assisted work, and all LLM-generated code and experimental outputs were verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## A MONTE CARLO BACKGROUND

Below we restate Algorithm 5 of Dong and Ganesh (2026), which allows for formal calibration of a parameter such as the noise multiplier with end-to-end DP guarantees using Monte Carlo accounting. In the context of `DP-MF`, "dominating pair" means the pair $P, Q$ given by Lemma 1. In our use-case, each $\mathcal{M}_i$ is a run of `DP-BandMF` with a different noise multiplier, and we have $\mathcal{M}_1, \mathcal{M}_2, \ldots$ correspond to a decreasing sequence of noise multipliers.

**Algorithm 2** Monte Carlo Calibration

**Require:** Candidate mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots \mathcal{M}_k$. Target privacy parameters $(\varepsilon, \delta)$, "base" privacy parameter $\delta_{base} \in (0, \delta)$, input dataset $D$.

$N \leftarrow$ minimum number of samples such that any mechanism not satisfying $(\varepsilon, \delta_{base})$-DP passes Monte Carlo verification of $(\varepsilon, \delta)$-DP w.p. at most $\frac{\delta - \delta_{base}}{1 - \delta_{base}}$ (computable using open-source `jax_privacy` utilities).
$i = 1$
**while** $i \leq k$ **do**
  $S^+ \leftarrow N$ samples from PLD for dominating pair $(P_i, Q_i)$ for $\mathcal{M}_i$
  $S^- \leftarrow N$ samples from PLD for $(Q_i, P_i)$
  **if** Monte Carlo estimate from $S^+$ (resp. $S^-$) of $H_{e^\epsilon}(P_i, Q_i)$ (resp. $H_{e^\epsilon}(Q_i, P_i)$)$> \delta_{base}$ **then**
    **break**
  **end if**
  $i = i + 1$
**end while**
**if** $i = 1$ **then**
  **return** $\perp$ (calibration failed)
**end if**
**return** $\mathcal{M}_{i-1}(D)$

**Theorem 7** (Theorem A.1 of Dong and Ganesh (2026))**.** *Algorithm 2 (calibration via Monte Carlo accounting and the final run/release of the inner mechanism $\mathcal{M}_{i-1}$) satisfies $(\epsilon, \delta)$-DP.*

### A.1 Sample Complexity of Monte Carlo Accounting

In Fig. 2 we plot the number of samples per candidate $\sigma$ to do formal Monte Carlo calibration at different target $\delta$ values. For $\delta = 10^{-5}$, we needed $10,745,967$ samples per candidate for formal calibration, which we rounded up to $11,000,000$ for simplicity in our experiments.

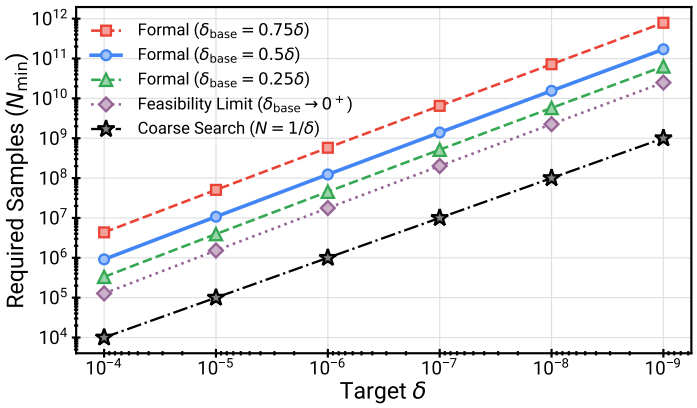


Figure 2: The number of samples needed for Monte Carlo accounting. The blue line ($\delta_{base} = 0.5\delta$) is the default setting for formal calibration recommended by prior work and used in this paper for formal calibration.

## B Related Work

Our work operates at the intersection of privacy accounting, batch selection, and noise addition, which are three key building blocks for `DP-SGD`-style mechanisms (along with gradient clipping).

