Correlating Cross-Iteration Noise for DP-SGD using Model Curvature

Xin Gu **Yingtai Xiao** **Guanlin He** **Jiamu Bai** **Daniel Kifer** **Kiwan Maeng**
Penn State University
{xingu, gbh5146, jvb6867, duk17, kvm6242}@psu.edu
yingtai0323@gmail.com

Abstract

Differentially private stochastic gradient descent (DP-SGD) is used to train deep learning models while mitigating many privacy risks. One popular line of work, designed to chip away at the accuracy gap between DP-SGD and normal SGD training, is known as DP-MF. To each update, it adds privacy noise that is correlated across training iterations, so that later noise partially cancels out earlier noise. The noise correlation structure is determined by solving an optimization problem, but a key question is not well-understood: what important properties does the noise correlation need in order to improve accuracy? In this paper, we identify two ways that noise affects the training trajectory — noise alters the *value* of the gradient, and also alters the *location* at which subsequent gradients are computed. Prior methods addressed the first but overlooked the second effect, realizing only limited benefits. Our analysis results in a new objective function for determining cross-iteration noise correlation. Our technique, NoiseCurve, consistently improves accuracy over DP-BANDMF, the state-of-the-art DP-MF scheme, on various computer vision and NLP tasks. NoiseCurve seeks to find an upper bound $\hat{\boldsymbol{H}}$ on the Hessians of the loss that could be encountered during training. We show how to estimate $\hat{\boldsymbol{H}}$ using public data and evaluate robustness to errors in $\hat{\boldsymbol{H}}$. To avoid direct computation of $\hat{\boldsymbol{H}}$, or even its eigenspectrum, we show how to adapt the Lanczos method.

1 Introduction

Differentially private stochastic gradient descent (DP-SGD; Abadi et al. (2016)) is a popular method for training machine learning (ML) models with formal privacy guarantees for the training records. DP-SGD differs from standard SGD in two important ways. First, the gradient of each sample within a batch is clipped by a constant ζ before being averaged. Second, *independent Gaussian noise* is added to the averaged (clipped) gradient before model weights are updated. Although promising, the barrier to widespread adoption is the accuracy gap between DP-SGD and normal SGD.

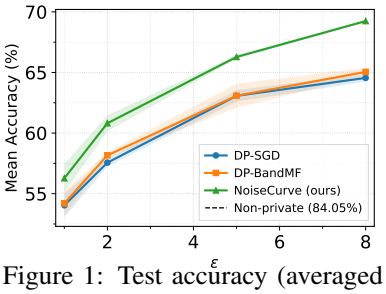


Figure 1: Test accuracy (averaged over 6 runs) on CIFAR-10 and VGG, with $\delta = 10^{-5}$. $\hat{\boldsymbol{H}}$ estimated with TinyImageNet. Non-private accuracy is 84.05%. Details in Section 6.

To improve the accuracy of DP-SGD, a line of research known as *DP-MF* replaces the independent noise with noise that is **correlated across iterations**. The correlations are carefully designed so that noise from some iterations can partially cancel out noise from other iterations, while providing the same privacy guarantee (Kairouz et al., 2021; Choquette-Choo et al., 2023a). The noise correlation is controlled by a mixing matrix $\boldsymbol{C}$, which is obtained by minimizing an objective function subject to

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

privacy constraints. The objective function serves as a data-independent approximation of how the noise causes parameter updates in DP-SGD to differ from ordinary SGD. In this paper, we propose a new objective for $\boldsymbol{C}$ that better models how noise changes the training trajectory. We derive this objective function using a non-convex convergence analysis that studies two effects of noise: (1) it makes the gradient noisy (as is widely recognized); (2) as a result, it alters the point at which the gradient of the *subsequent* iteration is computed — the change in this subsequent gradient is affected by the Hessian of the loss function used for model training. Existing DP-MF objective functions only consider the first error and *disregard the second*.

Due to that second effect, our proposed approach, called NoiseCurve, estimates a function of the eigenspectrum of an upper bound $\grave{\boldsymbol{H}}$ on the Hessians that can be encountered during training. We explain how it can be estimated using unlabeled public datasets and empirically study robustness to estimation errors. We also show how to use the Lanczos method to avoid direct computation of the eigenspectrum. Empirically, NoiseCurve consistently and significantly improves accuracy over the state-of-the-art (see Figure 1).

In summary, our contributions are:

1. **We propose NoiseCurve, a new objective function for generating the cross-iteration noise correlation in the *DP-MF* framework.** It accounts for the cross-iteration interaction between model curvature and gradient noise. It is a drop-in replacement for existing DP-MF software packages, so minibatch creation, privacy accounting, etc., do not need to change.
2. **We show how to use the Lanczos method to efficiently compute the curvature information needed by NoiseCurve**, thus sidestepping the computational difficulties associated with computing Hessians and their eigenspectra. We justify the use of public unlabeled data for obtaining the necessary curvature information and empirically study the effects of estimation errors.
3. **NoiseCurve consistently outperforms the state-of-the-art (SOTA)** DP-MF method, DP-BANDMF, across all the privacy budgets, datasets, and models we tested, on computer vision and NLP tasks (for example, see Figure 1).

2 Related Work

Cross-iteration correlated noise in DP training was first introduced by Kairouz et al. (2021) using tree aggregation; later work (DP-MF family) used matrix factorization instead (Denisov et al., 2022). Choquette-Choo et al. (2023b) extended it to multi-epoch training, and DP-BANDMF (Choquette-Choo et al., 2023a) enabled privacy amplification. Other improvements include: extending DP-MF to adaptive optimizers (Ganesh et al., 2025), optimizing matrix factorization error (Kalinin et al., 2025), and leveraging special matrix structures for efficiency (Dvijotham et al., 2024; McMahan & Pillutla, 2025; Pillutla et al., 2025). Koloskova et al. (2023) used theoretical convergence analysis to propose rescaling different parts of their objective function; however, their results were not competitive with Choquette-Choo et al. (2023a). Kalinin et al. (2026) propose DP-λCGD, a memory-efficient correlated-noise mechanism that restricts the strategy matrix to a one-parameter family controlled by a scalar λ. To the best of our knowledge, we are the first to improve the correlated noise using the approximated model curvature (Hessian upper bound).

Using Public Data. Pretraining with public data and finetuning with DP can improve utility in DP training (Yu et al., 2022; Li et al., 2022; De et al., 2022; Chua et al., 2024; Kurakin et al., 2022). Public data can also be used to pre-condition the noisy DP-SGD gradients (Zhou et al., 2021; Yu et al., 2021b). These techniques are orthogonal to, and can be used in conjunction with, our approach. However, obtaining a large *labeled* public dataset can be very costly. One benefit of NoiseCurve is that it can take advantage of *unlabeled* public data.

