DP-λCGD: Efficient Noise Correlation for Differentially Private Model Training

Anonymous authors
Paper under double-blind review

Abstract

Differentially private stochastic gradient descent (DP-SGD) is the gold standard for training machine learning models with formal differential privacy guarantees. Several recent extensions improve its accuracy by introducing correlated noise across training iterations. Matrix factorization mechanisms are a prominent example, but they can require storing previously added noise vectors, leading to substantial memory overhead. In this work, we propose a new algorithm for correlated-noise private training that correlates noise only with the immediately preceding iteration and cancels a tunable portion of it. By regenerating the previous noise using a pseudorandom number generator rather than storing it, our method has essentially the same memory footprint as DP-SGD. We show that the computational overhead is minimal and empirically demonstrate improved accuracy over DP-SGD.

1 Introduction

Protecting individuals' privacy is a fundamental constraint for algorithms that interact with humans or are trained on personal data. The disclosure of sensitive information can harm both users and the reputation of organizations deploying such algorithms. In machine learning, this challenge is particularly acute as the demand for non-public and user-generated data continues to grow. To mitigate these risks, *Differential Privacy* (Dwork, 2006) has emerged as a mathematically formal framework for privacy protection. By introducing noise into algorithms or model training procedures, it guarantees protection for individuals whose data are used, while still enabling meaningful utility for the resulting models.

The most well-studied approach for introducing differential privacy into model training is differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016). DP-SGD clips per-example gradients to bound the influence of any single data point and then adds independent Gaussian noise to ensure privacy. One of the principal ways to improve the utility of this method is to introduce correlations in the noise across training iterations. Mechanisms that correlate noise in this manner are generally referred to as matrix factorization mechanisms (Li et al., 2015; Kairouz et al., 2021; Denisov et al., 2022; Choquette-Choo et al., 2023b;a; Kalinin & Lampert, 2024; Kalinin & Andersson, 2026). Such mechanisms have been successfully applied in real-world systems, including large-scale deployments such as Google Gboard (Xu & Zhang, 2024).

Matrix factorization mechanisms, however, have inherent limitations. To introduce correlations across iterations, noise from previous steps must be accessible and is therefore typically stored. Because each noise vector has the same dimensionality as the (trainable) model parameters, this can lead to prohibitive memory overhead in some settings. For example, the original work of Choquette-Choo et al. (2023b) on multi-epoch training required storing as many noise vectors as there are iterations. Subsequent work by McKenna (2025) showed that the memory overhead of matrix factorization can be reduced; in particular, in multi-GPU settings with large batch sizes, the time overhead is negligible. However, for larger models trained on fewer GPUs, the overhead is still substantial. More recently, Kalinin et al. (2026) demonstrated that improvements over DP-SGD can be achieved by storing as few as four noise vectors. In contrast, we push this line of work to its logical limit by eliminating the need for any additional memory. Our method leverages noise regeneration: since noise is almost always produced by a pseudo-random generator, previously used noise vectors can be deterministically regenerated when needed rather than stored explicitly.

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.

Another important question concerns how to correlate the noise, or more precisely, which matrix yields the best performance. The most commonly used metric for evaluating the quality of a factorization is the root mean squared error (RMSE) (Denisov et al., 2022; Choquette-Choo et al., 2023b; McKenna, 2025; Kalinin et al., 2026). Maximum squared error (MaxSE) has been proposed as an alternative and is more commonly used in settings such as single-participation or item-level privacy (Henzinger et al., 2023; Henzinger & Upadhyay, 2025; Henzinger et al., 2025; Andersson & Pagh, 2025; Jacobsen & Fawaz, 2025). However, it has been observed that optimizing for these metrics does not perfectly predict the utility of the resulting factorization (Kalinin & Lampert, 2024; Kalinin et al., 2026), and this mismatch is identified as a major open problem in matrix factorization mechanisms in a recent survey (Pillutla et al., 2025).

Our proposed method enables continuous interpolation between a trivial factorization corresponding to standard DP-SGD and factorizations that are optimal with respect to RMSE or MaxSE, controlled by a single hyperparameter. This flexibility allows us to fully explore the space of factorizations within the considered matrix class and to systematically study the problem of optimal matrix factorization. Our results reaffirm prior findings (Koloskova et al., 2023; Choquette-Choo et al., 2024a; Gu et al., 2025) that the optimal factorization is problem- and loss-dependent, and cannot be determined solely from the matrix structure and training configuration. Consequently, we treat the optimal parametrization of our mechanism as a single tunable hyperparameter, which can be lightly adjusted to match the specific learning task.

Contribution We introduce DP-λCGD, a memory-efficient algorithm for correlated noise private training that has essentially the same memory usage as DP-SGD and incurs only small computational overhead. We systematically explore the one-parameter family induced by DP-λCGD and show that RMSE and MaxSE optimal choices of λ do not necessarily maximize downstream accuracy. We theoretically analyze this family, deriving error guarantees and structural properties of RMSE and MaxSE optimal factorizations. Finally, we show empirically that DP-λCGD improves over DP-SGD and is competitive with substantially higher bandwidth matrix factorization methods, while retaining the efficiency of bandwidth 2 noise correlation.

2 Background

We use *differential privacy* (Dwork, 2006) as our privacy notion and recall its definition below.

Definition 1 (Differential Privacy (DP) (Dwork, 2006))**.** A randomized algorithm M is said to provide (ϵ, δ)-differential privacy if, for all neighboring datasets D and D', and for all measurable subsets S of the mechanism's output space:

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S] + \delta. \tag{1}$$

Throughout, by neighboring datasets we mean the standard zero-out notion used for matrix mechanisms, where one data point is replaced by a null element.

To ensure that a model is differentially private, the most common approach is to use differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016). At each iteration t, a batch of data B_t is sampled, and gradients g_i^t are computed for all $i \in [1, |B_t|]$. These gradients are then clipped to a fixed ℓ_2-norm bound $\zeta > 0$ and averaged to obtain the clipped gradient $\hat{g}_t$. Finally, scaled Gaussian noise z_t is added to ensure that the resulting update satisfies a prescribed level of privacy.

The utility of this mechanism can be improved by correlating the noise added across iterations. Intuitively, one may add larger noise at each iteration while choosing the correlation so that components of previously added noise cancel over time. As a result, the accumulated noise can have smaller variance, improving model accuracy. A formal way to introduce linear noise correlation is via a *correlation matrix*[1] C^{-1} (Li et al., 2015; Denisov et al., 2022; Choquette-Choo et al., 2023b). If we stack all gradient vectors into a matrix G and all noise vectors into a matrix Z, then instead of adding Z directly to G, we add correlated noise $C^{-1}Z$ and obtain $G + C^{-1}Z$. By post-processing invariance, this is equivalent to adding noise Z to the transformed gradients CG. Ignoring privacy

[1]We refer to the *correlation* matrix as C^{-1}, and to C, which determines the noise scale, as the *strategy* matrix.

amplification due to subsampling for now, the amount of noise required by a matrix factorization mechanism is proportional to the *sensitivity* of CG:

$$\text{sens}(C) \;=\; \sup_{G \sim G'} \|CG - CG'\|_F, \tag{2}$$

where G and G' differ in all positions corresponding to the participation of a single data point.

Following the framework of (Choquette-Choo et al., 2023a), we consider a multi-epoch training setting in which each data point can participate up to k times, under the restriction that the time gap between consecutive participations is at least b. This framework can be directly transferred to the federated learning setting, where each user participates at most k times, and we do not include a user's input unless we receive sufficient input from the rest of the users.

For the remainder of this work, we restrict attention to positive lower triangular Toeplitz (LLT) strategy matrices. Under the above b-min-separation condition, the sensitivity of such a matrix can be computed using the following result.

Theorem 1 (Theorem 2 from Kalinin & Lampert (2024))**.** *Let C be a lower triangular Toeplitz matrix with decreasing non-negative entries $c_0 \geq c_1 \geq \cdots \geq c_{n-1} \geq 0$. Then, under b-min-separation, the sensitivity is given by*

$$\text{sens}_{k,b}(C) = \Big\| \sum_{j=0}^{k-1} C_{[:,\, jb+1]} \Big\|_2, \tag{3}$$

where $C_{[:,\, jb+1]}$ denotes the $(1 + jb)$-th column of C.

To achieve a target privacy level (ϵ, δ), we add Gaussian noise whose scale is proportional to the sensitivity, multiplied by the Gaussian mechanism noise multiplier (the multiplier can be computed numerically; see Balle & Wang (2018)). With this calibration, using correlated noise induced by C attains the desired privacy guarantee. The main remaining question is therefore how to choose the correlation (or strategy) matrix C. This turns out to be a subtle and decidedly nontrivial problem, and it is the core focus of the matrix factorization literature; see the recent survey of Pillutla et al. (2025).

As an efficiently computable proxy for solution quality, we use the Root Mean Squared Error (RMSE) and the Maximum Squared Error (MaxSE) between the true and noisy sequences of intermediate models. Let $G \in \mathbb{R}^{n \times d}$ be the matrix whose t-th row is the (vectorized) gradient at iteration t. Let $A \in \mathbb{R}^{n \times n}$ be the *prefix-sum* matrix, a lower triangular matrix defined by $A_{ij} = \mathbf{1}\{i \geq j\}$. Then AG stacks the cumulative sums of gradients, and therefore corresponds (up to the learning-rate factor and initialization) to the sequence of intermediate model updates.

Rather than estimating AG directly, we form a noisy estimate by first perturbing the per-iteration gradients with correlated noise and then taking prefix sums:

$$\widehat{AG} = A(G + C^{-1}Z) = AG + AC^{-1}Z, \tag{4}$$

where $Z \in \mathbb{R}^{n \times d}$ has i.i.d. standard Gaussian entries.

Natural measures of estimation quality include the expected squared Frobenius error $\mathbb{E}_Z\big[\|AC^{-1}Z\|_F^2\big]$ and the worst-case (over iterations) expected squared row error $\max_{i \in [n]} \mathbb{E}_Z\big[\|(AC^{-1}Z)_{i,:}\|_2^2\big]$. Let the left factor be $B := AC^{-1}$. For each error measure, we drop multiplicative constants that are independent of the choice of factorization (B, C) (e.g., dimensionality d, clipping norm ζ, noise multiplier $\sigma_{\epsilon,\delta}$), and work with the following equivalent objectives:

$$\text{RMSE}(B, C) \;:=\; \tfrac{1}{\sqrt{n}} \|B\|_F \cdot \text{sens}_{k,b}(C), \qquad \text{MaxSE}(B, C) \;:=\; \|B\|_{2\to\infty} \cdot \text{sens}_{k,b}(C), \tag{5}$$

where $\|B\|_{2\to\infty} := \max_{i \in [n]} \|B_{i,:}\|_2$.

Matrix Factorization mechanisms, as well as DP-SGD, can benefit from privacy amplification. The idea is that each training batch is selected at random (via shuffling or a more intricate scheme), creating uncertainty for an adversary about whether a particular data point was included in a given batch. With appropriate analysis, via so called privacy accountants, this ambiguity allows one to reduce the amount of noise added to gradients while maintaining the same privacy guarantees.