**Privacy accounting.** Tight privacy accounting provides utility gains without altering the training dynamics. Research in this area has progressed from RDP-based analyses (Abadi et al., 2016; Mironov, 2017; Balle et al., 2018; Feldman et al., 2018; Balle et al., 2020) to tight numerical composition via PLD (Koskela et al., 2020; Gopi et al., 2021). As batching and noise mechanisms have grown more complex, accounting has shifted toward tight mechanism-specific analyses (Shenfeld and Feldman, 2026; Feldman and Shenfeld, 2026; Schuchardt and Kalinin, 2026; Dong and Ganesh, 2026) and Monte Carlo accounting (Wang et al., 2023; Choquette-Choo et al., 2025).

**Batch selection.** Privacy analyses have traditionally assumed independent `Poisson` subsampling (Abadi et al., 2016; Balle et al., 2018) to simplify the math, while in practice training pipelines typically rely on fixed-size batches and sequential data access, creating a well-documented mismatch between theoretical accounting assumptions and actual implementations (Rodríguez Beltrán et al., 2024; Chua et al., 2024a;b). Beyond the selected batch selection strategies highlighted in the main paper, other ones that have been proposed in practice include shuffling, sampling with replacement, and sampling without replacement Balle et al. (2018). For cross-device federated learning settings where random access to individual users or records is infeasible, minimum separation (without any subsampling) is another batch selection strategy that has been proposed and analyzed in prior work (Kairouz et al., 2021; Choquette-Choo et al., 2023; Ganesh et al., 2025a).

**Noise addition.** Rather than injecting independent Gaussian noise at each step as in standard `DP-SGD`, `DP-MF` adds noise that is correlated across iterations to reduce the error on the *prefix sums* of gradients at the cost of increased error on individual gradients (Choquette-Choo et al., 2024a; Denisov et al., 2022; Choquette-Choo et al., 2022). Modern approaches to this are based on the *matrix mechanism*, originally developed for answering linear query workloads with DP (Li et al., 2015; McKenna et al., 2018). Work in this space includes advancing SOTA in RMSE (Choquette-Choo et al., 2023; 2024b; Dong and Ganesh, 2026), improving memory efficiency without sacrificing too much utility (McKenna, 2025; Dvijotham et al., 2024; Kalinin et al., 2026), optimizing strategy

matrices for realistic training scenarios rather than proxies based on RMSE (Koloskova et al., 2023; Gu et al., 2025), and theoretical analyses (Fichtenberger et al., 2023; Henzinger et al., 2023).

**Gradient clipping.** An orthogonal line of work improves per-example gradient clipping, a key primitive for `DP-SGD` and `DP-MF`. Key innovations include adaptive and quantile-based clipping thresholds (Pichapati et al., 2019; Andrew et al., 2021), computational methods such as ghost clipping and per-layer clipping (Bu et al., 2022; 2023; He et al., 2022), and geometry-aware clipping strategies (Li et al., 2022; Ganesh et al., 2025b; Gilani et al., 2026).

## C Deferred Figures

Table 2: Taxonomy of batch selection strategies.

| Strategy | Mechanism | Params | Fixed $K$ | $b$-Sep | Entropy | Accounting |
|---|---|---|---|---|---|---|
| `Poisson` | `DP-SGD` | $p$ | × | × | ✓✓ | `PLD` |
| `Random Allocation` | `DP-SGD` | $K$ | ✓ | × | ✓✓ | `PLD` |
| `Deterministic` | `DP-MF` + `DP-BandMF` | $K$ | ✓ | ✓ | × | `PLD` |
| `Cyclic Poisson` | `DP-BandMF` | $b, p$ | × | ✓ | ✓ | `PLD` |
| `Balls-in-Bins` | `DP-MF` + `DP-BandMF` | $b, p$ | ✓ | ✓ | ✓ | `PLD` |
| `Min Sep Poisson` | `DP-BandMF` | $b, p$ | × | ✓ | ✓✓ | `MC` |
| `Cyclic RA` | `DP-BandMF` | $b, K$ | ✓ | ✓ | ✓ | `PLD` |
| `Nested RA` | `DP-BandMF` | $b, K$ | ✓ | ✓ | ✓✓ | `MC` |
| `Min Sep RA` | `DP-BandMF` | $b, K$ | ✓ | ✓ | ✓✓ | `MC` |