Improving Convergence. DP-SGD can be seen as a composition of individual DP mechanisms applied to each weight update round, with the total privacy cost growing sublinearly in the total number of training iterations. Therefore, faster convergence results in lower privacy costs or, equivalently, lower noise scale under the same privacy budget. This line of work includes: using information from noisy history (Li et al., 2023), integrating second-order information in convex settings (Ganesh et al., 2023), and adaptively determining privacy parameters (Lee & Kifer, 2018; Bu et al., 2023). Again, these improvements are orthogonal to our work.

3 Motivation

Given a model Φ with learnable weights $\boldsymbol{w} \in \mathbb{R}^p$ and loss function $\mathcal{L}$, let $\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T$ be the weights after each update of SGD. $\boldsymbol{w}_0$ are the initial weights and $\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \eta \nabla \mathcal{L}(\boldsymbol{w}_i)$, where η is the learning rate. To simplify the discussion for this section, we assume all gradients have norm smaller than some constant ζ (e.g., the gradients are already clipped). In each iteration i, DP-SGD adds isotropic Gaussian noise $\boldsymbol{z}_i \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p)$ to each (clipped) gradient, resulting in the weight sequence $\widetilde{\boldsymbol{w}}_0, \widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, \ldots, \widetilde{\boldsymbol{w}}_T$, where $\widetilde{\boldsymbol{w}}_0 = \boldsymbol{w}_0$ and $\widetilde{\boldsymbol{w}}_{i+1} = \widetilde{\boldsymbol{w}}_i - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_i) + \boldsymbol{z}_i)$. Each Gaussian noise vector is independent: $\boldsymbol{z}_i \perp\!\!\!\perp \boldsymbol{z}_j$ when $j \neq i$.

The DP-MF line of work correlates noise across different iterations by searching for a lower triangular mixing matrix $\boldsymbol{C} \in \mathbb{R}^{T \times T}$ whose columns have norm 1. It generates noise vectors $\widetilde{\boldsymbol{z}}_0, \ldots, \widetilde{\boldsymbol{z}}_{T-1}$ from $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{T-1}$ through: $[\widetilde{\boldsymbol{z}}_0, \ldots, \widetilde{\boldsymbol{z}}_{T-1}] = [\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{T-1}]\boldsymbol{C}^{-\top}$. That is, each $\widetilde{\boldsymbol{z}}_i$ is a linear combination of $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_i$. During training, the new noise vectors are used instead, i.e., $\widetilde{\boldsymbol{w}}_{i+1} = \widetilde{\boldsymbol{w}}_i - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_i) + \widetilde{\boldsymbol{z}}_i)$. A proper choice of $\boldsymbol{C}$ can let DP-MF outperform DP-SGD at the same privacy cost (Denisov et al., 2022; Choquette-Choo et al., 2023a). In the current state-of-the-art methods (e.g., DP-BANDMF), $\boldsymbol{C}$ is obtained by solving a data-independent optimization problem (Denisov et al., 2022; Choquette-Choo et al., 2023a) to approximately minimize the sum of the squared errors of $\widetilde{\boldsymbol{w}}_1, \ldots, \widetilde{\boldsymbol{w}}_T$ (i.e., minimize $\sum_{i=0}^{T-1} ||\widetilde{\boldsymbol{z}}_i||_2^2$), subject to privacy constraints. That is, the error of the parameter estimate $\widetilde{\boldsymbol{w}}_i$ is modeled only through $||\widetilde{\boldsymbol{z}}_{i-1}||_2^2$, the amount of noise in $(i-1)^{\text{th}}$ iteration.

However, the difference between $\widetilde{\boldsymbol{w}}_i$ and $\boldsymbol{w}_i$ depends not just on the noise used in iteration $i-1$; it also depends on how prior noise had altered where the gradients are taken, i.e., $\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_{i-1})$ vs. $\nabla \mathcal{L}(\boldsymbol{w}_{i-1})$. To see this more clearly, consider the case of $i = 0$ and $i = 1$. Recall that $\boldsymbol{w}_0$ is the initial value of the weights and is the same for both private and non-private training:

$$
\begin{aligned}
\widetilde{\boldsymbol{w}}_1 - \boldsymbol{w}_1 &= (\boldsymbol{w}_0 - \eta(\nabla \mathcal{L}(\boldsymbol{w}_0) + \widetilde{\boldsymbol{z}}_0)) - (\boldsymbol{w}_0 - \eta \nabla \mathcal{L}(\boldsymbol{w}_0)) = -\eta \widetilde{\boldsymbol{z}}_0. \\
\widetilde{\boldsymbol{w}}_2 - \boldsymbol{w}_2 &= (\widetilde{\boldsymbol{w}}_1 - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_1) + \widetilde{\boldsymbol{z}}_1)) - (\boldsymbol{w}_1 - \eta \nabla \mathcal{L}(\boldsymbol{w}_1)) \\
&= -\eta \widetilde{\boldsymbol{z}}_0 - \eta \widetilde{\boldsymbol{z}}_1 - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_1) - \nabla \mathcal{L}(\boldsymbol{w}_1)) \\
&\approx -\eta \widetilde{\boldsymbol{z}}_0 - \eta \widetilde{\boldsymbol{z}}_1 - \eta \nabla^2 \mathcal{L}(\boldsymbol{w}_1)(\widetilde{\boldsymbol{w}}_1 - \boldsymbol{w}_1) = \boxed{-\eta(\widetilde{\boldsymbol{z}}_0 + \widetilde{\boldsymbol{z}}_1) + \eta^2 \nabla^2 \mathcal{L}(\boldsymbol{w}_1)\widetilde{\boldsymbol{z}}_0.}
\end{aligned}
$$

While previous works on DP-MF optimize the first term of the boxed equation ($-\eta(\widetilde{\boldsymbol{z}}_0 + \widetilde{\boldsymbol{z}}_1)$), they largely ignore the effect of the second term ($\eta^2 \nabla^2 \mathcal{L}(\boldsymbol{w}_1)\widetilde{\boldsymbol{z}}_0$), which is another source of error. The second term shows that the Hessian of the loss (i.e., model curvature) plays an important role in determining how the privacy-preserving update path $\boldsymbol{w}_0, \widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, \ldots$ diverges from the non-private path $\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$. In Section 4, we use non-convex convergence analysis to propose an alternative objective function, which incorporates an upper bound on these Hessians. In Section 5, we discuss how to estimate Hessian upper bounds for realistic deep learning settings.

4 Non-convex Convergence Analysis and New Objective for Correlated Noise

Given a function f, we say that $\hat{\boldsymbol{H}}$ is a Hessian *upper bound* on f if i) $\hat{\boldsymbol{H}}^T \hat{\boldsymbol{H}} - (\nabla^2 f)^T \nabla^2 f$, and ii) $\hat{\boldsymbol{H}} - \nabla^2 f$ are both positive semi-definite. Hessian upper bounds have two consequences that we use as assumptions in our convergence analysis:

- **Property 1**: $||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})||_2 \leq ||\hat{\boldsymbol{H}}(\boldsymbol{x} - \boldsymbol{y})||_2$.
- **Property 2** $\Big($define $||\boldsymbol{v}||_{\hat{\boldsymbol{H}}} = \sqrt{\boldsymbol{v}^T \hat{\boldsymbol{H}} \boldsymbol{v}}\Big)$: $f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{2}||\boldsymbol{y} - \boldsymbol{x}||_{\hat{\boldsymbol{H}}}^2$.