In this work, we use Balls in Bins subsampling and the corresponding accountant for matrix factorization (Choquette-Choo et al., 2024b). In this subsampling scheme, one first fixes the number of batches in an epoch and then randomly allocates each data point to one of the batches. The resulting allocation is reused across epochs, providing computational efficiency. See Subsection 5.1 for more details on amplification.

3 Differentially Private Correlated Gradient Descent (DP-λCGD)

In this paper, we propose DP-λCGD, a matrix factorization and noise-correlation technique that improves upon the utility of DP-SGD while remaining nearly as fast, requiring no additional memory. Formally, we correlate the noise using a lower-triangular Toeplitz strategy matrix C_λ and its inverse (the corresponding correlation matrix) C_λ^{-1}, defined entrywise as follows. For $i, j \in [n]$,

$$(C_\lambda)_{ij} := \begin{cases} \lambda^{\,i-j}, & i \geq j, \\ 0, & i < j, \end{cases} \qquad (C_\lambda^{-1})_{ij} := \begin{cases} 1, & i = j, \\ -\lambda, & i = j+1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Correlating the noise using C_λ^{-1} is equivalent to subtracting a λ-fraction of the noise added at the previous iteration (up to an overall scaling determined by $\mathrm{sens}_{k,b}(C_\lambda)$).

To make our method memory-free, we observe that at each step we only need the noise vector added in the previous iteration. As in essentially all practical private learning pipelines, this noise is generated by a pseudo-random number generator (PRNG). Thus, by storing the PRNG state from the previous iteration, we can roll back to that state, regenerate the previous noise for cancellation, and then generate fresh noise exactly as DP-SGD would. The drawback is that we must regenerate an additional noise vector each iteration, which could introduce overhead. Empirically, however, this cost is negligible compared to the rest of training (e.g., gradient computation): across multiple settings and model families, DP-λCGD runs within 2% of the runtime of DP-SGD for the larger models in our evaluation. See Section 4 for details.

Correlated noise can be combined with privacy amplification by subsampling. We use the Balls-in-Bins subsampling scheme and its corresponding accountant (Choquette-Choo et al., 2024b) to compute the resulting privacy guarantees. This allows improved utility at the same target privacy level. For further discussion of the role of subsampling, see Subsection 5.1.

Putting these ideas together, we outline the full procedure in Algorithm 1. The algorithm takes $\lambda \in [0, 1)$ as a hyperparameter; we discuss how to choose λ in Section 5.

We establish several properties of the DP-λCGD with respect to the RMSE and MaxSE error metrics. We begin by analyzing the MaxSE error, which is characterized in the following lemma.

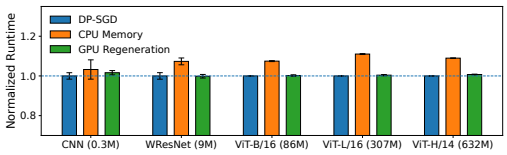
Lemma 2. *For $n = kb$, the MaxSE and RMSE of DP-λCGD, optimized over $\lambda \in [0, 1)$, satisfy*

$$\begin{aligned} \inf_{\lambda \in [0,1)} \mathrm{MaxSE}(AC_\lambda^{-1}, C_\lambda) &= \Theta\big(k + \sqrt{k}\, n^{1/4}\big), \\ \inf_{\lambda \in [0,1)} \mathrm{RMSE}(AC_\lambda^{-1}, C_\lambda) &= \Theta\big(k + \sqrt{k}\, n^{1/4}\big). \end{aligned} \tag{7}$$

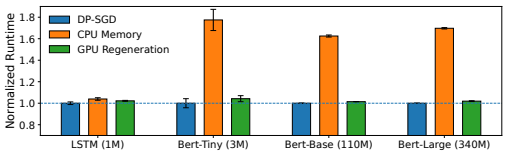
The $\Theta(n^{1/4})$ term for the single-participation setting, as well as the optimal factorization error $\Theta(k + \sqrt{k} \log n)$, were derived in Kalinin et al. (2026). Here, we extend the factorization result to the multi-participation regime. Our bound improves over the trivial (i.e., DP-SGD) factorization, which yields error $O(\sqrt{nk})$, though it remains weaker than the optimal factorization error. Despite this, we will show that improved RMSE does not always translate into better model performance. In particular, in certain regimes, DP-λCGD outperforms factorizations with smaller RMSE.

The DP-λCGD is the first framework that allows us to establish structural properties of optimal factorizations. In particular, we show that the minimizer of the MaxSE objective is always larger than the minimizer of the RMSE objective, which makes RMSE a more practically relevant metric. We discuss this phenomenon in detail in Section 5.

Lemma 3. *For DP-λCGD, a minimizer of MaxSE is always at least as large as a minimizer of RMSE, both with and without amplification by subsampling.*



(a) CIFAR-10 runtime for one epoch.



(b) IMDB fine-tuning runtime for one epoch.

Figure 1: Runtime for one epoch, comparing DP-SGD runtime using the Opacus implementation versus storing the noise on the CPU and regenerating it on the GPU using pseudorandomness.

Relation of C_λ matrix to prior works. We note that the correlation matrix C_λ has appeared as an exemplary factorization in several prior works. It is referred to as "Anti-PGD + damping" in Choquette-Choo et al. (2024a), and it is used to illustrate noise correlation in the recent survey of Pillutla et al. (2025). It can also be viewed as a special case of the BandInvMF factorization of Kalinin et al. (2026). However, it has not previously been used for DP training, λ has not been treated as a tunable hyperparameter, and its computational efficiency in terms of time and memory has not been evaluated.

4 Noise Regeneration

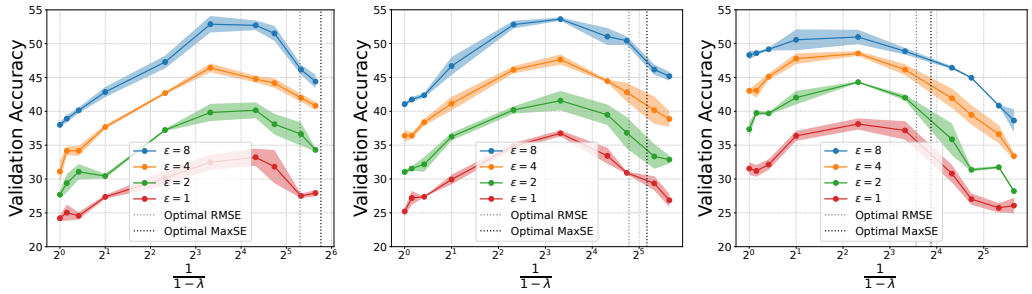
Modern private machine learning uses pseudo-random number generators (PRNGs) to generate noise. A PRNG has a small *state* that determines its next output as well as its next state. The state is typically initialized to a random "seed" value using hardware-supported random number generation, and the whole sequence of random values is determined by the initial state. So-called *counter-based* PRNGs have a fixed state given by a seed s and take a counter i as input to generate the ith random value (that is, it implements a hash function from counters to random values). PyTorch uses the counter-based `Philox4x32-10` pseudorandom number generator Salmon et al. (2011) on CUDA devices, which deterministically maps a 64-bit seed and a 128-bit counter to random values, enabling reproducible and scalable parallel random number generation on GPUs.

Regeneration. Previous approaches to noise correlation required storing noise vectors generated in earlier steps, which can require substantial memory that may not be available on the GPU. Alternatively, one can store the noise on the CPU and transfer it to the GPU, but this incurs substantial overhead. By storing a state of the PRNG it is possible to go back and "replay" any part of the pseudorandom sequence, trading the space for the time needed to regenerate randomness. This approach was believed to be impractical because previous factorizations had non-banded correlation matrices and required regenerating the entire noise sequence; see the discussion in Kim et al. (2026). We also note that the reference code of Choquette-Choo et al. (2023b) implements such "on-the-fly" noise.[2] However, their implementation reconstructs the entire noise sequence from the first iteration at each step, which they described as computationally expensive.

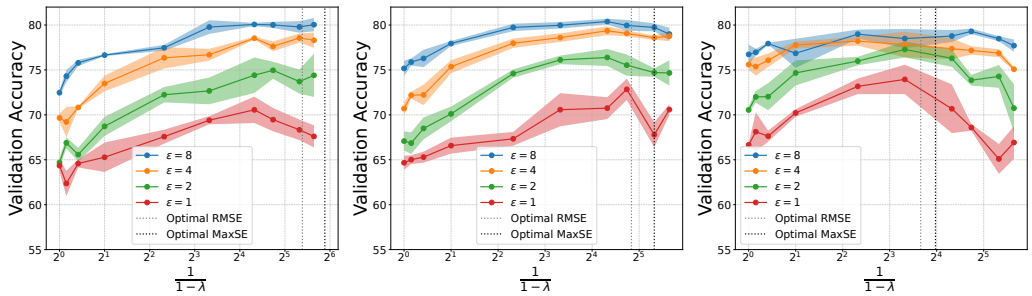
Our proposed DP-λCGD, as well as Banded Inverse Matrix Factorization and Banded Inverse Square Root with small bandwidth (Kalinin et al., 2026), needs to access only a small number of the most recent noise vectors. This makes it efficient to regenerate noise on the fly without storing the history. Note that banded methods such as Banded Matrix Factorization (McKenna, 2025) and Buffered Linear Toeplitz (Dvijotham et al., 2024; McMahan et al., 2024) can correlate noise using a relatively small in-memory buffer; however, at each step the correlated noise depends on components originating from the very first iteration. Regenerating the full sequence at every step would be impractical (see Table 1 for the time and space requirements).

Discussion. The random values generated by Philox and other popular high-performance PRNGs used in machine learning frameworks pass statistical tests of randomness but are not cryptographically secure. This potentially weakens the privacy protection, but since the de-facto standard is to not use slower, cryptographically secure PRNGs Egan (2024) we will follow this standard. Using

[2] `https://github.com/google-research/federated/blob/master/dp_matrix_factorization/matrix_factorization_query.py#L190`



(a) CIFAR-10: $B = 128$, $k = 10$ (b) CIFAR-10: $B = 256$, $k = 10$ (c) CIFAR-10: $B = 1024$, $k = 10$



(d) IMDB: $B = 64$, $k = 10$ (e) IMDB: $B = 128$, $k = 10$ (f) IMDB: $B = 512$, $k = 10$

Figure 2: Validation accuracy for different values of the parameter λ in DP-λCGD with different privacy budgets ϵ and $\delta = 10^{-5}$. The first row shows a CNN model trained on CIFAR-10 for $k = 10$ epochs, using different batch sizes B. The second row shows fine-tuning of the BERT-tiny model on the IMDB sentiment analysis dataset.

slower PRNGs may shift the trade-off between the cost of storing noise values and regenerating them.