**Algorithm 3** `Nested RA`

**Require:** Parameters $(T, K, b)$
1: Sample $i \sim \text{Uniform}([b])$
2: $\mathcal{E} \leftarrow \{t \in [T] : t \equiv i \pmod b\}$
3: Randomly sample $S \subseteq \mathcal{E}$ of size $K$
4: **return** $S$

**Algorithm 4** `Min Sep RA`

**Require:** Parameters $(T, K, b)$ with $T \geq Kb$
1: Let $\mathcal{V} \leftarrow \{S \subset [T] : |S| = K, |s_i - s_j| \geq b \forall i \neq j\}$
2: Sample $S \sim \text{Uniform}(\mathcal{V})$
3: **return** $S$

Figure 3: Conceptual definitions of our batch selection strategies applied to a single example. `Nested RA` restricts candidates to a residue class modulo $b$, then samples $K$ steps uniformly without replacement. `Min Sep RA` samples uniformly at random from the set of all $K$-element subsets that satisfy the minimum separation $b$. `Cyclic RA` is the same as `Nested RA`, but the residue class $i$ can be chosen arbitrarily rather than randomly.

## D Deferred Proofs

*Proof of Theorem 5.* To prove correctness of the recurrence, we first start with the base cases: If $k = 0$ we have $\mathbf{x} = 0$ in all remaining coordinates, hence the suffixes of $\mathbf{Cx} + \mathbf{z}$, $\mathbf{z}$ are distributionally equivalent. For the case $t > T$, note that when the remaining rounds $t$ to $T$ can only accommodate $k$ more participations, we will always have $p_{k,t} = 1$, i.e. the second term in Eq. (1) is 0. This means that the only way to reach any base cases where $t > T$ and $k > 0$ go through a path through the dynamic programming table where the base case is eventually multiplied by 0. In turn, any base case value for $t > T$ gives correctness.

Now, to prove the recurrence, there are two cases. The first, which happens with probability $\frac{k}{\lceil (T+1-t)/b \rceil}$ ($\lceil (T + 1 - t)/b \rceil$ is the number of possible iterations the example can participate in starting from iteration $t$), is that the example participates in round $t$. In this case, for $P$ we have $(\mathbf{Cx})[t : t + b - 1] = \mathbf{c}$, hence the likelihood ratio of coordinates $t$ to $t + b - 1$ is exactly $\ell_t$. After iteration $t + b - 1$, there are $k - 1$ participations remaining, and the likelihood ratio of the subsequent coordinates is independent of all past history once we condition on $k - 1$ remaining

participations, so $\mathrm{LR}(k-1, t+b)$ correctly gives the likelihood ratio for the remaining coordinates. In the second case, where the example doesn't participate in round $t$, we derive the remaining term $\left(1 - \frac{k}{\lceil (T+1-t)/b \rceil}\right) \cdot \mathrm{LR}(k, t+b)$ analogously.

Our overall likelihood ratio is $\frac{1}{b}\sum_{i=1}^{b} \mathrm{LR}(K, i)$; for each modulus $i$, $\mathrm{LR}(K, i)$ gives the likelihood ratio of the entire output conditioned on that modulus (the likelihood ratio of iterations before $i$ is 1), and each modulus is equally likely.

For the runtime, as we assume $\mathbf{C}$, each $\ell_t$ can be calculated by taking sliding dot products of $\mathbf{c}$ and $\mathbf{y}$, which can be done in $O(T \log T)$ time using FFT, and by computing the norm of each window $\mathbf{c}[t : t+b-1]$, which can be done in $O(b)$ time as we assume $\mathbf{C}$ is Toeplitz. The recurrence takes $O(1)$ time to compute after pre-computing all these values, and there are $KT$ values to compute the recurrence for. □

*Proof of Theorem 6.* To prove correctness, the base cases are analogous to Theorem 5. Conditioned on the example not participating in iterations $t-b+1, \ldots, t-1$, the example participates in $k$ of the remaining $T-t+1$ iterations while satisfying the $b$-separation property. By the observation that there is a bijection between subsets of $[T]$ of size $K$ satisfying the $b$-separation property, and subsets of $[T-(K-1)(b-1)]$ of size $K$, the probability of the example participating in iteration $t$ is $\frac{k}{T-t+1-(k-1)(b-1)}$. The two terms for when the example does or not does not participate can then be proven as in the proof of Theorem 5. □