We perform a non-convex convergence analysis to see how the convergence of DP-MF (i.e., gradient updates using noise correlated across iterations) is affected by a Hessian upper bound $\hat{\boldsymbol{H}}$. Recall from Section 3 that the weight updates are $\widetilde{\boldsymbol{w}}_{i+1} = \widetilde{\boldsymbol{w}}_i - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_i) + \widetilde{\boldsymbol{z}}_i)$, where the correlated noise sequence is created as $[\widetilde{\boldsymbol{z}}_0, \ldots \widetilde{\boldsymbol{z}}_{T-1}] = [\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{T-1}]\boldsymbol{C}^{-\top}$, $\boldsymbol{z}_i$ are independent, and $\boldsymbol{C}$ is a lower-triangular noise mixing matrix. The results are shown in the following theorem.

Theorem 4.1 (Non-convex Convergence)**.** *Consider the DP-MF iteration updates: $\widetilde{\boldsymbol{w}}_t = \widetilde{\boldsymbol{w}}_{t-1} - \eta\left(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_{t-1}) + \widetilde{\boldsymbol{z}}_{t-1}\right)$ until step T, where $\boldsymbol{C}$ is an arbitrary lower triangular mixing matrix whose column norms are all 1, $\boldsymbol{Z} \sim \mathcal{N}(0, \sigma^2)^{p \times T}$, and $\widetilde{\boldsymbol{z}}_t$ is the t-th column of $\boldsymbol{Z}\boldsymbol{C}^{-\top}$ (indexed from 0). Let $\mathcal{L}$ be a non-convex loss function with Hessian upper bound $\hat{\boldsymbol{H}}$ (satisfying Properties 1 and 2)*

such that $\mathcal{L}$ has a minimum value $\mathcal{L}^*$, let the learning rate $\eta\lambda_{\max}(\hat{\boldsymbol{H}}) \leq 1$. Then:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 \leq \frac{4\mathcal{L}(\boldsymbol{w}_0)-4\mathcal{L}^*}{\eta T} + 10\eta^2\sigma^2\boxed{\operatorname{Tr}(\boldsymbol{W}\boldsymbol{X}^{-1})} \tag{1}$$

where $\boldsymbol{X} = \boldsymbol{C}^\top\boldsymbol{C}$ and the matrix $\boldsymbol{W}$ is defined as:

$$\boldsymbol{W}[i,j] = \frac{1}{T}\sum_{t=\max(i,j)+1}^{T-1}\operatorname{Tr}\left(\hat{\boldsymbol{H}}^2(\boldsymbol{I}-\eta\hat{\boldsymbol{H}})^{2t-i-j-2}\right)$$

The proof can be found in Appendix B. The boxed term in Equation 1 is the only one that depends on the correlation $\boldsymbol{C}$ (since $\boldsymbol{X} = \boldsymbol{C}^\top\boldsymbol{C}$), and has the form $\operatorname{Tr}(\boldsymbol{W}(\boldsymbol{C}^\top\boldsymbol{C})^{-1})$. If we have an upper bound $\hat{\boldsymbol{H}}$ on the true Hessians, then $\boldsymbol{W}$ is solely determined by the eigenvalues of $\hat{\boldsymbol{H}}$, since each entry of $\boldsymbol{W}$ is the trace of a polynomial in $\hat{\boldsymbol{H}}$. Thus, if we had such an $\hat{\boldsymbol{H}}$, then Theorem 4.1 suggests we should choose a correlation $\boldsymbol{C}$ to minimize the boxed part of Equation 1 subject to privacy constraints. We next explain how to integrate this idea into the DP-MF framework and then consider the problem of estimating $\hat{\boldsymbol{H}}$ in Section 5.

4.1 Integration with DP-MF

DP-BANDMF (Choquette-Choo et al., 2023a) is a state-of-the-art DP-MF framework with privacy amplification via batches. In the version of DP they use, two datasets D and D' are neighbors if one can be obtained from the other by blanking out one record.

Definition 4.2 (Dwork et al. (2006))**.** Given privacy parameters $\epsilon > 0$ and $\delta \in [0,1]$, a mechanism (randomized algorithm) M satisfies (ϵ,δ)-differential privacy if for all pairs of neighboring datasets D, D' and all subsets $S \subset \text{range}(M)$, $P(M(D) \in S) \leq e^\epsilon P(M(D') \in S) + \delta$.

Algorithm 1: DP-BandMF Training

Input: Initial model f with parameters $\boldsymbol{w}_0 \in \mathbb{R}^p$, dataset $\mathcal{D}$, mixing matrix $\boldsymbol{C} \in \mathbb{R}^{T\times T}$, clip norm ζ, noise $\boldsymbol{z} \in \mathbb{R}^{T\times p}$ with entries $\sim \mathcal{N}(0,\sigma^2)$, learning rate η.
Output: DP trained model $\boldsymbol{w}_T$.

for $t = 1, 2, \ldots, T$ **do**
  Create a DP-BandMF batch $B_t \subseteq \mathcal{D}$
  $g_i = \nabla\ell(f(x_i; w_{t-1}), y_i)$ for $(x_i, y_i) \in B_t$
  $\tilde{g}_i = \min(1, \zeta/\|g_i\|_2)\, g_i$
  $w_t = w_{t-1} - \eta\left(\sum_{i=1}^{|B_t|}\tilde{g}_i + (\boldsymbol{C}^{-1}\boldsymbol{z})_{[t,:]}\right)$
end for

Given the privacy parameters ϵ, δ, and a desired iteration count T, DP-BANDMF optimizes $\boldsymbol{C}$ to minimize its objective function under the following constraints for privacy: (1) each diagonal entry of $\boldsymbol{X} = \boldsymbol{C}^\top\boldsymbol{C}$ is 1, (2) $\boldsymbol{X}$ is positive definite, and (3) banded (i.e., $\boldsymbol{X}_{i,j} = 0, \forall|i-j| \geq b$, where the band size b is a tunable hyperparameter and allows privacy amplification). We can simply replace the objective function of DP-BANDMF with ours and leave the rest of the framework unchanged (batch formation, privacy amplification/accounting), which gives the following optimization problem:

$$\begin{aligned} \boldsymbol{X} \quad &\leftarrow \arg\min \operatorname{Tr}(\boldsymbol{W}\boldsymbol{X}^{-1}) \\ \text{s.t.} \quad &\texttt{diag}(\boldsymbol{X}) = \mathbf{1} \quad \text{and} \quad \boldsymbol{X} \succ 0 \quad \text{and} \quad \boldsymbol{X}_{i,j} = 0, \forall|i-j| \geq b \end{aligned} \tag{P1}$$

where $\boldsymbol{W}$ is defined in Theorem 4.1. As in the original DP-BANDMF (Choquette-Choo et al., 2023a), this convex optimization problem can be solved by optimizers like L-BFGS (Nocedal, 1980) with projections onto the feasible region to enforce $\texttt{diag}(\boldsymbol{X}) = \mathbf{1}$, and starting with an initial guess for $\boldsymbol{X}$ that is symmetric and positive definite. Following DP-BANDMF, once an $\boldsymbol{X} = \boldsymbol{C}^\top\boldsymbol{C}$ is obtained, one can use the Cholesky decomposition to recover $\boldsymbol{C}$ as a lower triangular matrix.

