Learning Rate Scheduling with Matrix Factorization for Private Training

Nikita P. Kalinin
Institute of Science and Technology Austria, Klosterneuburg, Austria

Joel Daniel Andersson
Institute of Science and Technology Austria, Klosterneuburg, Austria

Abstract

We study differentially private model training with stochastic gradient descent under learning rate scheduling and correlated noise. Although correlated noise, in particular via matrix factorizations, has been shown to improve accuracy, prior theoretical work focused primarily on the prefix-sum workload. That workload assumes a constant learning rate, whereas in practice learning rate schedules are widely used to accelerate training and improve convergence. We close this gap by deriving general upper and lower bounds for a broad class of learning rate schedules in both single- and multi-epoch settings. Building on these results, we propose a learning-rate-aware factorization that achieves improvements over prefix-sum factorizations under both MaxSE and MeanSE error metrics. Our theoretical analysis yields memory-efficient constructions suitable for practical deployment, and experiments on CIFAR-10 and IMDB datasets confirm that schedule-aware factorizations improve accuracy in private training.

2012 ACM Subject Classification Security and privacy; Computing methodologies → Machine learning

Keywords and phrases differential privacy, machine learning, matrix factorization

Digital Object Identifier 10.4230/LIPIcs.FORC.2026.2

Related Version *Full Version*: `https://arxiv.org/abs/2511.17994`

Funding *Nikita P. Kalinin*: Funded in part by the Austrian Science Fund (FWF) [10.55776/COE12]. *Joel Daniel Andersson*: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (MoDynStruct, No. 101019564). Additional funding by Providentia, a Data Science Distinguished Investigator grant from Novo Nordisk Fonden, with additional support from VILLUM Investigator grant 54451.

Acknowledgements We thank Rasmus Pagh, Christoph Lampert and Jalaj Upadhyay for valuable comments on an early draft. We thank Ryan Mckenna for a fruitful discussion on the experiment design. We thank Antti Honkela for sharing insights on learning rate scheduling and DP.

1 Introduction

Privacy has become a major concern as machine learning systems are trained on sensitive data such as personal communications, financial transactions, and medical records. Beyond the risk of direct data exposure, models themselves may memorize and unintentionally reveal private information, creating serious ethical and security challenges. These concerns are especially pressing for production-level large language models trained on vast and heterogeneous datasets.

A widely studied approach to mitigating these risks is differential privacy (DP), which provides formal mathematical guarantees that the output of a learning algorithm does not reveal sensitive information about any individual training example [12]. In practice, DP is

often achieved by injecting carefully calibrated noise into either the gradients, ensuring that an adversary cannot infer the presence or absence of a single data point with high confidence. More recently, large-scale efforts such as VaultGemma [30] have demonstrated that it is possible to train billion-parameter models with rigorous privacy guarantees, showing that DP can be integrated into state-of-the-art architectures without prohibitive utility loss.

To make model training differentially private, algorithms typically inject noise into the gradients to mask the contribution of any individual data point. The most common approach, DP-SGD, adds independent Gaussian noise at each update, which provides strong privacy guarantees but can significantly reduce accuracy [1]. *Matrix factorization* has emerged as a more general alternative that introduces correlations in the injected noise, enabling improved accuracy while preserving privacy [8, 9]. The method is also applied beyond centralized model training to applications such as federated learning [6, 32] as well as decentralized learning [5]. The approach has also seen practical adoption, with Google reporting its use for training production on-device language models in their 2024 blog post "Advances in private training for production on-device language models" [31].

Recent work has focused on making matrix factorization memory efficient [25, 2, 20, 26], and it has also been analyzed theoretically, mostly in the setting of Toeplitz workloads [13, 16, 18, 11]. However, existing utility analyses assume a constant learning rate. While Denisov, McMahan, Rush, Smith and Thakurta [10] introduced a non-Toeplitz workload with varying learning rates, its theoretical properties remain largely unexplored. In this work, we address this gap by studying matrix factorization under learning rate schedules.

Learning rate scheduling plays a critical role in the optimization of machine learning models, and a variety of strategies have been proposed in the literature. Popular approaches include cosine annealing [23] and cyclical learning rates [29], which adapt the step size during training to improve convergence. Another common technique is warm-starting [15], where models begin with a small learning rate that is gradually increased, as used in large-scale training setups [14]. In this work, we focus on learning rate decays available in PyTorch [27] such as exponential, linear, polynomial and cosine, which are widely used and can be easily applied in practice.

Learning rate scheduling can be particularly useful in private training when the number of iterations is limited. By accelerating convergence, it enables higher accuracy in settings such as warm-up training [22], private fine-tuning [24], and training under computational constraints. It has also been combined with matrix factorization as a form of learning rate cool-down [9, 8, 7], and was shown to provide improvements over fixed learning rates in [10], where the workload of our interest was originally introduced.

Contributions

- We theoretically analyze the problem of matrix factorization under learning rate scheduling. We establish general lower and upper bounds for MaxSE and MeanSE in single-epoch, as well as for MeanSE in multi-epoch for a large class of schedulers. Here, MaxSE characterizes the maximum variance of the added noise, while MeanSE captures the average variance across iterations.
- We propose a learning-rate-aware Toeplitz factorization, which for exponentially decaying learning rate is provably optimal in MaxSE under single-epoch and improves upon the proposed upper bound for MeanSE. We adopt this factorization for memory efficient, multi-epoch training by making it banded inverse.
- We show numerically that the proposed factorization is close to optimal in all metrics.

- We show experimentally on CIFAR-10 and IMDB datasets that banded inverse factorizations benefit from learning rate scheduling. Moreover, we demonstrate that the proposed learning-rate-aware factorization achieves even further accuracy improvements.

2 Background

The most common way to train differentially private models is by using *DP-SGD* [1]. At each step we receive a gradient $g_i \in \mathbb{R}^d$, clip it to a fixed ℓ_2 norm $\zeta > 0$, and add appropriately scaled independent Gaussian noise $z_i \sim \mathcal{N}(0, I_d\,\sigma^2)$, where σ depends on the target privacy level (ϵ, δ). The model is then updated as

$$\theta_i = \theta_{i-1} - \eta_i\big(\mathrm{clip}(g_i, \zeta) + z_i\big), \tag{1}$$

where η_i is the learning rate at step i.

This procedure can be improved by correlating the noise across iterations. To formalize this, we define a matrix $G \in \mathbb{R}^{n\times d}$ of stacked gradients, a matrix $\Theta \in \mathbb{R}^{n\times d}$ of intermediate models, and a workload matrix $A \in \mathbb{R}^{n\times n}$ that encodes the training process such that $\Theta = AG$. If we use a constant learning rate, this matrix, denoted A_1, is a lower triangular matrix of ones. For varying learning rates we instead use a matrix A_χ, described later.

To ensure that the intermediate models are differentially private, we can apply a matrix factorization mechanism. Specifically, we factorize A into two matrices B and C, then compute CG, add Gaussian noise $Z \in \mathcal{N}(0, \sigma^2)^{n\times d}$ to ensure differential privacy, and finally multiply the result by B as post-processing:

$$\widehat{AG} = B(CG + Z) = A(G + C^{-1}Z), \tag{2}$$

which is equivalent to adding correlated Gaussian noise with covariance structure induced by C^{-1} to the gradients.

The remaining question is: *how much noise must be added, and does this procedure remain differentially private when the gradients are not known in advance but depend on the current model?* The foundational work of [10] shows that the procedure is indeed differentially private, even when the gradients adaptively depend on the model, provided we add noise of scale

$$\sigma = \zeta \cdot \sigma_{\epsilon,\delta} \cdot \mathrm{sens}(C) = \zeta \cdot \sigma_{\epsilon,\delta} \cdot \|C\|_{1\to 2}, \tag{3}$$

where ζ denotes the clipping norm, and $\sigma_{\epsilon,\delta}$ is the noise multiplier of the standard Gaussian mechanism, which can be computed numerically [4]. The term $\mathrm{sens}(C)$ represents the global sensitivity of the Gaussian mechanism for the product CG when the row or rows corresponding to a single datapoint in G change; it can be computed explicitly as $\|C\|_{1\to 2}$, the maximum column norm of C. The case of multi-participation (multi-epoch) is discussed in Section 4.2. Now we have all the steps to train the model with differential privacy as presented in Algorithm 1.

The choice of factorization $A = BC$ significantly impacts the quality of the private estimation. Following the work of [10, 16] we quantify the *approximation quality* by either the mean squared error (MeanSE) or the maximum expected squared error (MaxSE), which can be computed as

$$\mathrm{MeanSE}(B, C) = \sqrt{\tfrac{1}{n}\,\mathbb{E}_Z\|AG - \widehat{AG}\|_{\mathrm{F}}^2} = \tfrac{1}{\sqrt{n}}\,\|B\|_{\mathrm{F}}\,\|C\|_{1\to 2}\,\sigma_{\epsilon,\delta}\,\zeta, \tag{4}$$

$$\mathrm{MaxSE}(B, C) = \sqrt{\mathbb{E}_Z\|AG - \widehat{AG}\|_{\infty}^2} = \|B\|_{2\to\infty}\,\|C\|_{1\to 2}\,\sigma_{\epsilon,\delta}\,\zeta, \tag{5}$$

Algorithm 1 Differentially Private SGD with Matrix Factorization and Learning Rate Schedules.