## E Deferred Experimental Details

**Accounting Details** For methods evaluated via Privacy Loss Distribution (PLD) accounting (`Deterministic`, `Cyclic Poisson`, and the PLD bound for `Cyclic RA`), we use the `dp_accounting` library to calibrate $\sigma$ to the target $(\varepsilon, \delta)$ guarantee. For methods evaluated via Monte Carlo accounting (`Balls-in-Bins`, `Min Sep Poisson`, `Nested RA`, and `Min Sep RA`), we formally calibrate the required noise multiplier $\sigma^*$ using the descending search procedure from Algorithm 5 of Dong and Ganesh (2026). For our new methods, we initialize $\sigma_0$ to the calibrated value under PLD accounting for `Cyclic RA`, and sequentially steps downward by a factor of 1.01 ($\sigma_{k+1} = \sigma_k / 1.01$) until reaching the first candidate noise multiplier that fails the verification certificate. To certify $(\varepsilon, \delta)$-DP at our target $\delta = 10^{-5}$, we draw exactly $N = 10{,}745{,}968$ Monte Carlo samples per candidate $\sigma$. We run 3 independent random trials for each Monte Carlo calibration. This rigorous certification ensures that the calibrated noise levels (and resulting RMSE metrics) reported in our evaluation are provable upper bounds on privacy loss rather than just empirical point estimates. In figures and tables, Monte Carlo methods report the mean across the 3 independent calibration trials; the sample standard deviation is consistently $\leq 0.01$ and thus statistical variance does not meaningfully affect any comparisons we make.

**Batch Selection Strategy Configuration** In Table 3 we provide the settings of each batch selection strategy we use to normalize expected participations to $K$ to ensure fair comparisons.

Table 3: Parameter settings for each batch selection strategy to achieve $K$ expected participations over $T$ iterations with minimum separation $b$.

| Method | Participations | Native parameter | Setting for target $K$ |
|---|---|---|---|
| `Cyclic RA` | Exactly $K$ | $K$ directly | N/A |
| `Nested RA` | Exactly $K$ | $K$ directly | N/A |
| `Min Sep RA` | Exactly $K$ | $K$ directly | N/A |
| `Balls-in-Bins` | Exactly $K$ | Cycle length $L$ | $L = T/K$ |
| `Cyclic Poisson` | Expected K | Sampling prob. $p$ | $p = Kb/T$ |
| `Min Sep Poisson` | Expected K | Sampling prob. $p$ | $p = \frac{K}{T-K(b-1)}$ |
| `Deterministic` | Exactly $K$ | Cycle length $L$ | $L = T/K$ |

## F BIASED BATCH SIZES FOR `MIN SEP RA`

As mentioned in Section 3.2, `Min Sep RA` has one critical drawback that the other batch selection strategies don't have: the expected batch size varies across iterations, with iteration numbers near the endpoints enjoying larger batch sizes in expectation. The severity of this issue depends on the values of $T$, $K$, and $b$. At $b = 1$, there is no issue (standard random allocation), while as $b$ approaches $\frac{T}{K}$, the issue becomes more prominent. Fig. 4 demonstrates this issue for $T = 1024$ and $K = 4$. While the bias is negligible for $b \leq 8$ and small for $b = 32$, it is considerable at $b = 128$ and becomes completely degenerate at $b = 256$ (not shown). It's natural to ask which setting we might find ourselves in: Table 5 shows the optimal value of $b$ in our setting is 128, so this is potentially a serious concern in practice.

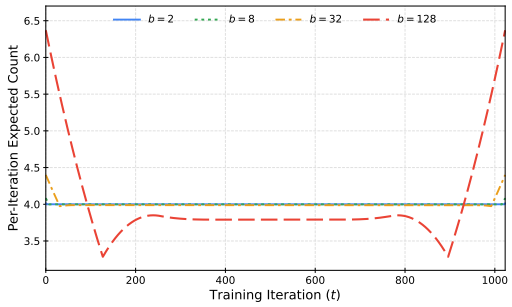


Figure 4: Per-Iteration expected batch size for `Min Sep RA` with $T = 1024$ and $K = 4$.