Evaluation. We evaluate GPU-based noise regeneration against DP-SGD and an alternative that stores and correlates noise on the CPU. In Fig. 1, we report the relative time per epoch on CIFAR-10 with batch size 512, for models ranging from a 0.3M-parameter CNN to ViT-H/14 with 632M parameters (Dosovitskiy et al., 2021). We also evaluate on IMDB using an LSTM and a family of BERT models (Devlin et al., 2019), with batch size 128. For larger models, GPU regeneration adds less than 1% overhead on CIFAR-10 and less than 2% for BERT-Base and BERT-Large on IMDB. The overhead is slightly higher for faster runs, reaching 1.6% for the CNN and 4.2% for BERT-Tiny. In contrast, storing noise on the CPU and transferring it to the GPU incurs substantially larger overhead, reaching 11% on CIFAR-10 and 78% on IMDB.

In Tables 3 and 4 in the appendix, we report the absolute runtime for different bandwidths, where bandwidth 2 corresponds to DP-λCGD and larger bandwidths correspond to Banded Inverse Matrix Factorization (Kalinin et al., 2026). For bandwidth 4, GPU regeneration adds at most about 9% overhead across all models. For bandwidth 16, the overhead remains below 11% except for BERT-Tiny, where it reaches 45%. In contrast, storing and transferring noise on the CPU can increase runtime by up to 147%.

5 How to choose the value λ?

The best value of λ for DP-λCGD can depend on the batch size, the number of epochs, the target privacy level, whether amplification by subsampling is used, and, ultimately, the learning problem (e.g., the loss function and data distribution). To study how to choose this parameter in practice, we report results of an extensive sweep over λ in Figure 2.

Figure 2 reports results *without* amplification, which allows us to characterize the behavior of RMSE- and MaxSE-optimal factorizations theoretically; for the role of subsampling, see Subsection 5.1. We make three empirical observations.

1. The best choice of λ appears to be task-dependent: under similar training settings and identical matrices, the optimal λ differs substantially between training a CNN on CIFAR-10 and fine-tuning BERT-tiny on IMDB. In particular, for CIFAR-10 the optimal λ is much smaller than the value suggested by the RMSE-optimal factorization. Moreover, as shown in Lemma 3, the minimizer of the MaxSE objective is always larger than the minimizer of the RMSE objective. Consequently, MaxSE is farther from the optimum and is therefore a less suitable objective for DP-λCGD.
2. As the batch size increases, the RMSE-optimal factorization approaches DP-SGD. However, Figure 5 in the appendix shows that, for a fixed non-full batch size, increasing the number of epochs pushes the optimal mechanisms further away from DP-SGD (e.g. the optimal λ increases). This suggests that even in long training runs, common in high-performance DP-SGD training, the benefit of matrix factorization, and DP-λCGD in particular, continues to grow.
3. The optimal λ is relatively insensitive to the privacy level across the range we tested.

The full-batch regime. We next provide theoretical support for the observed dependence of the optimal λ on the batch size. We consider the full-batch regime, corresponding to $b = 1$ and $k = n$, and study the multi-participation error in this limit. The optimal-factorization results of Kalinin et al. (2026) imply that DP-SGD, corresponding to the trivial factorization, is asymptotically optimal in this regime, with RMSE scaling as $O(n)$. This asymptotic result, however, does not determine whether DP-SGD is optimal at the level of the leading constant.

For the MaxSE metric, this question has already been resolved: Pillutla et al. (2025, Proposition 3.11) show that DP-SGD is constant-optimal in the full-batch regime. Their argument, however, does not extend to the RMSE objective. We therefore ask whether the same conclusion holds for RMSE.

The following theorem answers this question in the negative: DP-SGD is **not** constant-optimal under the RMSE metric. Instead, the weighted diagonal factorization with weights $(n-j+1)^{1/4}$ achieves a smaller leading constant, reducing the error by approximately 6% compared with DP-SGD. In fact, this factorization is optimal[3] among all factorizations, as shown in the theorem below.

Theorem 4. *Let $b = 1$ and $k = n$, and let $A \in \mathbb{R}^{n\times n}$ be the prefix-sum matrix. Define the diagonal factorization $C_{\mathrm{diag}} = \mathrm{diag}\big((n-j+1)^{1/4}\big)_{j=1}^{n}$ and $B_{\mathrm{diag}} = AC_{\mathrm{diag}}^{-1}$. Then, for any factorization $A = BC$, the following lower bound holds:*

$$\mathrm{RMSE}(B, C) \geq \mathrm{RMSE}(B_{\mathrm{diag}}, C_{\mathrm{diag}}) = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \sqrt{j} \sim \frac{2n}{3}. \tag{8}$$

Moreover, if the matrix C is lower triangular with positive diagonal, then equality holds if and only if $C = \alpha C_{\mathrm{diag}}$ for some $\alpha > 0$.

While DP-SGD is not constant optimal under the RMSE metric across all factorizations, the following result shows that it is also RMSE-optimal within the class of DP-λCGD factorizations.

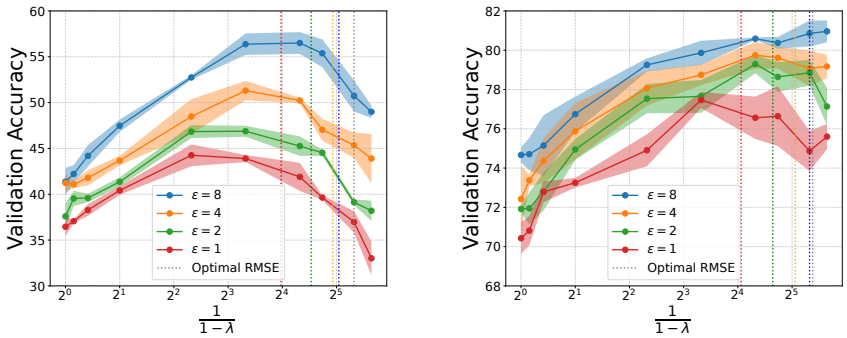
Corollary 5. *The trivial factorization $B = A$, $C = I$ is optimal for DP-λCGD under the RMSE and MaxSE metrics for $b = 1$ and $k = n$.*

We show that in the limit of having a full batch so $b = 1$, DP-SGD ($\lambda = 0$) is provably optimal, suggesting that as the batch size increases, the optimal λ decreases as observed in Figure 2.

5.1 Amplification

The most common amplification scheme for DP-SGD is Poisson subsampling (Abadi et al., 2016; Koskela et al., 2020; Zhu et al., 2022), in which each batch is formed independently and each element is included with a probability determined by the expected batch size. This method provides the strongest known privacy guarantees among sampling schemes. However, it has notable limitations.

[3]This improvement is obtained by injecting larger noise into later estimates, which may be undesirable in practice (see Remark 4.10 from Pillutla et al. (2025)) since model training typically prioritizes higher accuracy for the final model.



CIFAR-10 $B = 128$, $k = 10$. IMDb $B = 64$, $k = 10$.

Figure 3: Validation accuracy of DP-λCGD with Balls-in-Bins amplification by subsampling for different values of λ. Vertical lines indicate the optimal values based on the amplified RMSE, suggesting that Balls-in-Bins prefers smaller values of λ. However, the value of λ that is optimal for accuracy is even lower, at least for CIFAR-10 training.

In expectation, nearly one third of the data (specifically, a fraction $1/e$) does not participate in each epoch. Moreover, Poisson subsampling is difficult to implement efficiently at scale, since forming each batch would require scanning the entire dataset and independently deciding whether to include each element. This leads to the common but undesirable practice of training with shuffling while reporting privacy guarantees under Poisson subsampling (Lebeda et al., 2025; Beltran et al., 2024).

To address these issues, an alternative sampling scheme was proposed, known as Balls in Bins (BnB) (Chua et al., 2025; Choquette-Choo et al., 2024b; Schuchardt & Kalinin, 2026) or random allocation (Feldman & Shenfeld, 2025; 2026). In this approach, one first specifies the number of batches in an epoch and then assigns each data point to a random batch. This guarantees that every sample participates once per epoch, and by preprocessing the dataset, one can efficiently construct batches for multi epoch training.

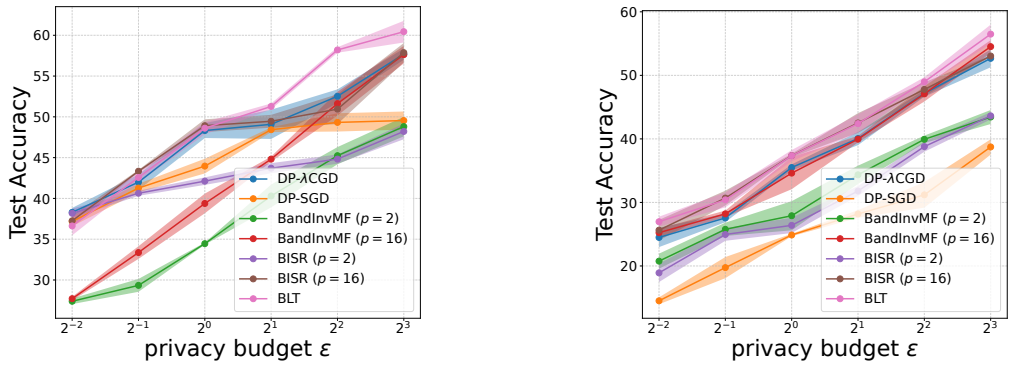
Balls-in-Bins amplification by subsampling can also be applied to matrix factorization (Choquette-Choo et al., 2024b) for any correlation matrix. We use it for DP-λCGD, as it is also the only currently available amplification scheme that is applicable to banded inverse matrices. Choquette-Choo et al. (2024b) compute the privacy level using Monte Carlo sampling. This procedure is effective in practice but acts as a black box and does not allow us to argue theoretically about its properties. We therefore study the effect of amplification numerically. Our experiments show that amplification by Balls in Bins subsampling prefers smaller values of λ than those that would be chosen based solely on RMSE. In particular, the values that minimize the amplification-aware RMSE: $\frac{1}{\sqrt{n}}\|B\|_F \cdot \sigma_{\epsilon,\delta}(C)$ are closer to the optimal factorization for DP-λCGD than those obtained using RMSE alone. See Figure 3 for an illustration.

Practical considerations for the choice of λ. Our main practical observation is that, in our experiments, the optimal value of λ is consistently smaller than the value suggested by amplified RMSE. The latter is itself smaller than the value suggested by RMSE, which is provably smaller than the value suggested by MaxSE. Thus, a practical guideline is to choose a value of λ no larger than the one minimizing amplified RMSE: in our experiments, the optimal choice is typically 2–4$\times$ smaller, measured on the scale $\frac{1}{1-\lambda}$, than the value suggested by amplified RMSE.

We also note that, when optimizing over λ is computationally expensive, simply using the value that minimizes amplified RMSE already yields most of the benefit in the amplified setting.

6 Experiments

We compare DP-λCGD, amplified via Balls-in-Bins subsampling, to other memory-efficient alternatives, including DP-SGD with Poisson subsampling (a commonly used amplification that empirically yields a slightly lower noise multiplier) and the banded inverse factorizations Banded Inverse Matrix Factorization (BandInvMF) and Banded Inverse Square Root (BISR) (Kalinin et al., 2026)



(a) With amplification by subsampling.

(b) Without amplification by subsampling.