Require: Model initialization $\theta_0 \in \mathbb{R}^d$, dataset D, batchsize b, model loss $\ell(\theta, d)$, clipnorm $\zeta > 0$, learning rate η, correlation matrix $C \in \mathbb{R}^{n\times n}$, learning rate scheduler χ_i, noise matrix $Z \in \mathbb{R}^{n\times d}$ with i.i.d. entries $\mathcal{N}(0, \mathrm{sens}^2(C)\sigma^2_{\epsilon,\delta}\zeta^2)$.

for $i = 1, 2, \ldots, n$ **do**

$S_i \leftarrow \{d_1, \ldots, d_b\} \subseteq D$ (select a data batch[1])

$g_i \leftarrow \nabla_\theta \ell(\theta_{i-1}, d_j)$ for $j = 1, \ldots, b$

$x_i \leftarrow \sum_{j=1}^{b} \min(1, \zeta/||g_j||) \cdot g_j$ (clip gradients)

$\hat{x}_i \leftarrow \frac{1}{b}\left(x_i + [C^{-1}Z]_{[i,\cdot]}\right)$

$\theta_i \leftarrow \theta_{i-1} - (\chi_i\eta)\hat{x}_i$

Ensure: θ_n

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm and $\|\cdot\|_{2\to\infty}$ the maximum row ℓ_2-norm. These approximation errors are independent of G, and the term $\sigma_{\epsilon,\delta}\zeta$ is independent of the matrix factorization. To isolate the contribution of the factorization (B, C), we will use the notation $\mathrm{MeanSE}(B, C)$, $\mathrm{MaxSE}(B, C)$ assuming $\zeta = \sigma_{\epsilon,\delta} = 1$ in the theoretical analysis.

3 Method

We now turn to the workload of stochastic gradient descent (SGD) with learning rate scheduling. Let $\chi_1, \chi_2, \ldots, \chi_n$ be a sequence with $\min \chi_t = \beta > 0$ and $\max \chi_t = 1$, representing a learning rate scheduler such that the actual learning rate at time t is $\eta_t = \eta\chi_t$. We assume that β is reasonably separated from 1, as the regime $\beta \to 1$ is not of interest since it nullifies the benefits of scheduling. The workload matrix of interest is then:

$$A_\chi = \begin{pmatrix} \chi_1 & 0 & 0 & \cdots & 0 \\ \chi_1 & \chi_2 & 0 & \cdots & 0 \\ \chi_1 & \chi_2 & \chi_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \chi_1 & \chi_2 & \chi_3 & \cdots & \chi_n \end{pmatrix} = A_1 \times D, \tag{6}$$

where A_1 is a prefix-sum matrix (lower triangular matrix of all ones) and D is a diagonal matrix of learning rates, i.e., $D = \mathrm{diag}(\chi_1, \ldots, \chi_n)$. We will study the problem of *optimal matrix factorization* in MaxSE and MeanSE metrics for the matrix A_χ with the **learning rate decays** given in Table 1. We emphasize that $\chi_n = \beta$ for all the listed decays. For the experiments, we will also include the constant learning rate ($\chi_k = 1$).

In this work, we prove general lower and upper bounds for the MaxSE and MeanSE errors. For the upper bound, we use a prefix-sum-based factorization given by $B = A_\chi(A_1)^{-1/2}$ and $C = A_1^{1/2}$, which has been shown to be nearly optimal up to the next asymptotic term for the prefix sum problem ($\chi_k = 1$) [17]. To further improve the bounds, we propose a learning-rate-aware factorization. To define it, let A^{Toep}_χ denote the Toeplitz matrix with $\chi_1, \ldots, \chi_n$ on its subdiagonals:

[1] For batch formation, one could simply take the data and go over the epochs in the same order, thereby guaranteeing the separation (see Subsection 4.2). When amplification by subsampling is included (see the discussion in the Experiments section, Section 5), one should first partition the data into batches in a randomized way and repeat this partitioning across epochs.

Table 1 Learning rate decays.

Exponential	$\chi_k = \beta^{\frac{k-1}{n-1}}$
Polynomial	$\chi_k = \beta + (1-\beta)\,\dfrac{\left(\frac{n}{k}\right)^{\gamma} - 1}{n^{\gamma} - 1},\ \gamma \geq 1$
Linear	$\chi_k = 1 - \frac{k-1}{n-1}(1-\beta)$
Cosine	$\chi_k = \beta + \frac{1-\beta}{2}\left(1 + \cos\left(\frac{k-1}{n-1}\pi\right)\right)$

$$A_{\chi}^{\mathrm{Toep}} = \begin{pmatrix} \chi_1 & 0 & 0 & \dots & 0 \\ \chi_2 & \chi_1 & 0 & \dots & 0 \\ \chi_3 & \chi_2 & \chi_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \chi_n & \chi_{n-1} & \chi_{n-2} & \dots & \chi_1 \end{pmatrix} \tag{7}$$

We propose $C_{\chi} = (A_{\chi}^{\mathrm{Toep}})^{1/2}$ as a learning-rate-aware correlation matrix. To analyze its properties, we consider the exponentially decaying learning rate $\chi_t = \beta^{\frac{t-1}{n-1}} = \alpha^{t-1}$ with $\alpha = \beta^{\frac{1}{n-1}}$. In this setting, the correlation matrix can be computed explicitly as

$$C_{\alpha} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \alpha r_1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{n-1} r_{n-1} & \alpha^{n-2} r_{n-2} & \dots & 1 \end{pmatrix}, \tag{8}$$

where the coefficients are $r_j = \left|\binom{-1/2}{j}\right| = \frac{1}{4^j}\binom{2j}{j}$.

4 Results

In this work, we derive upper and lower bounds on the MaxSE and MeanSE errors of the learning rate scheduling workload A_{χ} for a large class of learning rate schedulers $\chi_1, \dots, \chi_n$. In the following theorem, we prove an upper bound based on the prefix-sum factorization $A_{\chi} A_1^{-1/2} \times A_1^{1/2}$.

▶ **Theorem 1.** *Let $(\chi_t)_{t=1}^n$ be a sequence on $[\beta, 1]$ for some constant $\beta > 0$. For $n \geq 2$ we define*

$$\Delta_t = |\chi_t - \chi_{t+1}| \qquad (\textit{for all } 1 \leq t \leq n-1). \tag{9}$$

If either of the following two conditions holds ($c > 0$ an absolute constant):

$$\Delta_t \leq \frac{c}{t(1+\log t)} \qquad (\textit{for all } 1 \leq t \leq n-1), \qquad \text{or} \qquad \sum_{t=1}^{n-1} \Delta_t^2 = o\left(\frac{\log n}{n}\right), \tag{10}$$

then the factorization $B_{\chi} \times A_1^{1/2}$, where $B_{\chi} := A_{\chi}(A_1)^{-1/2}$, satisfies

$$\mathrm{MaxSE}(B_{\chi}, A_1^{1/2}) = \Theta\left(\sqrt{\log n} \cdot \sqrt{\max_{m \in [n]} \chi_m^2 \log m}\right), \tag{11}$$

$$\mathrm{MeanSE}(B_{\chi}, A_1^{1/2}) = \Theta\left(\sqrt{\log n} \cdot \sqrt{\frac{1}{n}\sum_{m=1}^{n} \chi_m^2 \log m}\right). \tag{12}$$

The conditions assumed in Theorem 1 are satisfied for all learning rate decays presented in Table 1, more formally:

▶ **Lemma 2.** *Every learning rate schedule $(\chi_t)_{t=1}^n$ with constant $\beta \in (0, 1/e)$ presented in Table 1 satisfies the assumptions of Theorem 1.*

Moreover, in this work we also prove general lower bounds for any learning rate schedules:

▶ **Theorem 3.** *Let $A_\chi = A_1 D_\chi$, where $D_\chi = \operatorname{diag}(\chi_1, \ldots, \chi_n)$ with positive $\chi_t > 0$. Then*

$$\inf_{B \times C = A_\chi} \operatorname{MaxSE}(B, C) \;\geq\; \max_{1 \leq t \leq n} \frac{1}{\pi} \left(\min_{j \leq t} \chi_j\right) \log t, \tag{13}$$

$$\inf_{B \times C = A_\chi} \operatorname{MeanSE}(B, C) \;\geq\; \max_{1 \leq t \leq n} \frac{1}{\pi} \sqrt{\frac{t}{n}} \left(\min_{j \leq t} \chi_j\right) \log t. \tag{14}$$

In particular, plugging in the exponential learning rate decay $\chi_k = \beta^{\frac{k-1}{n-1}}$ yields the following upper and lower bounds.