A simple modification to the sampling algorithm addresses this issue fairly naturally. Standard `Min Sep RA` samples uniformly from the set of indices $s_0, \ldots, s_{K-1}$ satisfying $s_i + b \leq s_{i+1}$ for $i = 0, \ldots, K-2$. Adding a simple "wraparound" constraint $s_{K-1} + b - T \leq s_0$ ensures if a record participates in an early iteration, they cannot also participate in a late iteration, which intuitively addresses the bias issue.

To sample efficiently from this modified set, we can first sample $K$ positive gaps $(g_1, \ldots, g_K)$ summing to $T$ such that every gap satisfies the separation constraint $g_j \geq b$. This can be accomplished via stars and bars by distributing the slack $T - K \cdot b$ across $K$ non-negative integer parts and adding $b$ to each. Accumulating these gaps yields a valid $b$-separated participation subset, to which we apply an independent uniform shift $R \sim \text{Uniform}([T])$ modulo $T$, giving strictly uniform expected batch sizes across iterations. That is, our subset is $\{R, R + g_1, R + g_1 + g_2, \ldots R + \sum_{i=1}^{K-1} g_i\}$, except for each of these values which exceeds $T$, we subtract $T$.

**Theorem 8.** *The above method is equivalent to sampling a uniformly random subset from those that satisfy the $b$-separation constraint (including the wraparound constraint). Furthermore, there are $\frac{T}{k} \cdot \binom{T-Kb+K-1}{K-1}$ such subsets.*

*Proof.* For any $b$-separated subset $t_1 < t_2 < \ldots t_k$, there are exactly $k$ values of $(R, g_1, g_2, \ldots, g_K)$ that produce that subset. Namely, the $i$th such value is $R = t_i, g_1 = t_{i+1 \pmod K} - t_i, g_2 = t_{i+2 \pmod K} - t_{i+1 \pmod K}, \ldots$, where we add $T$ to any negative $g_i$.

This proves a uniform distribution over $(R, g_1, g_2, \ldots, g_K)$ is equivalent to a uniform distribution over $b$-separated subsets, as desired. It also proves the number of such subsets: There are $\binom{T-Kb+K-1}{K-1}$ choices of $g_1, \ldots, g_k$ using stars-and-bars, and $T$ choices of $R$. So there are $T \cdot \binom{T-Kb+K-1}{K-1}$ values of $(R, g_1, g_2, \ldots, g_K)$, and $1/K$ times as many $b$-separated subsets.

□

While this change to the sampling procedure is minor, formal privacy accounting must account for these changes. We describe a new scoring dynamic program to correctly account for this random rotation.

Define $\ell_t$ as we defined it for Eqs. (1), (2). Let $\mathcal{S}$ be all subsets of $[T]$ in the support of our batch selection strategy. Recalling that $|\mathcal{S}| = \frac{T}{K} \cdot \binom{T-Kb+K-1}{K-1}$, our goal is to compute:

$$\frac{P(\mathbf{y})}{Q(\mathbf{y})} = \frac{\sum_{S \in \mathcal{S}} \prod_{t \in S} \ell_t}{\frac{T}{k} \cdot \binom{T-Kb+K-1}{K-1}}.$$

It suffices to give a method to compute the numerator. We decompose the numerator as:

$$\sum_{S \in \mathcal{S}} \prod_{t \in S} \ell_t = \sum_{i=1}^{b-1} \sum_{S \in \mathcal{S}: \min(S)=i} \prod_{t \in S} \ell_t + \sum_{S \in \mathcal{S}: \min(S) \geq b} \prod_{t \in S} \ell_t.$$