Figure 4: Test accuracy on CIFAR-10 across privacy budgets ($B = 128$, $k = 10$; error bars over three runs), comparing DP-SGD with DP-λCGD and other memory-efficient matrix-factorization baselines. Hyperparameters for training with and without amplification by subsampling are reported in Tables 5 and 6, respectively.

with bandwidths $2, 4, 16$. We also include Buffered Linear Toeplitz (BLT) factorization (Dvijotham et al., 2024); although BLT cannot be run efficiently without additional memory, we find it to be a competitive low-memory approach, requiring to store a small buffer of $4 - 5$ vectors. In Table 2, we compare the methods using the RMSE metric, with and without amplification. In the table, we also report banded factorizations such as Banded Matrix Factorization (BandMF) (McKenna, 2025) and Banded Square Root (BSR) (Kalinin & Lampert, 2024); however, they do not perform competitively at small bandwidths (2, 4, or even 16).

Figure 4 reports CNN test accuracy on CIFAR-10, both with and without privacy amplification (see Tables 5 and 6 for hyperparameters). In the amplified setting, we use Poisson subsampling for DP-SGD, which provides stronger amplification than BnB and therefore makes DP-SGD a more competitive baseline. DP-λCGD outperforms DP-SGD across the evaluated privacy budgets, including $\epsilon = 1/4, \delta = 10^{-5}$, and performs on par with BISR with bandwidth 16 while using bandwidth 2, reducing noise-regeneration work by $8\times$. BLT performs better in parts of the low-privacy regime, but requires either storing a noise buffer or regenerating the full sequence. Overall, DP-λCGD provides a strong accuracy–efficiency trade-off without storing past noise vectors. We also provide large scale experiments on full model fine-tuning of BERT-Base on IMDb and fine-tuning two blocks of ViT-Base on CIFAR-100. The results are reported in Figure 6.

Discussion

We proposed DP-λCGD, a time and memory efficient algorithm for introducing noise correlation in differentially private model training. Our experiments show that DP-λCGD improves upon the utility of DP-SGD and is competitive with other memory-efficient baselines, while running at nearly the same speed as DP-SGD. From the perspective of noise correlation, DP-λCGD is a particularly simple nontrivial factorization, making it a useful setting for isolating phenomena that are difficult to study in more general classes of factorizations. In particular, prior work has observed that the RMSE-optimal factorization does not necessarily yield the best downstream performance. The single parameter structure of DP-λCGD allows us to study this mismatch directly by interpolating between DP-SGD and the RMSE-optimal factorization. A promising direction for future work is to develop a similar interpolation for larger bandwidths, which could have smaller RMSE while preserving the time and memory efficiency of noise regeneration.

AI use statement

In this work, we used generative AI tools for code assistance, for language editing, and to help draft the proof of Theorem 4. We did not use generative AI tools for experiment design, for proofs of theoretical statements other than Theorem 4, or for developing research ideas. We have reviewed all

AI-assisted work. In particular, all proofs were manually written and verified by the authors, and all AI-assisted code was carefully reviewed by the authors.

We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

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

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.

Sebastian Rodriguez Beltran, Marlon Tobaben, Joonas Jälkö, Niki Loppi, and Antti Honkela. Towards efficient and scalable training of differentially private deep learning, 2024. arXiv preprint arXiv:2406.17298.

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)*, 2023a.

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

Christopher A Choquette-Choo, Krishnamurthy Dvijotham, Krishna Pillutla, Arun Ganesh, Thomas Steinke, and Abhradeep Thakurta. Correlated noise provably beats independent noise for differentially private learning. In *International Conference on Learning Representations (ICLR)*, 2024a.

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)*, 2024b.

Lynn Chua, Badih Ghazi, Charlie Harrison, Ethan Leeman, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. Balls-and-Bins sampling for DP-SGD. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American chapter of the association for computational linguistics (NAACL)*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Krishnamurthy 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.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming (ICALP)*, 2006.

Shannon Egan. High-speed random number generator co-processors for machine learning and AI acceleration. In *NeurIPS 2024 Workshop Machine Learning with new Compute Paradigms*, 2024.

Vitaly Feldman and Moshe Shenfeld. Privacy amplification by random allocation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Vitaly Feldman and Moshe Shenfeld. Efficient privacy loss accounting for subsampling and random allocation, 2026. arXiv preprint arXiv:2602.17284.

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

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

Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. Almost tight error bounds on differentially private continual counting. In *Symposium on Discrete Algorithms (SODA)*, 2023.

Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. Normalized square root: Sharper matrix factorization bounds for differentially private continual counting, 2025. arXiv preprint arXiv:2509.14334.

Ben Jacobsen and Kassem Fawaz. Private continual counting of unbounded streams. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Peter Kairouz, 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 (ICML)*, 2021.

Nikita P. Kalinin and Joel Daniel Andersson. Learning rate scheduling with matrix factorization for private training. In *Foundations of Responsible Computing (FORC)*, 2026.

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

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.

Donghwan Kim, Xin Gu, Jinho Baek, Timothy Lo, Younghoon Min, Kwangsik Shin, Jongryool Kim, Jongse Park, and Kiwan Maeng. Cocoon: A system architecture for differentially private training with correlated noises. In *Symposium on Operating Systems Design and Implementation (OSDI)*, 2026.

Anastasiia Koloskova, Ryan McKenna, Zachary Charles, John Rush, and H. Brendan McMahan. Gradient descent with linearly correlated noise: Theory and applications to differential privacy. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

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

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

Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *International Conference on Very Large Data Bases (VLDB)*, 2015.

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

H Brendan McMahan, Zheng Xu, and Yanxiang Zhang. A hassle-free algorithm for private learning in practice: Don't use tree aggregation, use BLTs. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2024.

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.

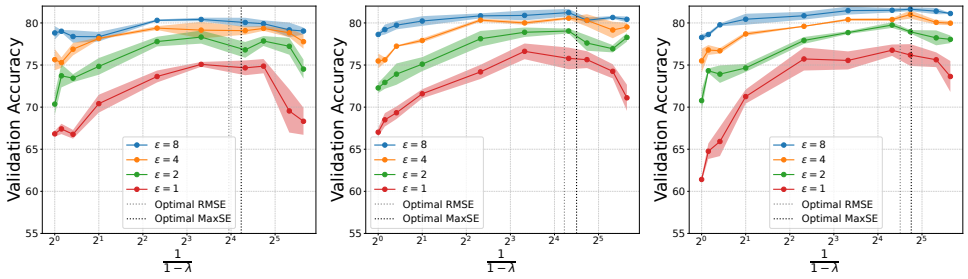
John K Salmon, Mark A Moraes, Ron O Dror, and David E Shaw. Parallel random numbers: as easy as 1, 2, 3. In *Proceedings of international conference for high performance computing, networking, storage and analysis (SC)*, 2011.

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

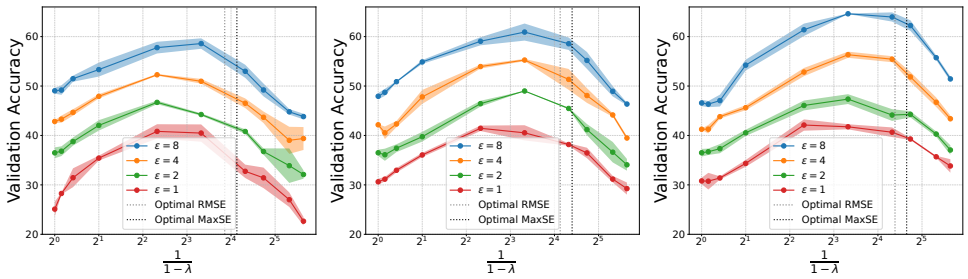
Zheng Xu and Yanxiang Zhang. Advances in private training for production on-device language models, 2024. Google Research Blog.

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

A Additional Experiments

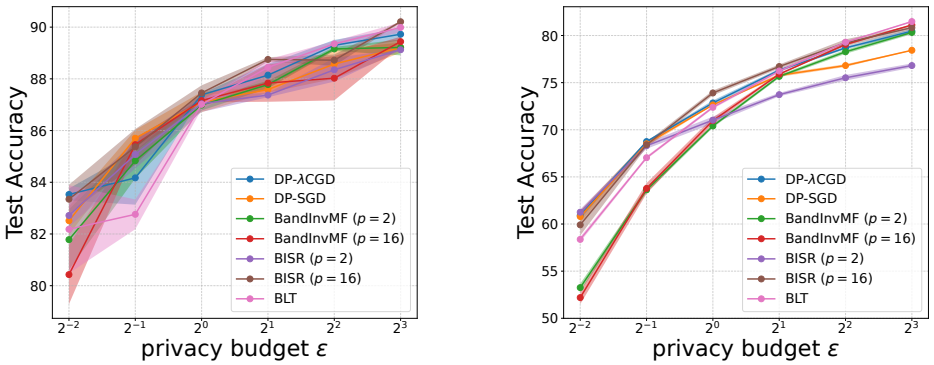


(a) IMDB: $k = 20$ (b) IMDB: $k = 40$ (c) IMDB: $k = 80$



(d) CIFAR-10: $k = 20$ (e) CIFAR-10: $k = 40$ (f) CIFAR-10: $k = 80$

Figure 5: Validation accuracy with error bars based on three runs for fine-tuning BERT-tiny on the IMDB dataset and CNN on CIFAR-10. In these runs, we fix the batch size B to be 512 for IMDb and 1024 for CIFAR-10, we also fix the number of iterations per epoch, $b = |D|/B$, and vary the number of epochs k. We observe improved utility compared to Figure 2, suggesting that using a larger batch size and training for longer is preferable for overall accuracy. We also find that, in this regime, the optimal mechanism is not DP-SGD. In fact, the trend is the opposite: as the number of epochs increases, the optimal λ also increases. This suggests that one can benefit from DP-λCGD even in long training runs, which are common for DP-SGD.



(a) BERT-base on IMDb. (b) ViT-base on CIFAR-100.

Figure 6: Test accuracy as a function of the privacy budget ϵ for **(a)** BERT-base fine-tuned on IMDb for sentiment analysis and **(b)** ViT-base fine-tuned (two blocks) on CIFAR-100 for image classification. Both models are trained with batch size 512 for 10 epochs. Our proposed DP-λCGD outperforms DP-SGD at most privacy levels, achieves strong performance under very strict privacy ($\epsilon = 1/4$), and is competitive with BLT, BandInvMF and BISR factorizations. Error bars are computed from three runs.

Table 1: Time to generate and correlate the noise at the final (nth) iteration and buffer size to produce correlated noise in dimension d. For BLT, BSR, and BandMF, regeneration requires replaying from iteration 1, so the last-iteration cost is nd. All methods also require correlating noise by summing p vectors of size d ($+pd$ in the runtime).