Pseudocode for DP-BANDMF training is shown in Algorithm 1. To use NoiseCurve, one only needs to replace the $\boldsymbol{C}$ used by DP-BANDMF with the $\boldsymbol{C}$ obtained via Problem P1 (the privacy guarantees and everything else remain the same). We briefly explain DP-BANDMF batch creation and noise addition, which are the key differences from standard DP-SGD (for completeness, Appendix C provides more details, or see Choquette-Choo et al. (2023a) for full details). **Batch Creation:** given a band hyperparameter b and batch size $|B|$, the dataset D is initially divided into b equal sized partitions $D_1, \ldots, D_b$. During model training, the batch B_t for iteration t is obtained by (1) selecting partition D_ℓ, where $\ell = 1 + (t \mod b)$ then (2) setting the batch B_t to be a random sample from D_ℓ of size $|B|$ (see Algorithm 4 in Appendix C for pseudocode). **Noise Addition:** using $\boldsymbol{C}$ to obtain correlated noise $\widetilde{\boldsymbol{z}}_0, \ldots, \widetilde{\boldsymbol{z}}_{T-1}$ in an online manner is essentially Gaussian elimination (Appendix C). Our new $\boldsymbol{C}$ from Problem P1 can be directly integrated into Algorithm 1's noise addition step.

5 Estimating the Hessian Upper Bound

The theory behind NoiseCurve requires information about a matrix $\hat{\boldsymbol{H}}$ that upper-bounds the Hessian of the loss along the training path, i.e., $\hat{\boldsymbol{H}} \succeq \nabla^2\mathcal{L}(\widetilde{\boldsymbol{w}}_t)$ and $\hat{\boldsymbol{H}}^\top\hat{\boldsymbol{H}} \succeq (\nabla^2\mathcal{L}(\widetilde{\boldsymbol{w}}_t))^\top\nabla^2\mathcal{L}(\widetilde{\boldsymbol{w}}_t)$ for every t. In other words, instead of upper bounding the Hessian of the loss at all possible parameter settings of the network, we just need to upper bound it for a "good" region of the parameter space. In practice, obtaining such an $\hat{\boldsymbol{H}}$ exactly is difficult. Therefore, in this section we consider: (1) how to obtain a good *proxy* of such an $\hat{\boldsymbol{H}}$, (2) the robustness of NoiseCurve to errors in estimating $\hat{\boldsymbol{H}}$, (3) using the Lanczos method to efficiently compute the objective function in Problem P1 without materializing $\hat{\boldsymbol{H}}$ or its eigenspectrum.

5.1 Estimating Upper Bounds from Public Data

Since DP-MF noise correlations must be determined before private training starts, information about $\hat{\boldsymbol{H}}$ is only estimated once (at the beginning). To minimize privacy impact, we rely on an unlabeled public dataset, since that is more common and easier to collect than labeled data. We propose a simple heuristic. First, we assign random labels to examples in the public dataset and use them to pre-train the model. At the resulting parameter values, the basic procedure is to compute the Hessian on the public dataset, compute its eigenspectrum, and set negative eigenvalues to 0 (since a Hessian upper bound needs to be at least positive semi-definite). Later, in Section 5.3, we present optimizations. The resulting Hessian information is used as our proxy for the Hessian upper bound $\hat{\boldsymbol{H}}$ in computing the matrices in Theorem 4.1 and Problem P1.

To justify this random-label pretraining approach, we provide intuition supporting one of the upper bound conditions, $\hat{\boldsymbol{H}} \succeq \nabla^2\mathcal{L}$, by analyzing logistic regression and drawing on empirical observations in prior work (Papyan, 2019). Then, we empirically evaluate how well random-label pretraining succeeds in finding an upper bound that satisfies both the $\hat{\boldsymbol{H}} \succeq \nabla^2\mathcal{L}$ and $\hat{\boldsymbol{H}}^\top\hat{\boldsymbol{H}} \succeq (\nabla^2\mathcal{L})^\top\nabla^2\mathcal{L}$ conditions, as measured by a dominance certificate. Note that even if $\hat{\boldsymbol{H}}$ is incorrectly estimated, it does not degrade privacy and only affects utility. While large errors in $\hat{\boldsymbol{H}}$ will surely ruin model accuracy, Section 5.2 empirically shows that even with moderate errors in $\hat{\boldsymbol{H}}$, NoiseCurve still provides accuracy improvements over not using any Hessian information (DP-SGD or DP-BANDMF).

Intuition: Why Would Using Random Labels Work? Consider a classification problem with k labels. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be the feature vectors and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ the corresponding one-hot encoded labels. Consider a network that consists of a vector-valued function $\mathbf{f} = [f_1, \ldots, f_k]$ that gets passed into a softmax and trained with cross-entropy loss $\mathcal{L}$. Given weights $\boldsymbol{w}$, the network output is a k-dimensional probability vector that we can write as $\boldsymbol{p}(\mathbf{x}; \boldsymbol{w}) \equiv \mathrm{softmax}(\mathbf{f}(\mathbf{x}; \boldsymbol{w}))$ or $(p_1(\mathbf{x}; \boldsymbol{w}), \ldots, p_k(\mathbf{x}; \boldsymbol{w})) \equiv \mathrm{softmax}(f_1(\mathbf{x}; \boldsymbol{w}), \ldots, f_k(\mathbf{x}; \boldsymbol{w}))$. The Hessian of the loss is:

$$\nabla^2_{\boldsymbol{w}}\mathcal{L} = \underbrace{\sum_{i=1}^{n}\sum_{\ell=1}^{k}(p_\ell(\mathbf{x}_i; \boldsymbol{w}) - y_i[\ell])\frac{\partial^2 f_\ell(\mathbf{x}_i; \boldsymbol{w})}{\partial \boldsymbol{w}^2}}_{\text{Piece H}} + \underbrace{\sum_{i=1}^{n}\frac{\partial \mathbf{f}(\mathbf{x}_i; \boldsymbol{w})}{\partial \boldsymbol{w}}^\top \left(\mathtt{diag}(\boldsymbol{p}(\mathbf{x}_i; \boldsymbol{w})) - \boldsymbol{p}(\mathbf{x}_i; \boldsymbol{w})\boldsymbol{p}(\mathbf{x}_i; \boldsymbol{w})^\top\right)\frac{\partial \mathbf{f}(\mathbf{x}_i; \boldsymbol{w})}{\partial \boldsymbol{w}}}_{\text{Piece G}}.$$