We solve for all $b$ sums over sets $S$ via dynamic program, using the following observation: for any $i \in [b-1]$, if $\min(S) = i$, then the remaining $K-1$ elements of $S$ must be in $[i+b, T-b+i]$ and are all separated by at least $b$, but without the wraparound constraint. Similarly, if $\min(S) \geq b$, then we have $K$ elements of $S$ in $[b, T]$ separated by at least $b$ without the wraparound constraint. Since we have reduced to a setting without the wraparound constraint, we can reduce to an analysis similar to that of Eq. (2). In particular, let $\text{SUM}(k, lo, up)$ denote the sum over sets $S$ of $k$ elements in $[lo, up]$ which are at least $b$ apart of $\prod_{t \in S} \ell_t$. Then we have base cases $\text{SUM}(0, lo, up) = 1$ (only the empty set, for which the sum is 1) and $\text{SUM}(k, lo, up) = 0$ if $k > 0$ and $up - lo < 1 + (k-1)b$ (no such set exists, so the sum is empty). We have the recurrence:

$$\text{SUM}(k, lo, up) = \text{SUM}(k, lo+1, up) + e^{\ell_{lo}} \cdot \text{SUM}(k-1, lo+b, up).$$

The proof of correctness for this recurrence is similar to that of Theorem 6 (since both recurrences are solving the same problem, only here we are solving for a sum and not a ratio), so we omit it here. Now we have:

$$\begin{aligned}\sum_{S \in \mathcal{S}} \prod_{t \in S} \ell_t &= \sum_{i=1}^{b-1} \sum_{S \in \mathcal{S}: \min(S)=i} \prod_{t \in S} \ell_t + \sum_{S \in \mathcal{S}: \min(S) \geq b} \prod_{t \in S} \ell_t \\ &= \sum_{i=1}^{b-1} \ell_i \cdot \text{SUM}(K-1, i+b, T-b+i) + \\ &\quad \text{SUM}(K, b, T).\end{aligned}$$

Each recurrence takes $O(1)$ time to evaluation, we evaluate SUM at $O(bKT)$ triplets of inputs (the third input only takes values in $[T-b+1, T]$, and the final combination of values of SUM takes $O(b)$ time, so the overall runtime for computing $P/Q$ is $O(bKT)$. Note this is a $O(b)$ multiplicative overhead compared to Theorem 6; we leave it to future work to determine if this overhead can be removed.

# G Supporting Experiments

## G.1 Optimized Toeplitz Parameters

In Tables 4 and 5 we provide the precise parameters needed to reproduce any cell of our main optimized `Nested RA` and `Min Sep RA` methods respectively. Using this reference table, readers can easily reproduce our reported results by running our formal calibration binary with the provided inputs, and do not need to independently figure out what the optimal settings we found were.

## G.2 Epsilon-Delta Trade-off Curves

While the main paper focuses primarily on RMSE of `DP-BandMF` mechanisms calibrated to specific $(\varepsilon, \delta)$ targets, it is also informative to look at the $(\varepsilon, \delta)$ trade-off curve that `DP-BandMF` has when

Table 4: **Formal Calibration Parameters and Reproduction Details for `Nested RA`** ($\delta = 10^{-5}, N_{\min} = 10{,}745{,}968$). The anchor configuration ($T = 1024, K = 4, \varepsilon = 4.0$) is highlighted in gray. Toeplitz coefficients $\mathbf{C}[i,1] \propto \frac{1}{i} + r^*$ are optimized over $(b, r)$ pairs and certified under Algorithm 5 across 3 independent calibration trials.