	Time per iter. Gen + Corr		Buffer size	
Method	w/o regen.	w/ regen.	w/o regen.	w/ regen.
DP-SGD	$d + d$	$d + d$	–	–
DP-λCGD	$d + 2d$	$2d + 2d$	d	–
BandInvMF	$d + pd$	$pd + pd$	$(p-1)d$	–
BISR	$d + pd$	$pd + pd$	$(p-1)d$	–
BLT	$d + pd$	$nd + pd$	$(p-1)d$	–
BSR	$d + pd$	$nd + pd$	$(p-1)d$	–
BandMF	$d + pd$	$nd + pd$	$(p-1)d$	–

Table 2: RMSE for different mechanisms across privacy budgets ϵ in the setting of training on CIFAR-10 with batch size 128 over $k = 10$ epochs. The first column reports results without amplification by subsampling; in this setting, BandMF (McKenna, 2025) with full bandwidth achieves the best performance. When Balls-in-Bins amplification via subsampling is applied, other methods perform better, including DP-λCGD, BLT (Dvijotham et al., 2024), and BISR (Kalinin et al., 2026) with bandwidth $p = 16$. In the high-privacy regime, subsampling substantially affects RMSE, and methods that account for amplification, especially DP-λCGD, benefit the most.

Method	$\epsilon = 8$ (w/o Amp)	$\epsilon = 8$	$\epsilon = 4$	$\epsilon = 2$	$\epsilon = 1$	$\epsilon = 0.5$	$\epsilon = 0.25$
BandInvMF ($p = 2$)	12.69	9.70	14.90	24.41	43.66	82.77	125.87
BandInvMF ($p = 4$)	10.27	8.07	12.48	20.76	37.37	70.90	130.48
BandInvMF ($p = 16$)	8.54	6.56	11.23	20.17	37.09	71.08	130.90
BandInvMF ($p = 64$)	8.15	6.39	11.36	20.87	39.13	74.06	140.38
BandInvMF ($p = b$)	7.87	5.66	9.86	17.76	33.25	62.84	119.20
BLT	8.14	5.87	10.67	19.50	35.97	69.41	127.69
DP-λCGD ($\lambda = 0.975$)	12.73	9.33	14.03	21.45	36.40	68.37	125.69
DP-λCGD ($\lambda = 0.95$)	14.74	10.27	15.28	21.45	31.70	56.75	103.44
DP-λCGD ($\lambda = 0.9$)	19.72	13.25	18.42	24.73	33.66	54.33	97.73
BISR ($p = 2$)	48.45	30.52	40.58	49.44	58.16	71.20	105.09
BISR ($p = 4$)	33.47	21.00	27.97	34.87	42.50	58.27	98.52
BISR ($p = 16$)	17.95	11.50	15.75	21.10	20.25	52.81	96.45
BISR ($p = 64$)	10.50	6.81	9.87	15.92	28.80	53.89	101.90
BISR ($p = b$)	8.45	6.34	11.30	20.77	38.96	73.70	139.68
BSR ($p = 2$)	62.51	39.65	51.87	62.97	74.14	88.33	116.43
BSR ($p = 4$)	46.80	29.29	39.32	48.13	57.03	70.47	104.90
BSR ($p = 16$)	26.27	16.88	22.41	28.59	36.78	55.03	97.09
BSR ($p = 64$)	14.89	9.62	13.32	18.88	29.34	52.16	97.51
BSR ($p = b$)	8.15	5.49	9.31	17.06	31.95	60.43	114.52
BandMF ($p = 2$)	59.44	37.67	49.08	59.95	71.17	86.25	114.94
BandMF ($p = 4$)	42.29	27.00	36.99	45.90	55.35	68.64	104.42
BandMF ($p = 16$)	22.05	15.00	21.20	28.13	37.06	56.29	99.33
BandMF ($p = 64$)	12.58	8.77	13.25	19.76	31.40	56.07	104.68
BandMF ($p = b$)	7.77	5.84	10.42	19.16	35.94	68.02	128.79
DP-SGD	83.85	21.82	26.27	31.68	40.10	59.17	100.27

B DP-λCGD Algorithm

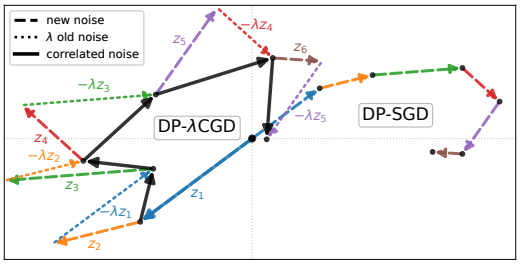


Figure 7: Visualization of noise correlation in DP-λCGD. At each step, fresh noise z_i is generated, and a λ-fraction of the previously added noise z_{i-1} is canceled. In contrast, DP-SGD adds independent noise at each iteration. Although the per-iteration noise vectors in DP-λCGD have larger variance, their correlation leads to partial noise cancellation, yielding a lower total variance.

Algorithm 1 DP-λCGD

Require: Initialization $\theta_0 \in \mathbb{R}^d$, dataset $\mathcal{D}$, batch size B, clip norm ζ, learning rate $\eta > 0$, loss $\ell(\theta, d)$, parameter $\lambda \in [0, 1)$, number of epochs k, target privacy level ϵ, δ, number of iterations n, pseudorandom noise generator gen.

Construct matrix $C_\lambda = \text{Lower_Triangular_Toeplitz}(1, \lambda, \ldots, \lambda^{n-1})$
if use Balls in Bins accountant **then**
  Compute noise multiplier $\sigma = \text{BnB}(C_\lambda, k, \epsilon, \delta)$ and allocate batches $D_0, \ldots, D_{|\mathcal{D}|/B-1}$
else if no amplification **then**
  $\sigma = \text{sens}_{k,|\mathcal{D}|/B}(C_\lambda)\, \sigma_{\epsilon,\delta}$, where $\sigma_{\epsilon,\delta}$ is from the Gaussian mechanism
end if
$Z_0 \leftarrow 0$
for $i = 1$ to n **do**
  if use Balls in Bins subsampling **then**
    $S_i = D_{i-1 \bmod (n/k)}$ ▷ Sample batch
  else if no amplification **then**
    $S_i = \{d_{(i-1)B \bmod |\mathcal{D}|}, \ldots, d_{(iB-1) \bmod |\mathcal{D}|}\}$
  end if
  for $j = 1$ to $|S_i|$ **do**
    $g_j \leftarrow \nabla_\theta \ell(\theta_{i-1}, d_j)$
    $\tilde{g}_j \leftarrow \min\left(1, \frac{\zeta}{\|g_j\|_2}\right) g_j$ ▷ per-example clipping
  end for
  $x_i \leftarrow \sum_{j=1}^{|S_i|} \tilde{g}_j$ ▷ aggregate clipped gradients
  if $i \neq 1$ **then**
    gen $\leftarrow$ gen.set_state(GSt)
    $Z_{i-1} \sim$ gen() ▷ regenerate noise
  end if
  $\hat{x}_i \leftarrow x_i - \zeta\sigma\lambda Z_{i-1}$
  GSt $\leftarrow$ gen.get_state()
  $Z_i \sim$ gen() ▷ generate fresh noise
  $\hat{x}_i \leftarrow \hat{x}_i + \zeta\sigma Z_i$
  $\theta_i \leftarrow \theta_{i-1} - \frac{\eta}{B}\hat{x}_i$ ▷ model update
end for
return θ_n

We present the full algorithm for DP-λCGD in Algorithm 1. In addition to noise correlation, we emphasize the role of the subsampling accountant BnB and the batch sampling scheme. The noise regeneration technique assumes a pseudo-random generator gen that supports setting and retrieving its state via gen.set_state() and gen.get_state(), respectively. We illustrate the noise correlation process in Figure 7.

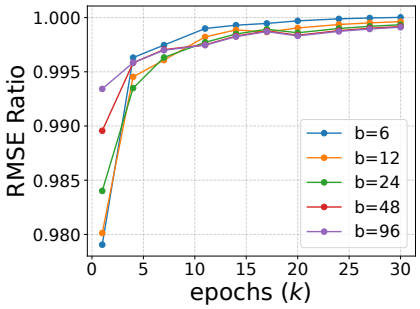


Figure 8: Ratio of RMSE for column-normalized DP-λCGD to DP-λCGD for varying number of epochs k and number of iterations per epoch (separation parameter) b. Small k yields a larger improvement from normalization.

C Column Normalized DP-λCGD

The performance of DP-λCGD can be further improved by column normalization. Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ denote the column norms of the matrix C_λ, and consider the normalized DP-λCGD factorization with strategy matrix $C_\lambda D^{-1}$.

It has been observed that column normalization improves performance for banded matrices (Pillutla et al., 2025), and it provably improves both RMSE and MaxSE for the square-root factorization under single participation (Henzinger et al., 2025). The efficiency of the DP-λCGD is preserved; in fact any linear noise correlation via matrix C^{-1} can be done without loss of efficiency, we formulate it as the following proposition.

Proposition 6 (Cost of column normalization)**.** *Let $C \in \mathbb{R}^{n\times n}$ be invertible and let $Z \in \mathbb{R}^n$ be any noise vector. Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ with $d_j > 0$ (e.g., $d_j = \|C_{:,j}\|_2$), and define the column-normalized matrix $\widetilde{C} = CD^{-1}$. Then for any Z, $\widetilde{C}^{-1}Z = D\,(C^{-1}Z)$. Consequently, if computing $C^{-1}Z$ has time complexity $T(n)$, then computing $\widetilde{C}^{-1}Z$ has time complexity $T(n) + \mathcal{O}(n)$, i.e., only $\mathcal{O}(1)$ additional work per component (one scaling by d_i).*

Proof. Since $\widetilde{C} = CD^{-1}$, we have $\widetilde{C}^{-1} = DC^{-1}$. Multiplying by Z gives $\widetilde{C}^{-1}Z = D(C^{-1}Z)$. The extra cost beyond $C^{-1}Z$ is applying the diagonal matrix D, which is n scalar multiplications. □

All efficiency properties are preserved: to correlate the noise Z, we first multiply it by C_λ^{-1}, which we have shown can be computed efficiently, and then rescale it by the diagonal matrix D.

There is, however, one subtle issue. The matrix $C_\lambda D^{-1}$ is no longer Toeplitz, and therefore the sensitivity Theorem 1 cannot be applied directly. In this work, we generalize the sensitivity theorem specifically for the normalized DP-λCGD in the following theorem.

Lemma 7. *The sensitivity of the column-normalized DP-λCGD with strategy matrix C_λ and column norms $D = \mathrm{diag}(d_1, \ldots, d_n)$ is given by*

$$\mathrm{sens}_{k,b}(C_\lambda D^{-1}) = \left\| \sum_{j=0}^{k-1} (C_\lambda D^{-1})_{:,\, jb+1} \right\|_2. \tag{9}$$

For the amplified version, there are no such restrictions: the Balls-in-Bins accountant applies to any positive lower-triangular matrix, which includes the (normalized) DP-λCGD matrices.

We show numerically that the column normalization leads to the improvement in RMSE metric, see Figure 8.

Lemma 8. *For any $\lambda \in (0, 1)$ under the single-participation model, the RMSE of column-normalized DP-λCGD is smaller than that of the DP-λCGD.*

D Full Experiment Details

Table 3: Runtime comparison (seconds) across models and bandwidths for CIFAR-10 training over one epoch. CNN, WideResNet-40, and ViT-B/16 are trained using 8 CPU cores and an NVIDIA A100 GPU. We use a batch size of 512 with a physical batch size of 64. For ViT-L/16 we use NVIDIA H100 GPU with 64 CPU cores and physical batch size 16. The standard deviations are computed over 3 runs for each model, except for CNN, for which we used 10 runs to get more reliable estimates.