▶ **Corollary 4.** *For exponential learning rate decay $\chi_k = \beta^{\frac{k-1}{n-1}}$ with $\beta \in (0, 1/e)$, the prefix-sum–based factorization $A_\chi = A_\chi (A_1)^{-1/2} \times A_1^{1/2}$ gives the following values for MaxSE and MeanSE:*

$$\operatorname{MaxSE}(B_\chi, A_1^{1/2}) = \Theta\left(\sqrt{\log n} \sqrt{\log \frac{n}{\log(1/\beta)}}\right), \tag{15}$$

$$\operatorname{MeanSE}(B_\chi, A_1^{1/2}) = \Theta\left(\frac{\log n}{\sqrt{\log(1/\beta)}}\right). \tag{16}$$

▶ **Corollary 5.** *Suppose $\chi_k = \beta^{\frac{k-1}{n-1}}$ with $\beta \in (0, 1/e)$. Then*

$$\inf_{B \times C = A_\chi} \operatorname{MaxSE}(B, C) = \Omega\left(\log \frac{n}{\log(1/\beta)}\right) \tag{17}$$

$$\inf_{B \times C = A_\chi} \operatorname{MeanSE}(B, C) \;=\; \Omega\left(\frac{1}{\sqrt{\log(1/\beta)}} \log \frac{n}{\log(1/\beta)}\right). \tag{18}$$

Note that our results do not report the leading constant in the error bounds. The reason is simply that the workload matrices we study are harder to analyze than the much-studied lower-triangular matrix of all-ones, for which more fine-grained analysis has been performed. For comparison, even for the square-root factorization of the Toeplitz workload with exponential weight decay, the exact leading constant has not been determined [16, 19].

We further improve the upper bound by considering a learning-rate–aware factorization $C = (A_\chi^{\text{Toep}})^{1/2}$, which can be computed explicitly for the exponential learning rate decay $\chi_k = \beta^{\frac{k-1}{n-1}} = \alpha^{k-1}$. This yields the factorization $A_\chi = B_\alpha \times C_\alpha$, where C_α is defined in equation (8), and B_α is obtained as $A_\chi (C_\alpha)^{-1}$.

In Lemma 7 of [19], the sensitivity of the matrix C_α has been computed as:

$$\|C_\alpha\|_{1 \to 2} = \mathcal{O}\left(\tfrac{1}{\alpha} \sqrt{\log \tfrac{1}{1-\alpha^2}}\right) = \mathcal{O}\left(\sqrt{\log \tfrac{n}{\log(1/\beta)}}\right). \tag{19}$$

We then bound both the maximum row norm and the Frobenius norm of B_α, which leads to the following lemma.

Table 2 Factorizations with corresponding MaxSE and MeanSE errors for exponential learning rate scheduling $\chi_t = \beta^{\frac{t-1}{n-1}}$ for $\beta \in (0, 1/e)$. The first three sets of bounds can be found in Lemma 15 in the Appendix. The errors for square-root factorization (d) can be found in Corollary 8. Learning-rate-aware factorization (e) corresponds to Lemma 6. The prefix-sum-based factorization (f) corresponds to Corollary 4. The lower bounds are Corollary 5. Complete proofs for every bound can be found in the full version of the paper.

Factorization	**MaxSE**	**MeanSE**
(a) $A_\chi = A_1^{1/2} \times A_1^{1/2} D$	$\Theta(\log n)$	$\Theta(\log n)$
(b) $A_\chi = A_\chi \times I$	$\Theta\left(\sqrt{\frac{n}{\log 1/\beta}}\right)$	$\Theta\left(\sqrt{\frac{n}{\log 1/\beta}}\right)$
(c) $A_\chi = I \times A_\chi$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
(d) $A_\chi = A_\chi^{1/2} \times A_\chi^{1/2}$	$\Omega\left(\sqrt{\log n}\sqrt{\log \frac{n}{\log 1/\beta}}\right)$	$\Omega\left(\frac{\log n}{\sqrt{\log(1/\beta)}}\right)$
(e) $A_\chi = A_\chi (A_\chi^{\mathrm{Toep}})^{-1/2} \times (A_\chi^{\mathrm{Toep}})^{1/2}$	$\mathcal{O}\left(\log \frac{n}{\log 1/\beta}\right)$	$\mathcal{O}\left(\sqrt{\frac{\log n}{\log 1/\beta}}\sqrt{\log \frac{n}{\log 1/\beta}}\right)$
(f) $A_\chi = A_1 D A_1^{-1/2} \times A_1^{1/2}$	$\Theta\left(\sqrt{\log n}\sqrt{\log \frac{n}{\log 1/\beta}}\right)$	$\Theta\left(\frac{\log n}{\sqrt{\log 1/\beta}}\right)$
Lower Bound	$\Omega\left(\log \frac{n}{\log 1/\beta}\right)$	$\Omega\left(\frac{1}{\sqrt{\log 1/\beta}} \log\left(\frac{n}{\log 1/\beta}\right)\right)$

▶ **Lemma 6.** *Let $\beta \in (0, 1/e)$ and $\alpha = \beta^{1/(n-1)}$. For the factorization $A_\chi = B_\alpha \times C_\alpha$,*

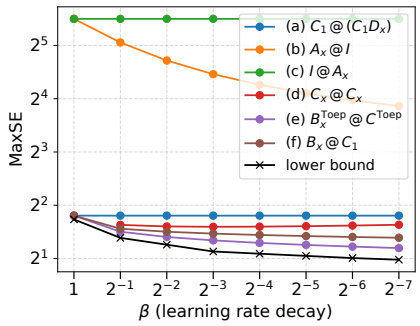
$$\mathrm{MaxSE}(B_\alpha, C_\alpha) = \mathcal{O}\left(\log \frac{n}{\log(1/\beta)}\right), \tag{20}$$

$$\mathrm{MeanSE}(B_\alpha, C_\alpha) = \mathcal{O}\left(\sqrt{\frac{\log n}{\log(1/\beta)}}\sqrt{\log \frac{n}{\log(1/\beta)}}\right). \tag{21}$$

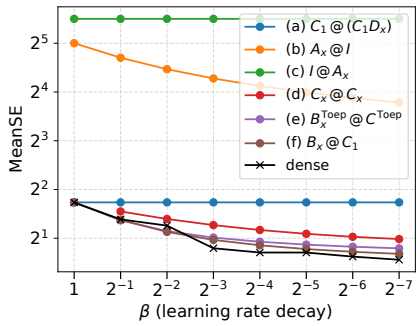
This factorization achieves the **optimal rate for the MaxSE error** and, asymptotically, performs better than alternative factorizations for the MeanSE error.

We summarize the errors for the exponential learning rate decay in Table 2. In addition, we consider four alternative factorizations: the trivial factorizations $A_\chi \times I$ and $I \times A_\chi$, the prefix-sum–inspired factorization $A_1^{1/2} \times A_1^{1/2} D$, and the square-root factorization $A_\chi^{1/2} \times A_\chi^{1/2}$. The square-root factorization is highly nontrivial to obtain since the matrix is not Toeplitz; we defer its treatment to Section 4.1.

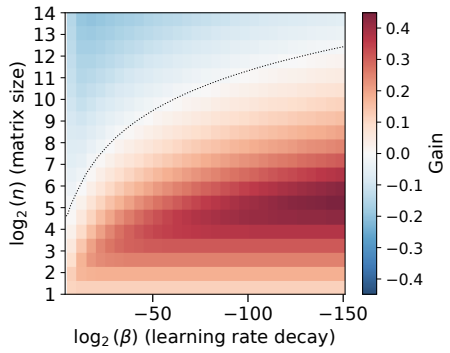
We then numerically compare the proposed factorizations in the single-epoch (single-participation) setting using the MaxSE and MeanSE metrics, as functions of the learning rate decay β and the matrix size n (see Figure 1 for exponential decay and Figure 5 in the Appendix for other learning rate decays). As an approximation of the actual optimal value for MeanSE, we use a dense factorization [10] implemented in the jax-privacy library [3]. On the plots, we refer to this approximation as "dense". For MaxSE, it is computationally infeasible to compute the exact optimal value for large matrix sizes. Therefore, we rely on the lower bound derived in Theorem 3, which we denote on the plots as "lower bound". We observe that our learning-rate-aware factorization outperforms the others in terms of MaxSE. However, for the proposed values of n and β, it performs worse than the prefix sum based factorization in terms of MeanSE. To further investigate this, we plot the colormap of the gain over the prefix sum based approach (see Figure 1). In the blue regions, our method performs worse, while in the red regions it performs better. As can be seen, for any fixed n, sufficiently small values of β lead to the learning-rate-aware factorization outperforming the prefix sum based approach, thereby numerically validating our theoretical findings.