| | $T$ | $K$ | $\varepsilon$ | $b^*$ | $r^*$ | $\|\|AC^{-1}\|\|_F^2$ | $\sigma^*$ | RMSE | RMSE$/K$ |
|---|---|---|---|---|---|---|---|---|---|
| Vary $T$ | 256 | 4 | 4.0 | 32 | +0.0397 | 3.435 | 1.119 | 3.84 | 0.96 |
| | 512 | 4 | 4.0 | 64 | +0.0375 | 3.758 | 1.037 | 3.90 | 0.97 |
| | 1024 | 4 | 4.0 | 128 | +0.0350 | 4.026 | 0.981 | 3.95 | 0.99 |
| | 2048 | 4 | 4.0 | 128 | +0.0513 | 4.817 | 0.828 | 3.99 | 1.00 |
| Vary $K$ | 1024 | 1 | 4.0 | 512 | +0.0254 | 3.358 | 0.505 | 1.70 | 1.70 |
| | 1024 | 2 | 4.0 | 256 | +0.0350 | 3.444 | 0.720 | 2.48 | 1.24 |
| | 1024 | 4 | 4.0 | 128 | +0.0350 | 4.026 | 0.981 | 3.95 | 0.99 |
| | 1024 | 8 | 4.0 | 32 | +0.1190 | 5.349 | 1.321 | 7.06 | 0.88 |
| | 1024 | 16 | 4.0 | 16 | +0.2464 | 6.602 | 2.045 | 13.50 | 0.84 |
| Vary $\varepsilon$ | 1024 | 4 | 1.0 | 32 | +0.0136 | 6.823 | 1.703 | 11.62 | 2.90 |
| | 1024 | 4 | 2.0 | 64 | +0.0438 | 4.825 | 1.361 | 6.57 | 1.64 |
| | 1024 | 4 | 4.0 | 128 | +0.0350 | 4.026 | 0.981 | 3.95 | 0.99 |
| | 1024 | 4 | 8.0 | 128 | +0.0150 | 5.761 | 0.482 | 2.78 | 0.70 |
| | 1024 | 4 | 16.0 | 256 | +0.0080 | 5.293 | 0.400 | 2.12 | 0.53 |

Table 5: **Formal Calibration Parameters and Reproduction Details for `Min Sep RA`** ($\delta = 10^{-5}, N_{\min} = 10{,}745{,}968$). The anchor configuration ($T = 1024, K = 4, \varepsilon = 4.0$) is highlighted in gray. Toeplitz coefficients $\mathbf{C}[i,1] \propto \frac{1}{i} + r^*$ are optimized over $(b, r)$ pairs and certified under Algorithm 5 across 3 independent calibration trials.

| | $T$ | $K$ | $\varepsilon$ | $b^*$ | $r^*$ | $\|\|AC^{-1}\|\|_F^2$ | $\sigma^*$ | RMSE | RMSE$/K$ |
|---|---|---|---|---|---|---|---|---|---|
| Vary $T$ | 256 | 4 | 4.0 | 32 | +0.0184 | 4.394 | 0.818 | 3.59 | 0.90 |
| | 512 | 4 | 4.0 | 64 | +0.0196 | 4.496 | 0.801 | 3.60 | 0.90 |
| | 1024 | 4 | 4.0 | 128 | +0.0211 | 4.657 | 0.767 | 3.63 | 0.91 |
| | 2048 | 4 | 4.0 | 256 | +0.0163 | 4.792 | 0.765 | 3.67 | 0.92 |
| Vary $K$ | 1024 | 1 | 4.0 | 512 | +0.0269 | 3.588 | 0.468 | 1.68 | 1.68 |
| | 1024 | 2 | 4.0 | 256 | +0.0226 | 3.963 | 0.570 | 2.26 | 1.13 |
| | 1024 | 4 | 4.0 | 128 | +0.0211 | 4.657 | 0.767 | 3.63 | 0.91 |
| | 1024 | 8 | 4.0 | 32 | +0.0709 | 5.922 | 1.155 | 6.84 | 0.86 |
| | 1024 | 16 | 4.0 | 32 | +0.0121 | 8.654 | 1.509 | 13.06 | 0.82 |
| Vary $\varepsilon$ | 1024 | 4 | 1.0 | 128 | +0.0232 | 6.956 | 1.579 | 10.98 | 2.75 |
| | 1024 | 4 | 2.0 | 128 | +0.0119 | 5.713 | 1.067 | 6.09 | 1.52 |
| | 1024 | 4 | 4.0 | 128 | +0.0211 | 4.657 | 0.767 | 3.63 | 0.91 |
| | 1024 | 4 | 8.0 | 128 | +0.0150 | 5.761 | 0.482 | 2.78 | 0.70 |
| | 1024 | 4 | 16.0 | 256 | +0.0080 | 5.293 | 0.400 | 2.12 | 0.53 |

instantiated with different Toeplitz coefficients and batch selection strategies. Fig. 5 shows that when using the Toeptliz strategy optimized for `Cyclic RA`, both `Nested RA` and `Min Sep RA` provide significant benefits over the PLD-based approaches, and are quite close in behavior with `Min Sep RA` providing slightly better trade-offs. When the Toeplitz coefficients are specifically optimized for `Min Sep RA`, the gap widens substantially.