	DP-SGD	GPU Noise Regeneration			CPU Noise Buffer		
Model		2	4	16	2	4	16
CNN (0.3M)	18.5 ± 0.3	18.8 ± 0.2	18.9 ± 0.3	19.4 ± 0.2	19.1 ± 0.9	19.0 ± 0.3	19.5 ± 1.5
WideResNet-40 (9M)	133.7 ± 2.2	133.5 ± 1.2	136.9 ± 1.3	144.1 ± 1.0	143.5 ± 2.3	143.1 ± 1.5	149.6 ± 1.9
ViT-B/16 (86M)	825.3 ± 1.8	826.9 ± 3.2	829.2 ± 1.7	842.9 ± 2.8	886.9 ± 2.0	897.9 ± 3.0	987.2 ± 0.4
ViT-L/16 (307M)	1357.9 ± 1.9	1363.2 ± 4.2	1364.7 ± 4.6	1367.3 ± 2.2	1507.6 ± 3.1	1543.3 ± 6.3	1842.7 ± 12.9
ViT-H/14 (632M)	3252.5 ± 3.8	3275.7 ± 6.2	3288.6 ± 4.6	3290.8 ± 8.5	3544.2 ± 3.1	3598.9 ± 6.7	4068.9 ± 7.2

Table 4: Runtime comparison (seconds) across models and noise-buffer sizes for IMDB fine-tuning over one epoch. We use batch size 128. LSTM, BERT-Tiny, and BERT-Base are run on an NVIDIA A100 with 64 CPU cores and physical batch size 64; BERT-Large is run on an NVIDIA H100 with 64 CPU cores with physical batch size 16. The standard deviations are computed over 3 runs for each model, except for BERT-Tiny, for which we used 10 runs to get more reliable estimates.

	DP-SGD	GPU Noise Regeneration			CPU Noise Buffer		
Model		2	4	16	2	4	16
LSTM (1M)	310.2 ± 3.8	317.1 ± 1.3	317.7 ± 5.8	318.8 ± 3.6	322.4 ± 3.7	322.8 ± 3.1	325.9 ± 3.0
BERT-Tiny (3M)	14.2 ± 0.6	14.8 ± 0.4	15.5 ± 0.7	20.6 ± 0.8	25.2 ± 1.4	25.5 ± 2.1	29.8 ± 2.6
BERT-Base (110M)	334.3 ± 0.8	339.1 ± 0.5	342.1 ± 1.6	369.7 ± 1.0	543.3 ± 3.2	554.4 ± 4.8	671.7 ± 4.5
BERT-Large (340M)	499.0 ± 0.1	508.7 ± 2.0	513.0 ± 1.5	539.2 ± 1.7	846.9 ± 3.2	859.4 ± 3.9	1236.5 ± 17.5

Table 5: Hyperparameters for the CIFAR-10 experiments with amplification by subsampling in Figure 4a, for each privacy budget ϵ with $\delta = 10^{-5}$. Batch size B and clipping norm ζ are fixed. We tune the learning rate (lr) for all methods and λ only for DP-λCGD, using a public validation set. We search over lr $\in \{0.01, 0.02, 0.05, 0.08, 0.1, 0.2, 0.5, 1\}$ and $\lambda \in \{0, 0.1, 0.25, 0.5, 0.8, 0.9, 0.95, 0.9625, 0.975, 0.98\}$.

			$\epsilon = 0.25$		$\epsilon = 0.5$		$\epsilon = 1$		$\epsilon = 2$		$\epsilon = 4$		$\epsilon = 8$	
Method	B	ζ	λ	lr	λ	lr	λ	lr	λ	lr	λ	lr	λ	lr
DP-SGD	128	8	0	0.05	0	0.08	0	0.2	0	0.2	0	0.2	0	0.2
DP-λCGD	128	8	0.8	0.05	0.9	0.08	0.9	0.2	0.9	0.2	0.95	0.5	0.95	0.5
BandInvMF ($p = 2$)	128	8	0.977	0.02	0.977	0.05	0.977	0.08	0.977	0.2	0.977	0.2	0.977	0.5
BandInvMF ($p = 16$)	128	8	—	0.02	—	0.08	—	0.08	—	0.2	—	0.5	—	0.5
BISR ($p = 2$)	128	8	0.5	0.05	0.5	0.08	0.5	0.1	0.5	0.1	0.5	0.2	0.5	0.2
BISR ($p = 16$)	128	8	—	0.05	—	0.1	—	0.2	—	0.2	—	0.5	—	0.5
BLT	128	8	—	0.05	—	0.1	—	0.2	—	0.5	—	0.5	—	0.5

Table 6: Hyperparameters for the CIFAR-10 experiments without amplification by subsampling in Figure 4b, for each privacy budget ϵ with $\delta = 10^{-5}$. Batch size B and clipping norm ζ are fixed. We tune the learning rate lr for all methods and λ only for DP-λCGD, using a public validation set.

			$\epsilon = 0.25$		$\epsilon = 0.5$		$\epsilon = 1$		$\epsilon = 2$		$\epsilon = 4$		$\epsilon = 8$	
Method	B	ζ	λ	lr	λ	lr	λ	lr	λ	lr	λ	lr	λ	lr
DP-SGD	128	8	0	0.002	0	0.005	0	0.01	0	0.02	0	0.05	0	0.05
DP-λCGD	128	8	0.9625	0.005	0.9625	0.02	0.95	0.05	0.95	0.1	0.95	0.2	0.95	0.5
BandInvMF ($p = 2$)	128	8	0.977	0.01	0.977	0.02	0.977	0.05	0.977	0.08	0.977	0.1	0.977	0.5
BandInvMF ($p = 16$)	128	8	—	0.01	—	0.01	—	0.1	—	0.2	—	0.2	—	0.5
BISR ($p = 2$)	128	8	0.5	0.005	0.5	0.005	0.5	0.02	0.5	0.02	0.5	0.05	0.5	0.1
BISR ($p = 16$)	128	8	—	0.01	—	0.02	—	0.05	—	0.1	—	0.2	—	0.5
BLT	128	8	—	0.01	—	0.02	—	0.05	—	0.1	—	0.2	—	0.5

E PROOFS

Lemma 2. *For $n = kb$, the MaxSE and RMSE of DP-λCGD, optimized over $\lambda \in [0, 1)$, satisfy*

$$\begin{aligned} \inf_{\lambda\in[0,1)} \mathrm{MaxSE}(AC_\lambda^{-1}, C_\lambda) &= \Theta\big(k + \sqrt{k}\, n^{1/4}\big), \\ \inf_{\lambda\in[0,1)} \mathrm{RMSE}(AC_\lambda^{-1}, C_\lambda) &= \Theta\big(k + \sqrt{k}\, n^{1/4}\big). \end{aligned} \tag{7}$$

Proof. For any factorization, MaxSE is at least as large as RMSE. Therefore, it suffices to prove the upper bound $O(k+\sqrt{k}n^{1/4})$ for MaxSE and the lower bound $\Omega(k+\sqrt{k}n^{1/4})$ for RMSE. We begin with the upper bound.

First, we compute the squared sensitivity of the matrix C_λ:

$$\mathrm{sens}_{k,b}^2(C_\lambda) = \left\| \sum_{j=0}^{k-1} (C_\lambda)_{:,jb+1} \right\|_2^2 = \sum_{j=0}^{k-1}\sum_{r=0}^{b-1} \left(\sum_{m=0}^{j} \lambda^{jb+r-mb} \right)^2 \tag{10}$$

$$= \sum_{j=0}^{k-1}\sum_{r=0}^{b-1} \lambda^{2r} \left(\frac{1-\lambda^{b(j+1)}}{1-\lambda^b} \right)^2 = \frac{1-\lambda^{2b}}{(1-\lambda^2)(1-\lambda^b)^2} \sum_{j=1}^{k} (1-\lambda^{bj})^2. \tag{11}$$

The squared maximum row ℓ_2-norm of the matrix AC_λ^{-1} is

$$\|AC_\lambda^{-1}\|_{2\to\infty}^2 = 1 + (1-\lambda)^2(n-1). \tag{12}$$

Combining these expressions gives the squared maximum error. To prove the claimed upper bound, we consider two regimes.

First, suppose $k \geq \sqrt{n}$ and set $\lambda = e^{-1/b}$. This yields

$$\|AC_\lambda^{-1}\|_{2\to\infty}^2 \cdot \mathrm{sens}_{k,b}^2(C_\lambda) = \left(1 + \frac{n-1}{b^2} + O(nb^{-3})\right) \left(\frac{(1-e^{-2})(k+O(1))}{(2b^{-1}+O(b^{-2}))(1-e^{-1})^2} \right) \tag{13}$$

$$= O\left(\frac{nkb}{b^2}\right) = O(k^2), \tag{14}$$

assuming $n = kb$.

If $k < \sqrt{n}$, take $\lambda = e^{-1/\sqrt{n}}$. Then $\lambda^b = e^{-\sqrt{n}/k} < 1/e = O(1)$ and $1 - \lambda^b = \Theta(1)$,

$$\|AC_\lambda^{-1}\|_{2\to\infty}^2 \cdot \mathrm{sens}_{k,b}^2(C_\lambda) = O(1)\left(\frac{O(1)}{2/\sqrt{n} + O(1/n)} O(k) \right) = O(k\sqrt{n}). \tag{15}$$

Combining the two bounds and taking square roots yields the claimed upper bound.