In the simple case of logistic regression ($k = 2$, $f_2 = 0$, and $\mathrm{softmax}(f_1, 0) = \sigma(f_1) = p_1$), it is well-known that $\nabla^2_{\boldsymbol{w}}\mathcal{L} = \boldsymbol{X}^\top\boldsymbol{D}_{\boldsymbol{w}}\boldsymbol{X}$, where $\boldsymbol{X}$ is the matrix whose rows are the feature vectors and $\boldsymbol{D}_{\boldsymbol{w}}$ is the $n \times n$ diagonal matrix whose (i, i)-th entry is $p_1(\mathbf{x}_i; \boldsymbol{w})(1 - p_1(\mathbf{x}_i; \boldsymbol{w})) \leq 1/4$. In other words, Piece H above equals 0 and the Hessian of the loss equals Piece G, which becomes $\boldsymbol{X}^\top\boldsymbol{D}_{\boldsymbol{w}}\boldsymbol{X}$. This is upper bounded by $\boldsymbol{X}^\top(\frac{1}{4}\boldsymbol{I}_n)\boldsymbol{X}$, since $\boldsymbol{D}_{\boldsymbol{w}}$ is upper bounded by $\frac{1}{4}\boldsymbol{I}_n$ (i.e., $\boldsymbol{D}_{\boldsymbol{w}} \preceq \frac{1}{4}\boldsymbol{I}_n$ or equivalently, $\frac{1}{4}\boldsymbol{I}_n - \boldsymbol{D}_{\boldsymbol{w}}$ is positive semidefinite). This Hessian upper bound, $\boldsymbol{X}^\top(\frac{1}{4}\boldsymbol{I}_n)\boldsymbol{X}$, is approximately what one would get when training with random labels, since a logistic regression that is maximally uncertain outputs 1/2 for all probabilities (i.e., $p_1(\mathbf{x}_i; \boldsymbol{w}) = 1/2$), so $\boldsymbol{D}_{\boldsymbol{w}} = \frac{1}{4}\boldsymbol{I}_n$. This

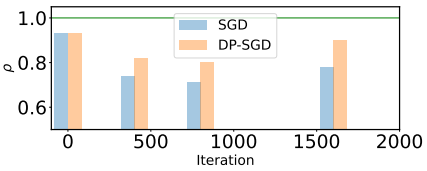


Figure 2: For different $\boldsymbol{w}_t$ encountered during training, we compute Hessian $\boldsymbol{H}_t$, and finetune it with random-label validation data to get $\hat{\boldsymbol{H}}_t$. $\rho(\boldsymbol{H}_t, \hat{\boldsymbol{H}}_t) < 1$ shows that the estimated $\hat{\boldsymbol{H}}$ serves as a true upper bound. This experiment uses the same training setup as Figure 1.

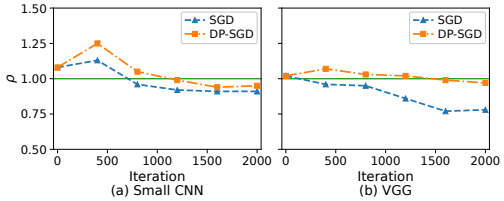


Figure 3: $\rho(\boldsymbol{H}_t, \hat{\boldsymbol{H}})$ for one-time random-label pre-training of $\hat{\boldsymbol{H}}$ on TinyImageNet. Model Hessians $\boldsymbol{H}_t$ (architecture details in Appendix A.1) during SGD/DP-SGD training are almost always upper-bounded by the estimated $\hat{\boldsymbol{H}}$.

simple example shows that, at least for logistic regression, training with random labels would tend to result in $\hat{\boldsymbol{H}} \succeq \nabla^2_{\boldsymbol{w}} \mathcal{L}$. Next, we extend this intuition to nonconvex models.

Let us examine what happens to neural network Hessians during *supervised* training and what this would imply for Hessian upper bounds. Papyan (2019) trained a VGG11 network on CIFAR10 and studied how the eigenspectrum of the Hessian, Piece H, and Piece G changed during training (for reference, the results appear in Figure 6 of that paper). Recall that NoiseCurve only needs the eigenspectrum of a Hessian upper bound, not the full matrix. The results of Papyan (2019) show that the large positive eigenvalues of the Hessian keep decreasing relative to the bulk as training progresses, suggesting that the eigenspectrum of earlier epochs (when a network is more uncertain) may be dominating the eigenspectrum of later epochs. Furthermore, in the first half of the epochs, the positive part of the Hessian eigenspectrum mostly came from Piece G (another similarity to logistic regression). Hence, the eigenspectrum of Piece G in the earlier epochs may be dominating the eigenspectrum of the Hessian in later epochs. Also, evaluating Piece G does not require labels.

Empirical support for random-label pretraining. We next empirically show that (1) using random labels tends to make Hessians "bigger" even in nonconvex cases, and (2) estimating the upper bound once before private training is fairly reliable. The following theorem introduces a dominance certificate, $\rho(\boldsymbol{H}, \hat{\boldsymbol{H}})$, which quantifies the extent to which a matrix $\hat{\boldsymbol{H}}$ is an upper bound on $\boldsymbol{H}$.

Theorem 5.1 (Dominance Certificate)**.** *Let $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ be two symmetric real matrices, with $\hat{\boldsymbol{H}}$ positive definite. Define (using the convention 0/0=0) two dominance certificates $\rho(\boldsymbol{H}, \hat{\boldsymbol{H}})$ as*

$$\rho_1(\boldsymbol{H}, \hat{\boldsymbol{H}}) := \max_{\boldsymbol{v} \neq \boldsymbol{0}} \frac{\|\boldsymbol{H}\boldsymbol{v}\|_2}{\|\hat{\boldsymbol{H}}\boldsymbol{v}\|_2} \quad \textit{and} \quad \rho_2(\boldsymbol{H}, \hat{\boldsymbol{H}}) := \max_{\boldsymbol{v} \neq \boldsymbol{0}} \frac{\boldsymbol{v}^\top \boldsymbol{H}\boldsymbol{v}}{\boldsymbol{v}^\top \hat{\boldsymbol{H}}\boldsymbol{v}}$$

Then, $\hat{\boldsymbol{H}}$ upper bounds $\boldsymbol{H} \iff \rho(\boldsymbol{H}, \hat{\boldsymbol{H}}) := \max(\rho_1, \rho_2) \leq 1$.

Thus, if $\rho(\boldsymbol{H}, \hat{\boldsymbol{H}}) \leq 1$, then $\hat{\boldsymbol{H}}$ is an upper bound on $\boldsymbol{H}$. If $\rho > 1$, it indicates by how much we "missed" the upper bound (i.e., how much $\hat{\boldsymbol{H}}$ needs to be scaled in order to become an upper bound). The quantities ρ_1 and ρ_2 can be computed efficiently using Lanczos procedure (Lanczos, 1950), which also allows for assessing the upper bound relation without instantiating the full Hessian matrix. With the dominance certificate as a tool, we empirically evaluate our approach.