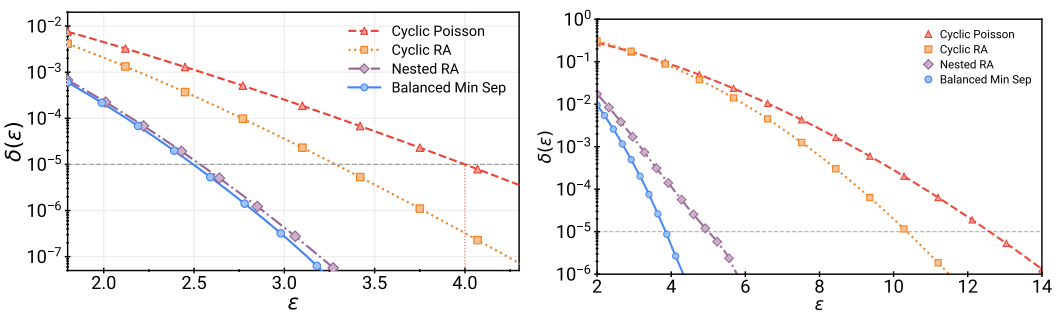


Figure 5: $(\varepsilon, \delta)$-trade-off curves for different batch selection strategies using Toeplitz coefficients optimized for `Cyclic RA` (left) and `Min Sep RA` (right) at a target privacy level of $(4, 10^{-5})$.

### G.3 Monte Carlo optimized Toeplitz Coefficients

When using `Cyclic Poisson` or `Cyclic RA`, the Toeplitz coefficients can be optimized independently of the accounting (since PLD accounting only requires knowing the L2 norm bound on the columns of $\mathbf{C}$). We can observe this directly in Fig. 6, where these two lines overlap [6]

When using Monte Carlo accounting, we showed that we can benefit substantially from optimizing Toeplitz coefficients and accounting for the impact on privacy amplification. Fig. 6 shows how the optimal coefficients vary under different batch selection strategies. We see that under `Nested RA` and `Min Sep RA`, the Toeplitz coefficients drop off more sharply. This drop off occurs because amplification for `Nested RA` and `Min Sep RA` improves as coefficients drop off faster, whereas the coefficients themselves do not affect amplification cyclic methods as long as the column norm is fixed.

To see why coefficients dropping off helps amplification, intuitively, privacy amplification helps because it increases the adversary's uncertainty about what signal an example contributes to $\mathbf{Cx}$. When pairs of columns of $\mathbf{C}$ have high dot products, this reduces the uncertainty. As an extreme example, if $\mathbf{C}$ were the all-ones matrix, then $\mathbf{Cx}$ would only depend on the number of participations (which we keep fixed) and then privacy amplification wouldn't help at all. More broadly, if $\mathbf{C}[:, i]$ and $\mathbf{C}[:, j]$ have high dot product, then randomizing whether an example participates in round $i$ or $j$ does not add much more uncertainty for the adversary. When the coefficients drop off, the dot products between columns decreases and thus privacy amplification is increased. `Min Sep RA` has more entropy in $\mathbf{x}$ than `Nested RA` does, so it benefits more from this uncertainty, hence why its coefficients drop off even faster.

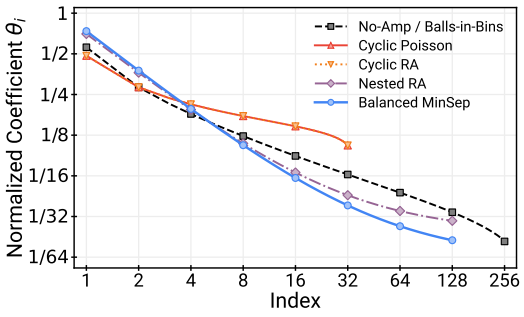


Figure 6: Toeplitz coefficients optimized for different batch selection strategies when $T = 1024$, $K = 4$, and $\varepsilon = 4$.

[6] This is not always the case, as in some settings the optimal number of bands may differ between these batch selection strategies.