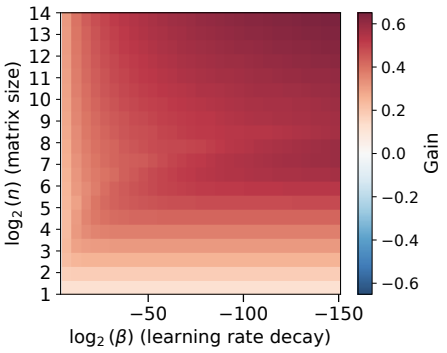
(a) $n = 2048$, MaxSE.



(b) $n = 2048$, MeanSE.



(c) MeanSE error gain.



(d) MaxSE error gain.

Figure 1 Comparison of MaxSE and MeanSE errors under an **exponentially decaying** learning rate, for the proposed factorizations (see Table 2), with fixed matrix size $n = 2048$ and varying decay β. We refer to the approximately optimal value of MeanSE computed by dense factorization [10] as "dense." For MaxSE, we report a lower bound since no scalable and accurate solution for its optimal value is available. The bottom row compares our learning-rate-aware factorization with the prefix-sum based one, validating the theoretical improvements in both MeanSE and MaxSE.

4.1 Matrix Square Root of the Workload

As one of the baseline factorizations we propose the square-root factorization

$$A_\chi = A_\chi^{1/2} \times A_\chi^{1/2}, \quad \text{where} \quad A_\chi = \begin{pmatrix} \chi_1 & 0 & 0 & \cdots & 0 \\ \chi_1 & \chi_2 & 0 & \cdots & 0 \\ \chi_1 & \chi_2 & \chi_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \chi_1 & \chi_2 & \chi_3 & \cdots & \chi_n \end{pmatrix}. \tag{22}$$

In the case of exponential learning rate decay we can compute the matrix square root explicitly and tightly bound its values from below.

▶ **Theorem 7.** *For any $n \geq 1$ and $\alpha \in (0, 1)$, with learning rates $\chi_i = \alpha^{i-1}$ the following lower bound holds:*

$$(A_\chi^{1/2})_{m,l} = \alpha^{(l-1)/2} \prod_{k=1}^{m-l} \frac{1 - \alpha^{k-1/2}}{1 - \alpha^k} \geq \alpha^{(l-1)/2} \max \left\{ \left| \binom{-1/2}{n} \right|, \frac{\sqrt{1-\alpha^2}}{\Gamma_{\alpha^2}(1/2)} \right\}, \tag{23}$$

where $\Gamma_q(x)$ denotes the q-Gamma function, and $\lim_{\alpha \to 1^-} \Gamma_{\alpha^2}(1/2) = \Gamma(1/2) = \sqrt{\pi}$.

Using the lower bound, we now establish the following bounds for the MaxSE and MeanSE errors under an exponentially decaying learning rate.

▶ **Corollary 8.** *Let $\beta \in (0, 1/e)$ and $\alpha = \beta^{1/(n-1)}$. For the square-root factorization $A_\chi = A_\chi^{1/2} A_\chi^{1/2}$, we have*

$$\mathrm{MaxSE}(A_\chi^{1/2}, A_\chi^{1/2}) = \Omega\left(\sqrt{\log n}\sqrt{\log \frac{n}{\log(1/\beta)}}\right), \tag{24}$$

$$\mathrm{MeanSE}(A_\chi^{1/2}, A_\chi^{1/2}) = \Omega\left(\frac{\log n}{\sqrt{\log(1/\beta)}}\right). \tag{25}$$

We prove these statements next, beginning with necessary lemmas.

▶ **Lemma 9.** *For a specific choice of the learning rate coefficients $\chi_i = \alpha^{2i}$ with $\alpha \in (0, 1)$, we have:*

$$(A_\chi^{1/2})_{m,l} = \alpha^l \prod_{k=1}^{m-l} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} \tag{26}$$

Proof. To prove that the coefficients of the square root have the proposed form, we need to show that the square of this matrix is equal to the original one. That is, for all $1 \le l \le m \le n$, we show that:

$$\sum_{j=l}^{m} \alpha^j \prod_{k=1}^{m-j} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} \cdot \alpha^l \prod_{k=1}^{j-l} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} = \alpha^{2l} \tag{27}$$

or equivalently,

$$\sum_{j=0}^{m-l} \alpha^j \prod_{k=1}^{m-l-j} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} \prod_{k=1}^{j} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} = 1, \tag{28}$$

which is a convolution of the sequences a_j and $a_j\alpha^j$, where

$$a_j = \prod_{k=1}^{j} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} = \frac{(\alpha;\alpha^2)_j}{(\alpha^2;\alpha^2)_j}, \tag{29}$$

and $(a;q)_n$ denotes the q-Pochhammer symbol, given by $\prod_{k=0}^{n-1}(1-aq^k)$. We will prove the identity using generating functions. First, we find the generating function of a_j:

$$f(x) = \sum_{j=0}^{\infty} a_j x^j = \sum_{j=0}^{\infty} \frac{(\alpha;\alpha^2)_j}{(\alpha^2;\alpha^2)_j} x^j = \frac{(\alpha x;\alpha^2)_\infty}{(x;\alpha^2)_\infty}, \tag{30}$$

where the last equality follows from the q-binomial theorem. Therefore, the generating function of the convolution of a_j and $a_j\alpha^j$ is:

$$f(x)f(\alpha x) = \frac{(\alpha x;\alpha^2)_\infty}{(x;\alpha^2)_\infty} \cdot \frac{(\alpha^2 x;\alpha^2)_\infty}{(\alpha x;\alpha^2)_\infty} = \frac{(\alpha^2 x;\alpha^2)_\infty}{(x;\alpha^2)_\infty} = \prod_{n=0}^{\infty} \frac{1-x\alpha^{2n+2}}{1-x\alpha^{2n}} = \frac{1}{1-x}, \tag{31}$$

as the product telescopes, yielding the generating function of the unit sequence $(1, 1, 1, \dots)$, thus concluding the proof. ◀

▶ **Lemma 10.** *For any $n \geq 1$ and $\alpha \in (0,1)$, the following lower bound holds:*

$$\prod_{k=1}^{n} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} \geq \max\left\{ \left| \binom{-1/2}{n} \right|, \frac{\sqrt{1-\alpha^2}}{\Gamma_{\alpha^2}(1/2)} \right\}, \tag{32}$$

where $\Gamma_q(x)$ denotes the q-Gamma function, and $\lim_{\alpha\to 1^-} \Gamma_{\alpha^2}(1/2) = \Gamma(1/2) = \sqrt{\pi}$.

Proof. First, we show that $f_n(\alpha) = \prod_{k=1}^{n} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}}$ is a decreasing function of α. Therefore,

$$f_n(\alpha) \geq f_n(1) = \prod_{k=1}^{n} \frac{2k-1}{2k} = \left| \binom{-1/2}{n} \right|. \tag{33}$$

To prove this, we observe that each individual term is a decreasing function of α:

$$\frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} = 1 - \frac{\alpha^{2k-1}-\alpha^{2k}}{1-\alpha^{2k}} = 1 - \frac{\alpha^{-1}-1}{\alpha^{-2k}-1} = 1 - \frac{1}{1+\alpha^{-1}+\cdots+\alpha^{-(2k-1)}}. \tag{34}$$

For the second part of the inequality, we show that

$$f_n(\alpha) \geq f_\infty(\alpha) = \prod_{k=1}^{\infty} \frac{1-\alpha^{2k-1}}{1-\alpha^{2k}} = \frac{(\alpha;\alpha^2)_\infty}{(\alpha^2;\alpha^2)_\infty} = \frac{\sqrt{1-\alpha^2}}{\Gamma_{\alpha^2}(1/2)}, \tag{35}$$

where the inequality holds because each term of the product is less than 1, the infinite product converges, and the q-Gamma function is defined by

$$\Gamma_q(x) = (1-q)^{1-x} \frac{(q;q)_\infty}{(q^x;q)_\infty}. \tag{36}$$

This concludes the proof. ◀

Proof of Theorem 7. The proof follows from combining Lemma 9 for the equality and Lemma 10 for the lower bound. For convenience, we considered $\chi_i = \alpha^{2i}$ in those lemmas. To achieve α^{i-1}, we first divide the square-root matrix by α so that we start from learning rates of 1 rather than α^2. Then, we replace α with $\alpha^{1/2}$, which concludes the proof. ◀

Proof of Corollary 8. To use Lemma 9 and Lemma 10, we need to adjust the choice of α, as previous lemmas consider $\chi_k = \alpha^{2k}$ while here $\chi_k = \alpha^{k-1}$. This gives

$$(A_\chi^{1/2})_{m,l} \;\geq\; \alpha^{(l-1)/2}\, r_{m-l}. \tag{37}$$

Thus the maximum column norm of $A_\chi^{1/2}$ is at least the norm of its first column, which in turn is at least the maximum column norm of $A_1^{1/2}$; the latter is $\Theta(\log n)$.