Question 1: Does random label fine-tuning produce an upper bound? Given a weight vector $\boldsymbol{w}_t$ and associated Hessian $\boldsymbol{H}_t$, what happens if, starting from $\boldsymbol{w}_t$, we fine-tune the model with randomly-labeled in-distribution features (e.g., "public" randomly-labeled validation data) to get a $\hat{\boldsymbol{H}}_t$? Does $\hat{\boldsymbol{H}}_t$ upper bound $\boldsymbol{H}_t$? We experiment with different $\boldsymbol{w}_t$ we encounter during SGD and DP-SGD training and compute the dominance certificate $\rho(\boldsymbol{H}_t, \hat{\boldsymbol{H}}_t)$. Figure 2 shows that $\rho(\boldsymbol{H}_t, \hat{\boldsymbol{H}}_t) < 1$ in all cases we tested, indicating that random-label training indeed appears to make Hessians "bigger".

*Question 2: Does the $\hat{\boldsymbol{H}}$ computed at the **beginning of training** from **public data** serve as an upper bound for later training stages as well?* $\hat{\boldsymbol{H}}$ must be computed before private training (in order to determine noise correlation structure), using public data (not necessarily in-distribution data). We evaluate whether (1) using public data and (2) estimating $\hat{\boldsymbol{H}}$ only once provide a good upper bound.

We compute $\hat{\boldsymbol{H}}$ from TinyImageNet and compare it to the Hessians $\boldsymbol{H}_t$ encountered during SGD and DP-SGD training over CIFAR-10 (i.e., TinyImageNet serves as the public dataset which is similar but still out-of-distribution to CIFAR-10). The dominance certificate is shown in Figure 3. In early iterations, $\hat{\boldsymbol{H}}$ is off by a factor of at most 1.25, but otherwise, it serves as a true upper bound. The result is consistent with the experiments of Papyan (2019), which showed that early training iterations have larger eigenspectra.

Additionally, our evaluation (Section 6) consistently shows that NoiseCurve outperforms the baselines in all setups we studied, which is a strong indication that our proposed approach works.

5.2 Robustness of NoiseCurve to Errors in the Hessian Upper Bound $\hat{\boldsymbol{H}}$

Random-label pretraining appears to be an effective heuristic, and has intuitive backing, but it is still a heuristic and can have errors. Further errors can come from concept drift (the unlabeled public data is out of distribution with respect to the labeled private data). Finally, the upper bound $\hat{\boldsymbol{H}}$ may also be loose. We next study how robust NoiseCurve is to errors in $\hat{\boldsymbol{H}}$.

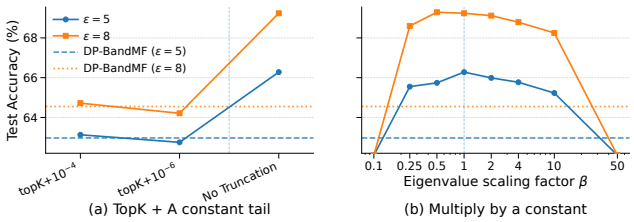


Figure 4: Test accuracy when errors are introduced into $\hat{\boldsymbol{H}}$ (one run).

First, we consider what happens when the bulk of the eigenvalues of $\hat{\boldsymbol{H}}$ are incorrect. To simulate this scenario, we keep the top $k = 2,000$ eigenvalues from $\hat{\boldsymbol{H}}$ and replace the rest ($\approx 998,000$ eigenvalues) with a small constant tail (e.g., 10^{-4} or 10^{-6}), resulting in an inaccurate and flat eigenspectrum tail. The resulting test errors for NoiseCurve (same experimental settings as Figure 1) compared to DP-BandMF are shown in Figure 4(a). Without this error in $\hat{\boldsymbol{H}}$, NoiseCurve clearly outperforms DP-BandMF and DP-SGD ("No Truncation"). Even with the error, the performance is competitive with DP-BandMF.

We next study scaling errors. What happens when $\hat{\boldsymbol{H}}$ is rescaled by some factor β? Since private and non-private neural network training involves hyperparameter tuning, we expect the scaling error to be partially mitigated by the learning rate η. To see why, note that the objective in Problem P1 can be multiplied by η^2 without affecting the solution for $\boldsymbol{X}$. When the eigenvalues of $\beta\hat{\boldsymbol{H}}$ are used to construct $\boldsymbol{W}$ in Theorem 4.1, we see each entry of $\boldsymbol{W}$ becomes a function of $\eta\beta\hat{\boldsymbol{H}}$. Hence, increases or decreases in β can be compensated by appropriate changes to the learning rate (within reason). We also study this phenomenon experimentally in Figure 4(b) and see that the testing accuracy of NoiseCurve is relatively flat to most rescalings of $\hat{\boldsymbol{H}}$, but can collapse under extreme conditions.

We also evaluated an $\hat{\boldsymbol{H}}$ computed from unlabeled data at a random initialization (omitting the random-label pre-training). Without pre-training, $\hat{\boldsymbol{H}}$ is probably not a Hessian upper bound. The results, in Appendix A.5, show that the accuracy is marginally better than DP-SGD and DP-BandMF.

Finally, we study concept drift. We note that all experiments in Section 6 involve concept drift, as $\hat{\boldsymbol{H}}$ is computed from TinyImageNet on vision tasks and WikiText for NLP while the private datasets are CIFAR-10, ChestX-Ray14 and MNLI. In Section 6, the testing accuracy of NoiseCurve consistently outperforms DP-BandMF and DP-SGD. Aside from those accuracy experiments, here we perform direct comparisons of the eigenspectra. Figure 5 shows the top 10,000 eigenvalues of $\hat{\boldsymbol{H}}$ estimated for two different CNNs using random-label pre-training on four different computer vision datasets: CIFAR-10, TinyImageNet, StreetView House Numbers (SVHN) and ChestX-Ray14. There is clearly a very strong match in this part of the eigenspectrum even though the domains are very different. We speculate that this is another instance of the Deep

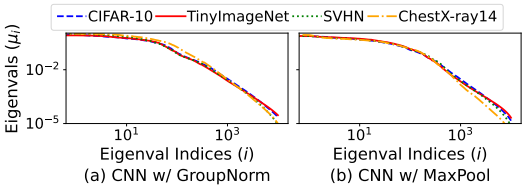


Figure 5: Largest 10,000 eigenvalues of four datasets estimated on two CNNs. Setup details are in Appendix A.1.

Image Prior effect (Ulyanov et al., 2018), in which modern architectures incorporate a lot of inductive bias for images and so, regardless of the image domain or class labels, the lower network layers implicitly perform similar operations (e.g., edge detection, corner detection, identifying textures).