For the lower bound on RMSE, there is a universal $\Omega(k)$ lower bound that holds for any factorization (Kalinin et al., 2026). It remains only to prove the $\Omega(\sqrt{k}n^{1/4})$ lower bound. We first lower-bound the sensitivity:

$$\begin{aligned} \mathrm{sens}_{k,b}^2(C_\lambda) = \left\| \sum_{j=0}^{k-1} (C_\lambda)_{:,jb+1} \right\|_2^2 &\geq \sum_{j=0}^{k-1} \|(C_\lambda)_{:,jb+1}\|_2^2 \geq \sum_{j=0}^{\lceil k/2\rceil - 1} \|(C_\lambda)_{:,jb+1}\|_2^2 \\ &\geq \lceil k/2\rceil \cdot \|(C_\lambda)_{:,\lceil k/2\rceil b-b+1}\|_2^2 = \lceil k/2\rceil \sum_{j=0}^{n-\lceil k/2\rceil b+b-1} \lambda^{2j} \\ &= \lceil k/2\rceil \frac{1-\lambda^{2n-2\lceil k/2\rceil b+2b}}{1-\lambda^2} \geq \lceil k/2\rceil \frac{1-\lambda^n}{1-\lambda^2} \geq \frac{k}{4}\frac{1-\lambda^n}{1-\lambda}. \end{aligned} \tag{16}$$

Next, we prove the following auxiliary inequality for any $\lambda \in [0, 1)$ and n:

$$1 - \lambda^n \geq \frac{n(1-\lambda)}{1+n(1-\lambda)}. \tag{17}$$

This follows from Bernoulli's inequality:

$$\lambda^n = (1-(1-\lambda))^n \leq \frac{1}{(1+(1-\lambda))^n} \leq \frac{1}{1+n(1-\lambda)}. \tag{18}$$

Multiplying the lower bound on the sensitivity by $\frac{1}{n}\|AC_\lambda^{-1}\|_F^2$ gives

$$\begin{aligned}\inf_{\lambda\in[0,1)} \tfrac{1}{n}\|AC_\lambda^{-1}\|_F^2 \cdot \mathrm{sens}_{k,b}^2(C_\lambda) &\geq \inf_{\lambda\in[0,1)} \frac{kn(1+(1-\lambda)^2(n-1)/2)}{4(1+n(1-\lambda))} \\ &\geq \frac{kn}{4} \inf_{\lambda\in[0,1)} \frac{1+(1-\lambda)^2(n-1)/2}{1+n(1-\lambda)}.\end{aligned} \tag{19}$$

To lower-bound the infimum, we consider two cases. First, suppose that $1-\lambda \leq \frac{1}{\sqrt{n}}$. Then

$$\frac{1+(1-\lambda)^2(n-1)/2}{1+n(1-\lambda)} \geq \frac{1}{1+n(1-\lambda)} \geq \frac{1}{1+\sqrt{n}} \geq \frac{1}{8\sqrt{n}}. \tag{20}$$

Second, suppose that $1-\lambda \geq \frac{1}{\sqrt{n}}$. Then, for $n \geq 2$,

$$\frac{1+(1-\lambda)^2(n-1)/2}{1+n(1-\lambda)} \geq \frac{(1-\lambda)^2(n-1)}{4n(1-\lambda)} = \frac{(1-\lambda)(n-1)}{4n} \geq \frac{1-\lambda}{8} \geq \frac{1}{8\sqrt{n}}. \tag{21}$$

Therefore,

$$\inf_{\lambda\in[0,1)} \tfrac{1}{n}\|AC_\lambda^{-1}\|_F^2 \cdot \mathrm{sens}_{k,b}^2(C_\lambda) \geq \frac{kn}{32\sqrt{n}} = \frac{k\sqrt{n}}{32} = \Omega(k\sqrt{n}). \tag{22}$$

Taking square roots yields the $\Omega(\sqrt{k}n^{1/4})$ lower bound. Combining this with the $\Omega(k)$ lower bound concludes the proof. □

Lemma 3. *For DP-λCGD, a minimizer of MaxSE is always at least as large as a minimizer of RMSE, both with and without amplification by subsampling.*

Proof. Define, for $\gamma > 0$,

$$F_\gamma(\lambda) = \left(1+(1-\lambda)^2(n-1)\gamma\right)\sigma_{\epsilon,\delta}^2(C_\lambda), \tag{23}$$

where $\sigma_{\epsilon,\delta}(C_\lambda)$ is the noise multiplier in the amplified setting and sensitivity in the non-amplified setting. Then squared MaxSE is given by $F_1(\lambda)$ and squared RMSE is given by $F_{1/2}(\lambda)$. We claim that $\frac{F_1(\lambda)}{F_{1/2}(\lambda)}$ is a monotonically decreasing function of λ. Indeed, for $\lambda \in (0,1)$:

$$\frac{d}{d\lambda}\left(\frac{F_1(\lambda)}{F_{1/2}(\lambda)}\right) = \frac{d}{d\lambda}\left(\frac{1+(1-\lambda)^2(n-1)}{1+(1-\lambda)^2(n-1)/2}\right) = -\frac{(n-1)(1-\lambda)}{(1+(1-\lambda)^2(n-1)/2)^2} < 0. \tag{24}$$

Let λ_1 be a minimizer of $F_1(\lambda)$ and let $\lambda_{1/2}$ be a minimizer of $F_{1/2}(\lambda)$. Assume for contradiction that $\lambda_1 < \lambda_{1/2}$. Since λ_1 and $\lambda_{1/2}$ are minimizers and $\sigma_{\epsilon,\delta}(C_\lambda) > 0$, we have

$$F_1(\lambda_1) \leq F_1(\lambda_{1/2}) \quad \text{and} \quad F_{1/2}(\lambda_{1/2}) \leq F_{1/2}(\lambda_1). \tag{25}$$

Combining these two inequalities gives

$$\frac{F_1(\lambda_1)}{F_{1/2}(\lambda_1)} \leq \frac{F_1(\lambda_{1/2})}{F_{1/2}(\lambda_{1/2})}. \tag{26}$$

However, the ratio $\frac{F_1(\lambda)}{F_{1/2}(\lambda)}$ is decreasing in λ, and $\lambda_1 < \lambda_{1/2}$ would imply the reverse inequality, which is a contradiction. Therefore $\lambda_1 \geq \lambda_{1/2}$. □

Theorem 4. *Let $b = 1$ and $k = n$, and let $A \in \mathbb{R}^{n\times n}$ be the prefix-sum matrix. Define the diagonal factorization $C_{\mathrm{diag}} = \mathrm{diag}\big((n-j+1)^{1/4}\big)_{j=1}^{n}$ and $B_{\mathrm{diag}} = AC_{\mathrm{diag}}^{-1}$. Then, for any factorization $A = BC$, the following lower bound holds:*

$$\mathrm{RMSE}(B,C) \geq \mathrm{RMSE}(B_{\mathrm{diag}}, C_{\mathrm{diag}}) = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \sqrt{j} \sim \frac{2n}{3}. \tag{8}$$

Moreover, if the matrix C is lower triangular with positive diagonal, then equality holds if and only if $C = \alpha C_{\mathrm{diag}}$ for some $\alpha > 0$.

Proof. By the definition of RMSE (Equation 5), for any factorization $A = BC$, we have

$$\mathrm{RMSE}(B,C) = \frac{1}{\sqrt{n}} \|B\|_F \,\mathrm{sens}_{n,1}(C), \quad \text{for} \quad \mathrm{sens}_{n,1}(C) = \sup_{X \sim X'} \|CX - CX'\|_F. \tag{27}$$

Here, neighboring datasets may differ in any coordinates, and for each i, $\|X_i - X_i'\|_2 \leq 1$. To obtain a lower bound, take X and X' that differ in all coordinates and lie along a single dimension. Thus,

$$\mathrm{sens}_{n,1}(C) \geq \|C\|_{\infty\to 2} = \sup_{\|x\|_\infty \leq 1} \|Cx\|_2. \tag{28}$$

It would then suffice to prove that

$$\inf_{BC=A} \frac{1}{\sqrt{n}} \|B\|_F \|C\|_{\infty\to 2} = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \sqrt{j}. \tag{29}$$

Let $u_j = \frac{Ae_j}{\|Ae_j\|_2}$ and consider the matrix $U = [u_1, \ldots, u_n]$. Then

$$\sum_{j=1}^{n} \sqrt{j} = \sum_{j=1}^{n} \|Ae_j\|_2 = \mathrm{tr}(U^\top A) = \mathrm{tr}(U^\top BC) = \mathrm{tr}(BCU^\top) \leq \|B\|_F \|UC^\top\|_F. \tag{30}$$

It is thus sufficient to show that $\|UC^\top\|_F \leq \|C\|_{\infty\to 2}$. For this, we use a probabilistic argument.

Let z be a random vector such that $\mathbb{E}[zz^\top] = U^\top U$ and $\|z\|_\infty \leq 1$. Then

$$\begin{aligned} \|C\|_{\infty\to 2}^2 = \sup_{\|x\|_\infty \leq 1} \|Cx\|_2^2 \geq \mathbb{E}\|Cz\|_2^2 = \mathbb{E}[\mathrm{tr}(Czz^\top C^\top)] = \mathrm{tr}\big(C^\top C\, \mathbb{E}[zz^\top]\big) \\ = \mathrm{tr}(C^\top C U^\top U) = \|UC^\top\|_F^2. \end{aligned} \tag{31}$$

It remains only to show that such a random vector z exists. The entries of the matrix $U^\top U$ are given by

$$(U^\top U)_{i,j} = \langle u_i, u_j \rangle = \frac{n - \max(i,j) + 1}{\|Ae_i\|_2 \|Ae_j\|_2} = \sqrt{\frac{n - \max(i,j) + 1}{n - \min(i,j) + 1}} \leq 1. \tag{32}$$

Consider auxiliary independent random variables $y_1, \ldots, y_{n-1}$ taking values in $\{-1, 1\}$, with the probability of taking the value 1 given by

$$\mathbb{P}(y_j = 1) = \frac{1 + \sqrt{\frac{n-j}{n-j+1}}}{2}, \qquad \text{and therefore} \qquad \mathbb{E}[y_j] = \sqrt{\frac{n-j}{n-j+1}}. \tag{33}$$

Let z_1 be uniformly distributed on $\{-1, 1\}$, independently of $y_1, \ldots, y_{n-1}$, and let $z_j = z_1 \prod_{i=1}^{j-1} y_i$. Then $\|z\|_\infty = 1$ by construction, while

$$\begin{aligned} \mathbb{E}[z_i z_j] = \mathbb{E}\left[z_1^2 \prod_{l=1}^{i-1} y_l \prod_{r=1}^{j-1} y_r \right] = \mathbb{E}\left[\prod_{l=\min(i,j)}^{\max(i,j)-1} y_l \right] = \prod_{l=\min(i,j)}^{\max(i,j)-1} \mathbb{E}[y_l] \\ = \sqrt{\frac{n - \max(i,j) + 1}{n - \min(i,j) + 1}} = \langle u_i, u_j \rangle. \end{aligned} \tag{34}$$

Consequently,

$$\frac{1}{\sqrt{n}}\|B\|_F\|C\|_{\infty\to 2} \geq \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\|Ae_j\|_2 = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\sqrt{n-j+1} = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\sqrt{j}. \tag{35}$$

If equality holds, then

$$\|C\|_{\infty\to 2}^2 = \sup_{\|x\|_\infty\leq 1}\|Cx\|_2^2 = \mathbb{E}\|Cz\|_2^2. \tag{36}$$

The vector z has non-zero probability of taking every possible assignment in $\{-1,1\}^n$. Since its expected value attains the largest possible value of $\|Cz\|_2^2$, every assignment of z must give the same value. This is possible only if$(C^\top C)_{i,j} = 0$, for $i \neq j$, hence $C^\top C$ is diagonal.

Since C is lower triangular and invertible, this implies that C itself is diagonal. Indeed, the last column of C contains only the non-zero entry $C_{n,n}$, and its orthogonality to every preceding column implies $C_{n,j} = 0$ for all $j < n$. Repeating the argument recursively for the leading principal submatrix shows that all off-diagonal entries of C vanish.