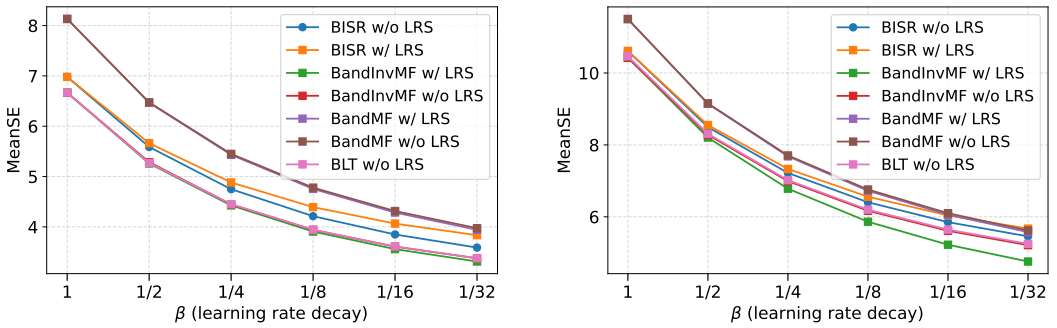
For the m-th row-sum of squares,

$$\sum_{l=1}^{m} (A_\chi^{1/2})_{m,l}^2 \;\geq\; \sum_{l=1}^{m} \alpha^{\,l-1} r_{m-l}^2 \;\geq\; \frac{\alpha^m}{\pi} \sum_{l=1}^{m} \frac{1}{l\,\alpha^l} \;\geq\; \frac{\alpha^m}{\pi} \log m. \tag{38}$$

MaxSE. Taking the maximum over m and applying Lemma 16 yields

$$\max_{1\leq m\leq n} \sum_{l=1}^{m} (A_\chi^{1/2})_{m,l}^2 = \Omega\left(\log \frac{n}{\log(1/\beta)} \right), \tag{39}$$

so the maximum row norm is $\Omega\big(\sqrt{\log \frac{n}{\log(1/\beta)}}\big)$. Multiplying by the maximum column norm $\Omega(\log n)$ gives the first bound.



(a) number of participation $k = 4$. **(b)** number of participation $k = 8$.

Figure 2 Multi-participation MeanSE error with matrix size $n = 2048$. Lines are computed for bandwidth $p = 64$. For the exponential workload, we observe that with a larger participation number it becomes beneficial to optimize the factorization with respect to the learning rate decay workload. However, for the considered values of n and β, we do not observe any benefit from incorporating learning rate scheduling for BISR.

MeanSE. Averaging over m and using Lemma 17,

$$\frac{1}{n}\sum_{m=1}^{n}\sum_{l=1}^{m}(A_\chi^{1/2})_{m,l}^2 = \Theta\left(\frac{\log n}{\log(1/\beta)}\right), \tag{40}$$

so the average row norm is $\Omega(\sqrt{\frac{\log n}{\log(1/\beta)}})$. Multiplying by the maximum column norm $\Omega(\log n)$ gives the second bound. ◀

4.2 Multi-participation

Following the line of work on multi-participation matrix factorization [9, 8, 19, 25, 20], we allow each user or datapoint to participate multiple times. Without imposing any restriction on the participation pattern, the guarantees would be no stronger than those obtained via the privacy composition. To overcome this, we adopt the notion of *b-min separation*, which requires that the gap between any two consecutive participations of the same user be at least $b > 0$. Under this condition, each user may participate up to $k = \lceil n/b \rceil$ times. The assumption is practical because, in a centralized setting, one has full control over the participation pattern. In the federated learning setting, if a few users contribute disproportionately, their updates could be ignored for the sake of their privacy until the next contribution from another user. This restriction naturally affects the computation of sensitivity, which we refine as

$$\mathrm{sens}_{k,b}(C) \;=\; \sup_{G\sim G'} \|CG - CG'\|_F, \tag{41}$$

where G and G' differ in the participations of a single user, with the corresponding rows separated by at least b. We then generalize the notion of MeanSE error to the multi-participation setting:

$$\mathcal{E}(B,C) \;=\; \frac{1}{\sqrt{n}}\,\|B\|_F \cdot \mathrm{sens}_{k,b}(C). \tag{42}$$

In this section, we establish both upper and lower bounds on the optimal value $\mathcal{E}(B,C)$ among all factorizations, for the learning-rate workload. This extends the results of [20] on SGD with momentum and weight-decay workloads to the non-Toeplitz case. For the prefix-sum

workload, it was shown that the **Banded Inverse Square Root (BISR)** factorization is asymptotically optimal in the multi-participation setting. The BISR is defined as follows: given a workload matrix A, we compute the square root of its inverse, $C = A^{-1/2}$, band it to width p by nullifying all elements below the p-th diagonal and then invert the result. The corresponding correlation matrix is denoted C^p. Then there exists a unique matrix B^p such that $B^pC^p = A$. By using the BISR matrix corresponding to the prefix-sum workload A_1, we establish a general upper bound in the multi-participation setting for workloads with learning rates A_χ.

▶ **Theorem 11.** *Under the same assumptions on learning rate scheduling χ_t as in Theorem 1, the following holds.*

$$\mathcal{E}(B^p_\chi, C^p_1) = \mathcal{O}\left(\sqrt{\frac{k}{n}\left(\log p + \frac{p}{b}\right)\sum_{m=1}^{n}\left[\chi_m^2 \log\left(\min\{m,p\}\right) + \frac{1}{p}\sum_{t=p}^{m-1}\chi_t^2\right]}\right). \tag{43}$$

For exponential decay the upper bound (after optimizing over p) has the following form:

▶ **Corollary 12.** *Let $\chi_t = \beta^{\frac{t-1}{n-1}}$ with $\beta \in (0, 1/e)$. Then, in multi-participation with b-min-separation and at most $k = \lceil \frac{n}{b} \rceil$ participations, we have for $p^* = O(b \log b)$ the following optimized upper bound:*

$$\mathcal{E}(B^p_\chi, C^p_1) = \mathcal{O}\left(\frac{\sqrt{k}\log n + k}{\sqrt{\log(1/\beta)}}\right). \tag{44}$$

We prove a general lower bound for multi-participation error with arbitrary learning rate scheduling.

▶ **Theorem 13** (Lower bound for multi-participation)**.** *Let $A_\chi = A_1 D_\chi$, where $D_\chi = \operatorname{diag}(\chi_1, \ldots, \chi_n)$ with positive $\chi_t > 0$. Assume any factorization $A_\chi = B \times C$. Then, in multi-participation with b-min-separation and at most $k = \lceil \frac{n}{b} \rceil$ participations, we have*

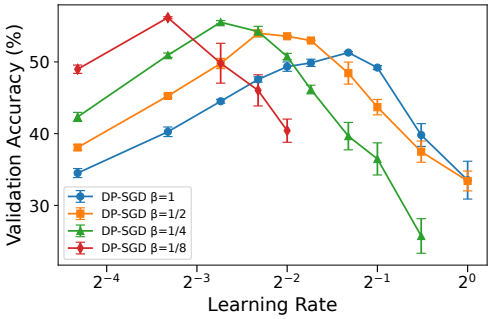
$$\mathcal{E}(B, C) \;\geq\; \max\left\{\max_{t \leq n} \frac{\sqrt{k}\, t\, \chi_t}{\pi\sqrt{2}n}\left(\min_{j \leq t} \chi_j\right)\log(t),\; \sum_{j=0}^{k-1} \chi_{1+jb}\left(1 - \frac{j}{k-1}\right)\right\}. \tag{45}$$

For the exponential learning rate decay we can simplify the lower bound.

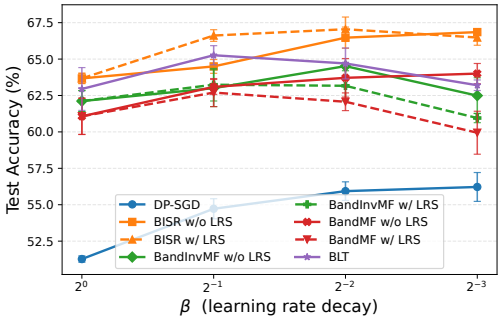
▶ **Corollary 14.** *Let $\chi_k = \beta^{\frac{k-1}{n-1}}$ with $\beta \in (0, 1/e)$. Then Theorem 13 yields*

$$\mathcal{E}(B, C) = \Omega\left(\frac{\sqrt{k}}{\log(1/\beta)} \log \frac{n}{\log(1/\beta)} + \frac{k}{\log(1/\beta)}\right). \tag{46}$$

For the numerical comparison in the multi-participation we study several recently proposed memory-efficient factorizations. Including banded matrix factorization [25], banded inverse factorization BandInvMF and BISR [20] and Buffered Linear Toeplitz (BLT) [26]. We can optimize banded and banded inverse matrices, accounting for the learning rate decay, as well as like if it was a prefix-sum workload with constant learning rate, we refer to this difference as "w/ LRS" and "w/o LRS". See the plots in the Figure 2 for the exponential decay, and Figure 6 in the Appendix for other learning rate schedulers.