5.3 Efficient Computation of the NoiseCurve Objective

We next consider efficient computation of the matrices in the noise correlation objective function (Problem P1) which depend on the eigenspectrum of $\hat{\boldsymbol{H}}$. We note that even the eigenspectrum of a 70B parameter DeepSeek model can be computed with enough hardware (Granziol, 2025), but efficient techniques that avoid directly computing the eigenspectrum are preferable whenever they are possible. To do this, we use the Lanczos method (Lanczos, 1950) and efficient Hessian-vector products (HVP; Dagréou et al. (2024)).

Recall that in Theorem 4.1, each entry (i, j) of $\boldsymbol{W}$ can be written as:

$$\boldsymbol{W}[i,j] = \frac{1}{T} \sum_{t=\max(i,j)+1}^{T-1} \operatorname{Tr}\left(\hat{\boldsymbol{H}}^2 (\boldsymbol{I} - \eta \hat{\boldsymbol{H}})^{2t-i-j-2}\right)$$

Therefore, if we define the *target moments* ω_m as:

$$\omega_m = \operatorname{Tr}(\hat{\boldsymbol{H}}^2 (\boldsymbol{I} - \eta \hat{\boldsymbol{H}})^m), \text{ for } m = 0, \ldots, 2T-2$$

then every entry of $\boldsymbol{W}$ can be recovered from these $2T-1$ values (recall T is the intended number of weight update rounds during private training). Note that if the eigendecomposition of $\hat{\boldsymbol{H}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ then $\operatorname{Tr}(\hat{\boldsymbol{H}}^2 (\boldsymbol{I} - \eta \hat{\boldsymbol{H}})^{2t-i-j})$ is equal to $\operatorname{Tr}(\boldsymbol{U}\boldsymbol{D}^2\boldsymbol{U}^\top (\boldsymbol{U}\boldsymbol{U}^\top - \eta \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top)^{2t-i-j}) = \operatorname{Tr}(\boldsymbol{D}^2 (\boldsymbol{I} - \eta \boldsymbol{D})^{2t-i-j})$ so only the eigenvalues matter for the target moments ω_m.

Define $f(\lambda) = \max(\lambda, 0)$ and for any symmetric matrix $\boldsymbol{W}$, let $\bar{f}(\boldsymbol{W})$ be the application of f to the eigenvalues of $\boldsymbol{W}$ (i.e., $\bar{f}(\boldsymbol{W})$ takes the eigen-decomposition of $\boldsymbol{W}$ and modifies it by changing negative eigenvalues to 0). Let $\boldsymbol{H}$ be the Hessian of the network after random-label pretraining. Then $\hat{\boldsymbol{H}} = \bar{f}(\boldsymbol{H})$ and the target moments can be expressed as:

Algorithm 2: Efficient Computation

Input: Hessian-vector-product function `hvp`, Thresholding function f, Learning rate η, Number of iterations T, Number of probes n, Lanczos steps L.
Output: Moment vector $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{2T-2})$.

for $i = 1, 2, \ldots, n$ **do**
  Sample $\boldsymbol{v}_i \sim \mathcal{N}(0, \boldsymbol{I}_p)$ and $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i / \|\boldsymbol{v}_i\|_2$
  $\boldsymbol{T}_i \leftarrow \texttt{lanczos}(\texttt{hvp}(\boldsymbol{v}_i), L)$
  $\boldsymbol{Q}_i, \Lambda_i \leftarrow \texttt{eig}(\boldsymbol{T}_i)$
  $a_{ij} = (\boldsymbol{e}_1^\top \boldsymbol{q}_{ij})^2$
end for
for $m = 0, 1, \ldots, 2T-2$ **do**
  $\omega_m \leftarrow \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{L} a_{ij} f(\lambda_{ij})^2 \left(1 - \eta f(\lambda_{ij})\right)^m$
end for

$$\omega_m = \operatorname{Tr}\left(\bar{f}(\boldsymbol{H})^2 (\boldsymbol{I} - \eta \bar{f}(\boldsymbol{H}))^m\right)$$

We estimate these moments using stochastic Lanczos quadrature. Given an initial probe vector $\boldsymbol{v}$, the Lanczos method (Lanczos, 1950) constructs an orthonormal basis for the Krylov subspace generated by $\boldsymbol{v}, \boldsymbol{H}\boldsymbol{v}, \ldots, \boldsymbol{H}^{L-1}\boldsymbol{v}$, using only HVPs, which does not materialize the Hessian (Dagréou et al., 2024). After L steps, it outputs a small tridiagonal matrix $\boldsymbol{T} \in \mathbb{R}^{L \times L}$, which is the projection of $\boldsymbol{H}$ onto this Krylov subspace. Given a function g, the eigenvalues and eigenvectors of $\boldsymbol{T}$ define a Gaussian quadrature rule for approximating quantities of the form $\boldsymbol{v}^\top g(\boldsymbol{H})\boldsymbol{v}$. Averaging this estimate over random probes gives a trace estimate for $\operatorname{Tr}(g(\boldsymbol{H}))$. We perform this procedure for m different g functions $g_m(\lambda) \equiv f(\lambda)^2(1 - \eta f(\lambda))^m, m = 0, \ldots, 2T-2$, as shown in Algorithm 2.

6 Evaluation Results

We compare our proposed NoiseCurve to DP-BandMF and DP-SGD under various privacy budgets ϵ, with $\delta = 10^{-5}$. For vision tasks, we use CIFAR-10 (Krizhevsky et al., 2009) and ChestX-ray14 (Wang et al., 2017) datasets for training, and we trained/finetuned a small CNN, ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2015), and ViT (Dosovitskiy et al., 2020). For all vision tasks, we use TinyImageNet as the public dataset for pre-training. For the language task, we finetune

a pretrained RoBERTa (Liu et al., 2019) on MNLI (Wang et al., 2019) using LoRA (Hu et al., 2022), and use WikiText as the public data. We split 10% of data samples from training sets as validation sets and **tune hyperparameters on validation sets** (no tuning on the test sets). We additionally show the result of hyperparameter tuning on the test set in Appendix A.2, where NoiseCurve still consistently outperforms competitors. All the reported accuracy numbers here are averaged across 6 runs. Further details on the setup can be found in Appendix A.1.

Table 1: Test mean accuracy and standard deviation (over 6 runs).

(a) Last layer finetuning of ResNet152 (convex loss). Non-private training accuracy is 82.01%.

ϵ	DP-SGD	DP-BANDMF	NoiseCurve
1	75.88 ± 0.16	75.92 ± 0.12	**76.70 ± 0.19**
2	77.14 ± 0.17	77.33 ± 0.09	**78.54 ± 0.15**
5	78.31 ± 0.23	78.35 ± 0.07	**80.31 ± 0.22**

(b) Full finetuning of a small CNN (nonconvex loss). Non-private training accuracy is 72.99%.