For a diagonal matrix C, it follows from Proposition E.1 in Choquette-Choo et al. (2023a) that $\mathrm{sens}_{n,1}(C) = \|Ce\|_2$. To show the matching upper bound, we use the following explicit diagonal construction:

$$C_{\mathrm{diag}} = \mathrm{diag}\big(n^{1/4},(n-1)^{1/4},\ldots,1^{1/4}\big), \qquad (B_{\mathrm{diag}})_{i,j} = \begin{cases}(n-j+1)^{-1/4}, & i\geq j,\\ 0, & \text{o/w.}\end{cases} \tag{37}$$

Then

$$\|C_{\mathrm{diag}}e\|_2 = \sqrt{\sum_{j=1}^{n}\sqrt{j}}, \quad \|B_{\mathrm{diag}}\|_F = \sqrt{\sum_{j=1}^{n}(n-j+1)\cdot(n-j+1)^{-1/2}} = \sqrt{\sum_{j=1}^{n}\sqrt{j}}. \tag{38}$$

Thus,

$$\mathrm{RMSE}(B_{\mathrm{diag}},C_{\mathrm{diag}}) = \frac{1}{\sqrt{n}}\|B_{\mathrm{diag}}\|_F\|C_{\mathrm{diag}}e\|_2 = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\sqrt{j}, \tag{39}$$

showing that the stated lower bound is attained. Finally, by the Cauchy–Schwarz inequality, this diagonal factorization is optimal among all diagonal choices of C. Thus, if C is diagonal and equality is attained, then C must be a positive scalar multiple of C_{diag}.

The error of the trivial factorization. The trivial factorization $C = I$ yields

$$\mathrm{RMSE}(A,I) \;=\; \sqrt{\frac{n(n+1)}{2}} \;\sim\; \frac{n}{\sqrt{2}}. \tag{40}$$

This is approximately 6% larger than the error of the optimal factorization with $C = C_{\mathrm{diag}}$, whose error is

$$\mathrm{RMSE}(B_{\mathrm{diag}},C_{\mathrm{diag}}) = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\sqrt{j} \sim \frac{2n}{3}. \tag{41}$$

□

Corollary 5. *The trivial factorization $B = A$, $C = I$ is optimal for DP-λCGD under the RMSE and MaxSE metrics for $b = 1$ and $k = n$.*

Proof. For the MaxSE metric, Pillutla et al. (2025, Proposition 3.11) show that, in dimension $d = 1$, the trivial factorization $B = A$, $C = I$ is optimal among all factorizations when $b = 1$ and $k = n$. Since the sensitivity $\mathrm{sens}_{n,1}(C)$ for $d > 1$ is at least as large as in dimension $d = 1$, the same lower bound applies in any dimension. Hence, DP-SGD is MaxSE-optimal for all d in the full-batch regime. In particular, the trivial factorization is also optimal within the DP-λCGD family under the MaxSE metric.

Finally, Lemma 3 shows that the RMSE-optimal value of λ is no larger than the MaxSE-optimal one. Since $\lambda \geq 0$ and the latter is 0 in the full-batch regime, the RMSE-optimal value must also be $\lambda = 0$. Therefore, the trivial factorization is optimal for DP-λCGD under both the RMSE and MaxSE metrics.

□

Lemma 7. *The sensitivity of the column-normalized DP-λCGD with strategy matrix C_λ and column norms $D = \mathrm{diag}(d_1, \ldots, d_n)$ is given by*

$$\mathrm{sens}_{k,b}(C_\lambda D^{-1}) = \Big\| \sum_{j=0}^{k-1} (C_\lambda D^{-1})_{:,\, jb+1} \Big\|_2. \tag{9}$$

Proof. We first note that the matrix $C_\lambda D^{-1}$ is no longer Toeplitz, and therefore Theorem 1 does not apply directly. We instead repeat the argument of the proof of that theorem for this specific matrix.

For an element-wise nonnegative matrix C, the sensitivity can be written (Choquette-Choo et al., 2023a) as

$$\mathrm{sens}(C) = \sup_{\pi} \sqrt{\sum_{i,j \in \pi} \langle C_{:,i}, C_{:,j} \rangle}, \tag{42}$$

where π ranges over sets of at most k column indices with minimum separation b.

Following the argument of Kalinin & Lampert (2024), we show that the maximizing set π consists of the leftmost admissible columns. Specifically, if the optimal indices are not the leftmost ones (subject to b-separation), then the entire block can be shifted left without decreasing the objective. It suffices to show two facts:

1. If $i + b < j$, then
$$\langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j} \rangle \le \langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j-1} \rangle.$$
2. If $i > 1$ and $i + b \le j$, then
$$\langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j} \rangle \le \langle (C_\lambda D^{-1})_{:,i-1}, (C_\lambda D^{-1})_{:,j-1} \rangle.$$

Since all columns are normalized to unit norm, these properties imply that the maximum sensitivity is attained by choosing the leftmost k columns with separation b, i.e., indices $\{1,\, 1+b,\, \ldots,\, 1+(k-1)b\}$.

We now verify the two claims. The inner product between the ith and jth columns of $C_\lambda D^{-1}$ is

$$\langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j} \rangle = \frac{1}{d_i d_j} \sum_{t=0}^{n-j} \lambda^t \lambda^{j-i+t}, \tag{43}$$

where

$$d_i = \sqrt{\sum_{t=0}^{n-i} \lambda^{2t}} = \sqrt{\frac{1-\lambda^{2(n-i+1)}}{1-\lambda^2}}. \tag{44}$$

Similarly, the inner product between columns i and $j-1$ is

$$\langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j-1} \rangle = \frac{1}{d_i d_{j-1}} \sum_{t=0}^{n-j+1} \lambda^t \lambda^{j-i-1+t}. \tag{45}$$

Taking the difference yields

$$\langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j-1} - (C_\lambda D^{-1})_{:,j} \rangle \tag{46}$$

$$= \frac{\lambda^{j-i-1}}{d_i d_{j-1}} d_{j-1}^2 - \frac{\lambda^{j-i}}{d_i d_j} d_j^2 = \frac{\lambda^{j-i-1}}{d_i} (d_{j-1} - \lambda d_j) \ge 0, \tag{47}$$

since the sequence (d_j) is decreasing. This proves the first claim.

For the second claim, we consider

$$\langle (C_\lambda D^{-1})_{:,i-1}, (C_\lambda D^{-1})_{:,j-1} \rangle - \langle (C_\lambda D^{-1})_{:,i}, (C_\lambda D^{-1})_{:,j} \rangle \tag{48}$$

$$= \lambda^{j-i} \left(\frac{d_{j-1}}{d_{i-1}} - \frac{d_j}{d_i} \right). \tag{49}$$

Thus it suffices to show that

$$\frac{d_{j-1}d_i}{d_{i-1}d_j} \geq 1.$$

Using the explicit form of d_i, we obtain

$$\frac{d_{j-1}d_i}{d_{i-1}d_j} = \sqrt{\frac{(1-\lambda^{2(n-j+2)})(1-\lambda^{2(n-i+1)})}{(1-\lambda^{2(n-i+2)})(1-\lambda^{2(n-j+1)})}}. \tag{50}$$

Expanding the numerator and denominator, we see that this ratio is at least one if and only if

$$\lambda^{2(n-j+1)} + \lambda^{2(n-i+2)} - \lambda^{2(n-j+2)} - \lambda^{2(n-i+1)} \geq 0. \tag{51}$$

Rearranging terms gives

$$(1-\lambda^2)\big(\lambda^{2(n-j+1)} - \lambda^{2(n-i+1)}\big) \geq 0, \tag{52}$$

which holds whenever $i \leq j$. This concludes the proof. □

Lemma 8. *For any $\lambda \in (0,1)$ under the single-participation model, the RMSE of column-normalized DP-λCGD is smaller than that of the DP-λCGD.*

Proof. The strategy matrix of DP-λCGD is

$$C_\lambda = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda^{n-1} & \lambda^{n-2} & \cdots & 1 \end{pmatrix}. \tag{53}$$

The squared norms of its columns are

$$d_j^2 = \sum_{i=0}^{n-j} \lambda^{2i} = \frac{1-\lambda^{2(n-j+1)}}{1-\lambda^2}. \tag{54}$$

The sensitivity of C_λ under single participation is d_1. For the factorization $A = B_\lambda C_\lambda$, where A is the prefix-sum matrix, the corresponding left matrix is

$$B_\lambda = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1-\lambda & 1 & 0 & \dots & 0 \\ 1-\lambda & 1-\lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1-\lambda & 1-\lambda & 1-\lambda & \cdots & 1 \end{pmatrix}. \tag{55}$$

Its Frobenius norm can be written as

$$\|B_\lambda\|_F^2 = (1-\lambda)^2 \frac{(n-1)n}{2} + n. \tag{56}$$

The sensitivity of the normalized DP-λCGD is equal to 1 by construction. The normalized strategy matrix is $\tilde{C}_\lambda = C_\lambda D^{-1}$, and the corresponding left matrix is $\tilde{B}_\lambda = ADC_\lambda^{-1}$. Its entries are

$$(\tilde{B}_\lambda)_{i,j} = \begin{cases} d_j - \lambda d_{j+1} & \text{if} \quad j < i, \\ d_j & \text{if} \quad j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

Therefore, its Frobenius norm satisfies

$$\|\tilde{B}_\lambda\|_F^2 = \sum_{j=1}^{n} d_j^2 + \sum_{j=1}^{n-1} (d_j - \lambda d_{j+1})^2 (n-j). \tag{58}$$

To show that the normalized DP-λCGD has lower RMSE than DP-λCGD, it is sufficient to prove

$$\sum_{j=1}^{n} d_j^2 + \sum_{j=1}^{n-1} (d_j - \lambda d_{j+1})^2 (n-j) \leq d_1^2 \left(n + (1-\lambda)^2 \frac{(n-1)n}{2} \right). \tag{59}$$

First, we compute the sum of d_j^2:

$$\sum_{j=1}^{n} d_j^2 = \sum_{j=1}^{n} \sum_{i=0}^{n-j} \lambda^{2i} = \sum_{i=0}^{n-1} \lambda^{2i}(n-i) = n d_1^2 - \sum_{j=1}^{n-1} j\lambda^{2j}. \tag{60}$$

Next, we bound $d_j - \lambda d_{j+1}$ as follows:

$$\begin{aligned} (d_j - \lambda d_{j+1})^2 &= d_j^2 + \lambda^2 d_{j+1}^2 - 2\lambda d_j d_{j+1} \\ &= d_{j+1}^2 + \lambda^{2(n-j)} + \lambda^2 d_{j+1}^2 - 2\lambda d_j d_{j+1} \\ &\leq (1-\lambda)^2 d_{j+1}^2 + \lambda^{2(n-j)}. \end{aligned} \tag{61}$$

Thus,

$$\sum_{j=1}^{n-1} (d_j - \lambda d_{j+1})^2 (n-j) \leq (1-\lambda)^2 \sum_{j=1}^{n-1} d_{j+1}^2 (n-j) + \sum_{j=1}^{n-1} j\lambda^{2j} \tag{62}$$

$$\leq d_1^2 (1-\lambda)^2 \sum_{j=1}^{n-1} (n-j) + \sum_{j=1}^{n-1} j\lambda^{2j} \tag{63}$$

$$= d_1^2 (1-\lambda)^2 \frac{n(n-1)}{2} + \sum_{j=1}^{n-1} j\lambda^{2j}. \tag{64}$$

Combining this bound with equation 60 yields equation 59. □