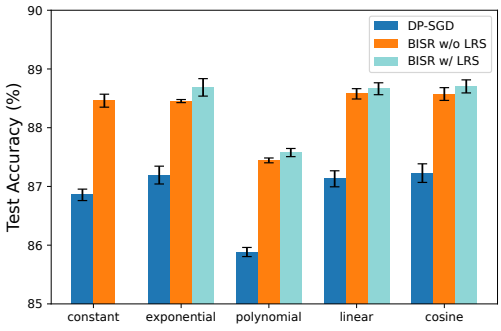


(a) Validation accuracy with DP-SGD.

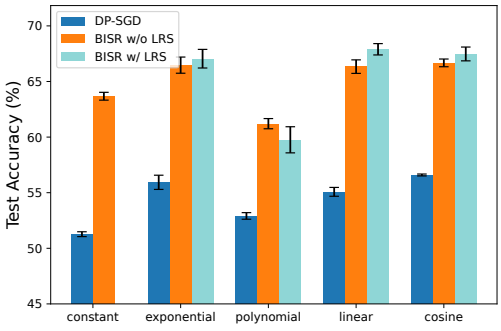


(b) Test accuracy with different matrix factorizations.

Figure 3 CIFAR-10 results under $(9, 10^{-5})$-differential privacy. (a) Validation accuracy with **exponential learning rate scheduling** for different learning rates in DP-SGD. We report the points corresponding to the lowest learning rate; for example, a learning rate of 1/2 for $\beta = 1/4$ indicates that training starts with a learning rate of 2 and decays to 1/2. (b) Test accuracy across different matrix factorizations with **exponential learning rate scheduling**. Training hyperparameters are provided in Table 3.



(a) BERT-base on the IMDB dataset ($\varepsilon = 4$).



(b) CNN on the CIFAR-10 dataset ($\varepsilon = 9$).

Figure 4 Test accuracy of different learning rate schedulers for (a) BERT-base on IMDB and (b) CNN on CIFAR-10 under differential privacy with $\varepsilon = 4$ and $\varepsilon = 9$, respectively. Training hyperparameters are listed in Table 4.

5 Experiments

We demonstrate the practical benefits of learning rate scheduling in Figure 3 on CIFAR-10 dataset. All experiments satisfy $(9, 10^{-5})$-DP and use a 3-block CNN trained for 10 epochs with batch size 128 and clipping norm 1. For privacy accounting, we use Poisson subsampling with PLD accounting [21] for DP-SGD and amplification by Ball-in-Bins subsampling with Monte Carlo accounting [7] for all matrix mechanisms. Subfigure (a) shows validation accuracy across different initial learning rates, where exponential learning rate scheduling improves performance compared to DP-SGD with a fixed learning rate ($\beta = 1$). Subfigure (b) reports test accuracy using the best learning rate chosen on the validation set. All factorizations benefit substantially from scheduling, and the learning-rate–aware factorization (denoted as BISR w/ LRS) achieves even further improvements. However, optimizing the factorization with respect to learning rate workload does not necessarily lead to additional gains: while RMSE can serve as a proxy for performance, it does not perfectly predict it. In practice,

workload optimization increases the added noise per iteration, and this effect is not fully compensated during training due to the non-linearity introduced by large noise. This was also stated as an open problem in a recent survey on matrix factorization [28].

In Figure 4, we compare different learning rate schedulers with a constant one. We observe that learning rate scheduling improves accuracy for DP-SGD in all cases except for the polynomial decay with $\gamma = 2$, which deteriorates performance. The other schedulers substantially improve the accuracy of BISR. Moreover, our proposed learning-rate-aware factorization (BISR w/ LRS) further improves the quality, with the largest improvement for linear LRS, making it a suitable factorization for high-performance private training.

6 Conclusion and Future Directions

Learning rate scheduling has been shown to improve convergence in both private and non-private machine learning. In this work, we combine learning rate scheduling with matrix factorization and propose a learning-rate-aware factorization, which in the case of exponential learning rate decay is theoretically shown to improve the error. Through numerical experiments using the MaxSE and MeanSE metrics, as well as CIFAR-10 model training, we demonstrate its benefits.

We have primarily studied learning rate decay, but similar techniques can be applied to warm-starting, where the learning rate is initially small and then gradually increased. Optimization-based approaches for matrix factorization are generally agnostic to the choice of learning rate scheduling, but adapting our learning-rate-aware factorization to this setting may pose extra challenges.

References

1 Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Conference on Computer and Communications Security (CCS)*, 2016.

2 Joel Daniel Andersson and Rasmus Pagh. Streaming private continual counting via binning. In *Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

3 Borja Balle, Leonard Berrada, Zachary Charles, Christopher A Choquette-Choo, Soham De, Vadym Doroshenko, Dj Dvijotham, Andrew Galen, Arun Ganesh, Sahra Ghalebikesabi, Jamie Hayes, Peter Kairouz, Ryan McKenna, Brendan McMahan, Aneesh Pappu, Natalia Ponomareva, Mikhail Pravilov, Keith Rush, Samuel L Smith, and Robert Stanforth. JAX-Privacy: Algorithms for privacy-preserving machine learning in JAX, 2025. http://github.com/google-deepmind/jax_privacy.

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

5 Aurélien Bellet, Edwige Cyffers, Davide Frey, Romaric Gaudel, Dimitri Lerévérend, and François Taïani. Unified privacy guarantees for decentralized learning via matrix factorization. In *International Conference on Learning Representations (ICLR)*, 2026.

6 Alexander Bienstock, Ujjwal Kumar, and Antigoni Polychroniadou. Dmm: Distributed matrix mechanism for differentially-private federated learning based on constant-overhead linear secret resharing. In *International Conference on Machine Learning (ICML)*, 2025.

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

8 Christopher A Choquette-Choo, Arun Ganesh, Ryan McKenna, H Brendan McMahan, John Rush, Abhradeep Guha Thakurta, and Zheng Xu. (amplified) banded matrix factorization: A unified approach to private training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

9 Christopher A. Choquette-Choo, Hugh Brendan McMahan, J. Keith Rush, and Abhradeep Guha Thakurta. Multi epoch matrix factorization mechanisms for private machine learning. In *International Conference on Machine Learning (ICML)*, 2023.

10 Sergey Denisov, H Brendan McMahan, John Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

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

12 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2006.

13 Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained error bound on differentially private continual observation. In *International Conference on Machine Learning (ICML)*, 2023.

14 Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2017. arXiv:1706.02677. `arXiv:1706.02677`.

15 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

16 M. Henzinger, J. Upadhyay, and S. Upadhyay. A unifying framework for differentially private sums under continual observation. In *Symposium on Discrete Algorithms (SODA)*, 2024.

17 Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. Normalized square root: Sharper matrix factorization bounds for differentially private continual counting. In *Foundations of Responsible Computing (FORC)*, 2026.

18 Monika Henzinger and Jalaj Upadhyay. Improved differentially private continual observation using group algebra. In *Symposium on Discrete Algorithms (SODA)*, 2025.

19 Nikita P Kalinin and Christoph Lampert. Banded square root matrix factorization for differentially private model training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

20 Nikita P Kalinin, Ryan McKenna, Jalaj Upadhyay, and Christoph H Lampert. 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.

21 Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled Gaussian mechanism using FFT. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

22 Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022. `arXiv:2201.12328`.

23 Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

24 Zelun Luo, Daniel J Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

26 Hugh Brendan McMahan, Zheng Xu, and Yanxiang Zhang. A hassle-free algorithm for strong differential privacy in federated learning systems. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

27 Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

28 Krishna Pillutla, Jalaj Upadhyay, Christopher A Choquette-Choo, Krishnamurthy Dvijotham, Arun Ganesh, Monika Henzinger, Jonathan Katz, Ryan McKenna, H Brendan McMahan, Keith Rush, et al. Correlated noise mechanisms for differentially private learning, 2025. arXiv preprint arXiv:2506.08201. doi:10.48550/arXiv.2506.08201.

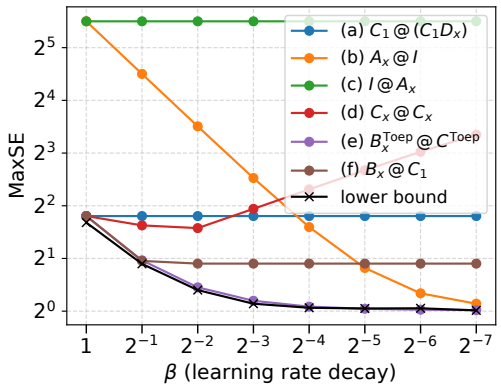
29 Leslie N Smith. Cyclical learning rates for training neural networks. In *Winter Conference on Applications of Computer Vision (WACV)*, 2017.

30 VaultGemma Team. VaultGemma: the world's most capable differentially private LLM, September 2025. Google Research Blog. URL: https://research.google/blog/vaultgemma-the-worlds-most-capable-differentially-private-llm/.

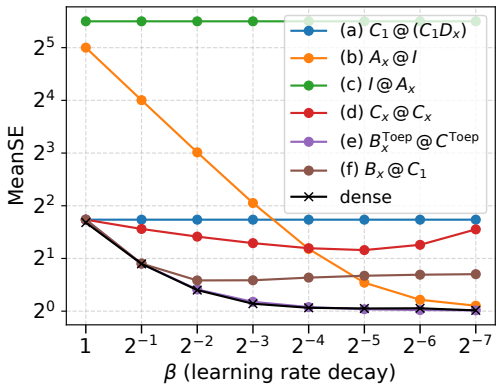
31 Zheng Xu and Yanxiang Zhang. Advances in private training for production on-device language models, 2024. Google Research Blog.

32 Jiaojiao Zhang, Linglingzhi Zhu, Dominik Fay, and Mikael Johansson. Locally differentially private online federated learning with correlated noise. *IEEE Transactions on Signal Processing*, 2025.

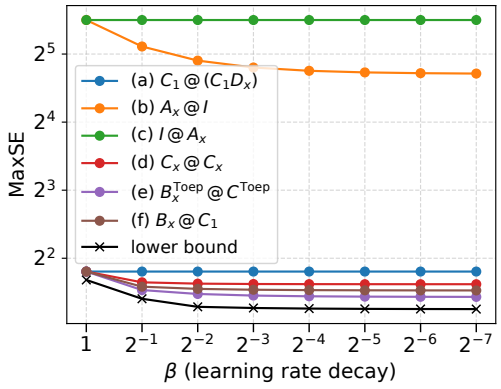
A Additional Experiments



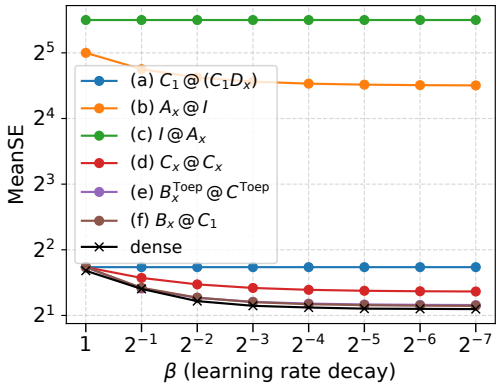
(a) MaxSE, Polynomial $\gamma = 2$.



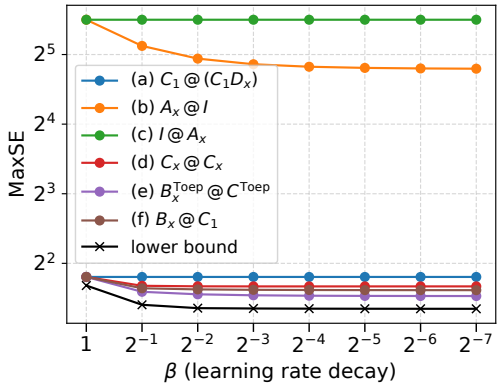
(b) MeanSE, Polynomial $\gamma = 2$.



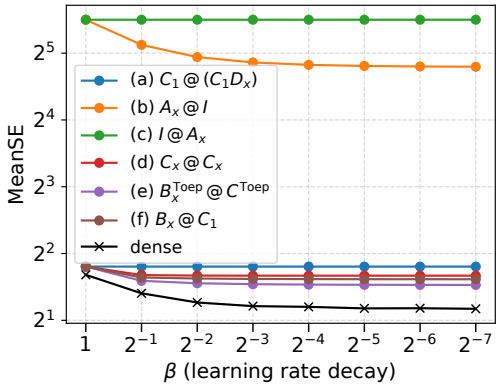
(c) MaxSE, Linear.



(d) MeanSE, Linear.

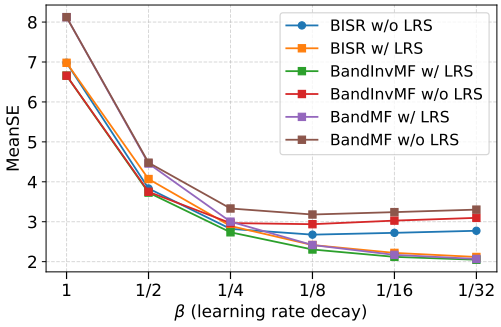


(e) MaxSE, Cosine.

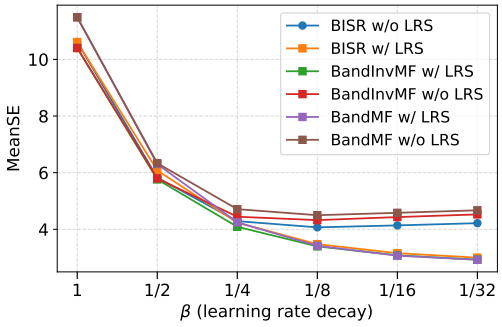


(f) MeanSE, Cosine.

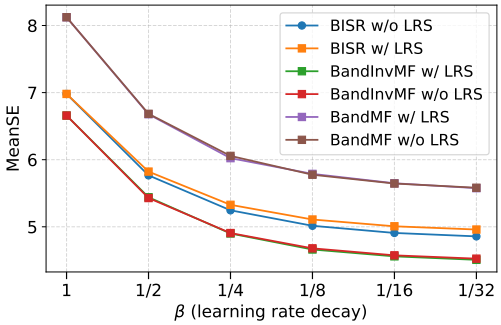
Figure 5 Comparison of different LR schedulers ($n = 2048$) in single participation.



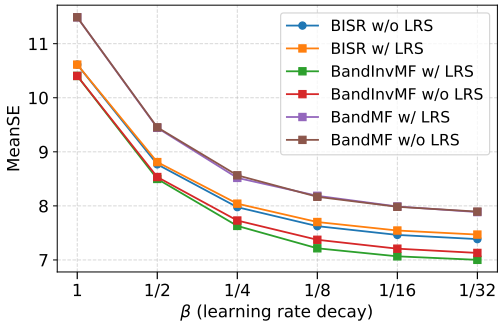
(a) Polynomial $\gamma = 2$, $k = 4$.



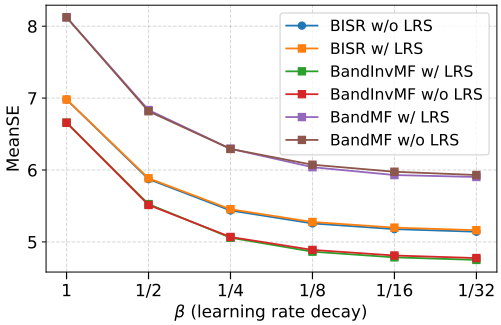
(b) Polynomial $\gamma = 2$, $k = 8$.



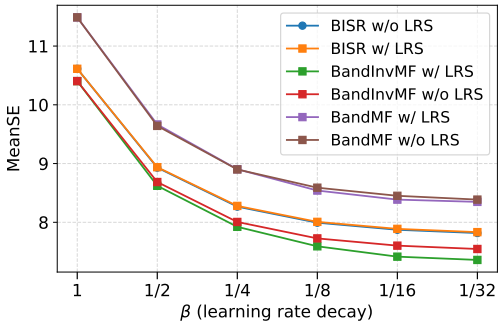
(c) Linear, $k = 4$.



(d) Linear, $k = 8$.



(e) Cosine, $k = 4$.



(f) Cosine, $k = 8$.

Figure 6 Multi-participation MeanSE error under different learning-rate schedulers (Polynomial $\gamma = 2$, Linear, Cosine) for $k = 4$ and $k = 8$. Matrix size $n = 1024$, bandwidth $p = 64$.

Table 3 We train four different methods for matrix optimization: DP-SGD, BISR, BandInvMF, and BandMF. Each factorization method can be computed either with a workload induced by learning rate scheduling (w/ LRS) or with a constant workload corresponding to prefix sums (w/o LRS). All experiments use clipping norm $\zeta = 1$ and batch size 128. For each method, the noise multiplier σ is computed using a privacy accountant: Poisson accounting for DP-SGD and bins-and-balls sampling with an MCMC accountant [7] for the matrix factorization methods. Learning rates η are tuned on a validation set separately for each method and decay setting.

Method				$\beta = 1$		$\beta = \frac{1}{2}$		$\beta = \frac{1}{4}$		$\beta = \frac{1}{8}$	
	ζ	BS	p	η	σ	η	σ	η	σ	η	σ
DP-SGD	1	128	1	0.4	0.479	0.4	0.479	0.6	0.479	0.8	0.479
BISR (w/o LRS)	1	128	64	0.8	1.910	1.6	1.910	1.8	1.910	1.8	1.910
BISR (w/ LRS)	1	128	64	0.8	1.908	1.6	1.901	1.9	1.894	2.0	1.888
BandInvMF (w/o LRS)	1	128	64	1.0	2.597	1.5	2.597	1.6	2.597	1.7	2.597
BandInvMF (w/ LRS)	1	128	64	1.0	2.597	1.5	2.681	1.6	2.814	1.6	2.870
BandMF (w/o LRS)	1	128	64	0.9	2.921	1.5	2.921	1.6	2.921	1.7	2.921
BandMF (w/ LRS)	1	128	64	0.9	2.921	1.1	3.053	1.6	3.158	1.7	3.222
BLT	1	128	64	0.9	2.580	1.3	2.580	1.4	2.580	1.8	2.580

Table 4 Comparison of different learning rate schedulers for training with matrix factorization with fixed learning rate decay $\beta = \frac{1}{4}$. We evaluate DP-SGD, BISR (w/o LRS), and BISR (w/ LRS) under four learning rate decay strategies: exponential, polynomial, linear, and cosine. All experiments use clipping norm $\zeta = 1$ and batch size 128, for BISR we use bandwidth $p = 64$. Learning rates η are tuned on a validation set for each decay setting.

Dataset	Method	ζ	BS	p	Learning rate η by scheduler			
					Exponential	Polynomial	Linear	Cosine
CIFAR-10	DP-SGD	1	128	1	0.6	1.1	0.6	0.5
	BISR (w/o LRS)	1	128	64	1.8	1.8	1.6	1.5
	BISR (w/ LRS)	1	128	64	1.9	1.8	1.6	1.4
IMDB	DP-SGD	1	512	1	0.05	0.05	0.05	0.05
	BISR (w/o LRS)	1	512	64	0.1	0.1	0.1	0.1
	BISR (w/ LRS)	1	512	64	0.1	0.1	0.1	0.1

B Naive Factorizations

In this section we briefly state and prove error bounds for simpler factorization choices.

▶ **Lemma 15.**

(a) $\mathrm{MaxSE}(A_1^{1/2}, A_1^{1/2}D) = \Theta(\log n)$ $\quad$ $\mathrm{MeanSE}(A_1^{1/2}, A_1^{1/2}D) = \Theta(\log n)$

(b) $\mathrm{MaxSE}(A_\chi, I) = \Theta\left(\sqrt{\frac{n}{\log 1/\beta}}\right)$ $\quad$ $\mathrm{MeanSE}(A_\chi, I) = \Theta\left(\sqrt{\frac{n}{\log 1/\beta}}\right)$

(c) $\mathrm{MaxSE}(I, A_\chi) = \Theta(\log n)$ $\quad$ $\mathrm{MeanSE}(I, A_\chi) = \Theta(\log n)$

Proof.

(a) Since $\chi_1 = 1$ and all other $\chi_t \le 1$, the maximum column norm is still achieved in the first column and is exactly the same as that of $A_1^{1/2}$. Thus,

$$\mathrm{MaxSE}(A_1^{1/2}, A_1^{1/2}D) = \mathrm{MaxSE}(A_1^{1/2}, A_1^{1/2}) = \Theta(\log n),$$
$$\mathrm{MeanSE}(A_1^{1/2}, A_1^{1/2}D) = \mathrm{MeanSE}(A_1^{1/2}, A_1^{1/2}) = \Theta(\log n),$$

which follows from the analysis of the prefix-sum square-root factorization by [16].

(b) The maximum column norm of I is 1. The maximum row norm of A_χ is

$$\sqrt{\sum_{k=1}^{n} \chi_k^2} = \sqrt{\sum_{k=0}^{n-1} \beta^{\frac{2k}{n-1}}} = \sqrt{\frac{1-\beta^{\frac{2n}{n-1}}}{1-\beta^{\frac{2}{n-1}}}} = \Theta\left(\sqrt{\tfrac{n}{\log(1/\beta)}}\right). \tag{47}$$

The normalized Frobenius norm $\frac{1}{\sqrt{n}}\|A_\chi\|_F$ is

$$\frac{1}{\sqrt{n}}\|A_\chi\|_F = \frac{1}{\sqrt{n}}\sqrt{\sum_{k=1}^{n}(n+1-k)\chi_k^2} = \frac{1}{\sqrt{n}}\sqrt{\sum_{k=1}^{n}(n+1-k)\beta^{\frac{2(k-1)}{n-1}}} \tag{48}$$

$$= \frac{1}{\sqrt{n}}\sqrt{\sum_{k=0}^{n-1}(n-k)\beta^{\frac{2k}{n-1}}} = \sqrt{\frac{\alpha^{2(n+1)} - \alpha^2(n+1) + n}{n(1-\alpha^2)^2}}, \tag{49}$$

where $\alpha = \beta^{\frac{1}{n-1}}$. Hence $1 - \alpha^2 \sim \frac{2\log(1/\beta)}{n}$ and $\alpha^{2n} \sim \beta^2$, which results in

$$\mathrm{MeanSE}(A_\chi, I) = \frac{1}{\sqrt{n}}\|A_\chi\|_F = \Theta\left(\sqrt{\tfrac{1}{1-\alpha^2}}\right) = \Theta\left(\sqrt{\tfrac{n}{\log(1/\beta)}}\right). \tag{50}$$

(c) The maximum row norm of I is 1, as is its normalized Frobenius norm. The maximum column norm of A_χ is attained in the first column and is exactly $\sqrt{n}$, which concludes the proof. ◀

C Exponential Learning Rate Decay

The following two lemmas are needed to complete the proof of Corollary 8.

▶ **Lemma 16.** *Let $\beta \in (0, 1/e)$ and $\alpha = \beta^{1/(n-1)}$. Then*

$$\max_{1 \le m \le n} \alpha^m \log m = \Theta\left(\log \frac{n}{\log(1/\beta)}\right). \tag{51}$$

Proof. For the lower bound, take $m_0 = \lceil 1/\log(1/\alpha) \rceil$. Since $\log(1/\alpha) = \frac{1}{n-1}\log(1/\beta)$, we have $m_0 \le (n-1)/\log(1/\beta) < n$, so m_0 is admissible. Moreover, $\alpha^{m_0} \ge e^{-1}\alpha$ and $\log m_0 \ge \log \frac{1}{\log(1/\alpha)}$, giving

$$\max_{1\le m\le n} \alpha^m \log m \;\ge\; \Omega\Big(\log \tfrac{1}{\log(1/\alpha)}\Big). \tag{52}$$

For the upper bound, write $f(m) = \alpha^m \log m$ with real $m > 1$. Then $\frac{d}{dm}\log f(m) = \log\alpha + 1/(m\log m)$, so the maximizer satisfies $m\log m = 1/\log(1/\alpha)$. At this point, $\log m \sim \log \frac{1}{\log(1/\alpha)}$ and $\alpha^m = e^{-1/\log m} = \Theta(1)$, hence $f(m) = \mathcal{O}(\log \frac{1}{\log(1/\alpha)})$.

Thus

$$\max_{1\le m\le n} \alpha^m \log m = \Theta\Big(\log \tfrac{1}{\log(1/\alpha)}\Big). \tag{53}$$

Finally, since $\log \frac{1}{\log(1/\alpha)} = \log \frac{n-1}{\log(1/\beta)} = \Theta(\log \frac{n}{\log(1/\beta)})$, the claim follows. ◀

▶ **Lemma 17.** *Let $\beta \in (0, 1/e)$ and $\alpha = \beta^{1/(n-1)}$. Then*

$$\frac{1}{n}\sum_{m=1}^{n} \alpha^m \log m = \Theta\Big(\frac{\log n}{\log(1/\beta)}\Big). \tag{54}$$

Proof. Splitting $\log m = \log n + \log(m/n)$ gives

$$\frac{1}{n}\sum_{m=1}^{n} \alpha^m \log m = \frac{\log n}{n}\sum_{m=1}^{n} \alpha^m + \frac{1}{n}\sum_{m=1}^{n} \alpha^m \log(m/n). \tag{55}$$

The first sum is geometric: $\sum_{m=1}^{n} \alpha^m = \alpha(1-\alpha^n)/(1-\alpha)$. Since $\alpha = 1 - \frac{\log(1/\beta)}{n-1} + o(1/n)$, we have $1-\alpha \sim \frac{\log(1/\beta)}{n-1}$ and $\alpha^n \to \beta$. Thus $\frac{1}{n}\sum_{m=1}^{n} \alpha^m \sim (1-\beta)/\log(1/\beta)$, so the first term is $\sim \frac{1-\beta}{\log(1/\beta)}\log n = \Theta(\frac{\log n}{\log(1/\beta)})$.

The second sum is a Riemann sum, converging to $I(\beta) = \int_0^1 \beta^x \log x\, dx$. Since I is monotone decreasing with $I(0) = 0$, $I(1) = -1$, we have $|I(\beta)| = O(1)$. Hence the first term dominates, and the result follows. ◀