ϵ	DP-SGD	DP-BANDMF	NoiseCurve
1	48.67 ± 0.26	49.17 ± 0.22	**50.31 ± 0.30**
2	51.82 ± 0.3	52.15 ± 0.64	**54.83 ± 0.36**
5	54.64 ± 0.76	55.39 ± 0.66	**58.59 ± 0.23**

Small Models. We first compare our method with DP-SGD and DP-BANDMF on CIFAR-10 classification with two models: (1) finetuning the last layer of an ImageNet-pretrained ResNet152, where the loss is convex, and (2) fully finetuning a small CNN, whose loss is nonconvex. Table 1 summarizes the results: NoiseCurve consistently outperformed the baselines in all setups we studied, and *this result is statistically significant*. Specifically, for any setting of ϵ and architecture, each of the 6 runs of NoiseCurve had higher accuracy than each of the runs of DP-BANDMF and DP-SGD. Statistical significance can be assessed using a permutation test: if we mix the 12 runs from each row (6 from NoiseCurve and 6 from DP-BANDMF) and reassign 6 each to NoiseCurve and DP-BANDMF, there are $12!/(6!6!)$ ways of reassigning. However, there is only one assignment that makes all 6 runs of NoiseCurve have higher accuracies than DP-BANDMF (which is what we observe). Hence, the p-value is $\frac{6!6!}{12!} \approx 0.001$ for every ϵ setting and every architecture (ResNet and CNN). Put another way, every row in Table 1 is statistically significant.

Larger Models. Figure 1 (in Section 1) compares NoiseCurve to DP-BANDMF and DP-SGD on the VGG training setup used by Choquette-Choo et al. (2023b) on CIFAR-10. Across all the ϵ we tested, NoiseCurve outperformed all runs of DP-BANDMF (a statistically significant result with p-value $\lesssim 0.001$ under permutation testing).

We also experimented with finetuning ViT with LoRA on ChestX-ray14 (Table 2a). The ChestX-ray14 experiment represents a case when the public data (TinyImageNet; natural images) is fairly different from the target data (ChestX-ray14; medical images). The results are shown in Table 2a. For $\epsilon = 1, 2$ all methods had AUC near 0.5 (i.e., they were no better than random guessing). However, all runs of NoiseCurve at $\epsilon = 5$ outperformed all runs of the competitors (similarly, p-value ≈ 0.001 under permutation testing), and same with $\epsilon = 8$ (also p-value ≈ 0.001 under permutation testing). The results show that, even under a significant domain shift (TinyImageNet vs. ChestX-ray14), NoiseCurve was able to improve the accuracy over the baselines.

We additionally tried using the validation set of CIFAR-10 and ChestX-ray14 as the public dataset, which represents cases when a public dataset that is highly similar to the target data is available. While the accuracy improved with better public data as expected, the additional improvement was only around 1% for CIFAR-10 and was a moderate 2% for ChestX-ray14. The result implies that a close dataset like TinyImageNet for CIFAR-10 is good enough as a public dataset. TinyImageNet also serves as a reasonable public dataset for ChestX-ray14, although the accuracy could further improve with a public dataset that is more similar.

Table 2b summarizes our results for the language task, MNLI. This is another setting where there is a significant domain shift between the private data (MNLI) and the public dataset (WikiText). As in vision tasks, *NoiseCurve consistently outperforms both baselines*, showing that NoiseCurve benefits not only vision models but also language models. Again, each of the six runs of NoiseCurve at $\epsilon = 2$ scored higher than all of the competitor runs for a p-value of $\frac{6!6!}{12!} \approx 0.001$ under permutation testing, and the same goes for $\epsilon = 5$ and $\epsilon = 8$. Hence, all of the results are statistically significant.

Table 2: Test mean accuracy and standard deviation (over 6 runs) for $\delta = 10^{-5}$. For ChestX-ray14, we use mean AUROC as its accuracy metric.

(a) ChestX-ray14 dataset (ViT, LoRA finetuning). Non-private training accuracy is 73.63.

ϵ	DP-SGD	DP-BANDMF	NoiseCurve
5	58.16 ± 0.62	59.32 ± 0.27	**63.14 ± 0.54**
8	59.21 ± 0.29	62.55 ± 0.43	**65.23 ± 0.67**

(b) MNLI dataset (RoBERTa, LoRA finetuning). Non-private training accuracy is 83.73.

ϵ	DP-SGD	DP-BANDMF	NoiseCurve
2	46.17 ± 0.93	50.72 ± 0.41	**55.31 ± 0.66**
5	54.73 ± 2.64	63.8 ± 1.37	**68.54 ± 0.49**
8	63.77 ± 0.7	69.57 ± 0.82	**74.68 ± 0.77**

7 Discussion and Limitations

We propose NoiseCurve, which improves the quality of cross-iteration noise correlation through model curvature information. Although NoiseCurve achieves non-trivial accuracy improvements over SOTA baselines, we recognize a few limitations that require further study.

1. NoiseCurve uses random-label pretraining to estimate an upper bound $\hat{\boldsymbol{H}}$. Although this heuristic worked consistently well, an interesting open problem is whether this can be proven mathematically under minimal assumptions.

2. Since the Hessian eigenspectra tend to shrink as training progresses (Papyan, 2019), NoiseCurve could benefit from periodically re-estimating a tighter Hessian upper bound as training progresses, and using the updated estimates to dynamically adjust noise correlations. This is an interesting direction for future work.

Acknowledgments

This work was supported by the US National Science Foundation under Awards CNS-2317232, CNS-2349610, and CCF-2529883. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Automatic clipping: Differentially private deep learning made easier and stronger. *Advances in Neural Information Processing Systems*, 36:41727–41764, 2023.

Choquette-Choo, C. A., Ganesh, A., McKenna, R., McMahan, H. B., Rush, J. K., Thakurta, A. G., and Xu, Z. (amplified) banded matrix factorization: A unified approach to private training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=zEm6hF97Pz.

Choquette-Choo, C. A., McMahan, H. B., Rush, K., and Thakurta, A. Multi-epoch matrix factorization mechanisms for private machine learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023b.

Chua, L., Ghazi, B., Huang, Y., Kamath, P., Kumar, R., Liu, D., Manurangsi, P., Sinha, A., and Zhang, C. Mind the privacy unit! user-level differential privacy for language model fine-tuning. *arXiv preprint arXiv:2406.14322*, 2024.

Dagréou, M., Ablin, P., Vaiter, S., and Moreau, T. How to compute hessian-vector products? In *The Third Blogpost Track at ICLR 2024*, 2024.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Denisov, S., McMahan, H. B., Rush, J., Smith, A., and Guha Thakurta, A. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5910–5924. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/271ec4d1a9ff5e6b81a6e21d38b1ba96-Paper-Conference.pdf.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

Dvijotham, K. D., McMahan, H. B., Pillutla, K., Steinke, T., and Thakurta, A. Efficient and near-optimal noise generation for streaming differential privacy. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 2306–2317, 2024. doi: 10.1109/FOCS61266.2024.00135.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pp. 486–503, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540345469. doi: 10.1007/11761679_29. URL https://doi.org/10.1007/11761679_29.

Ganesh, A., Haghifam, M., Steinke, T., and Thakurta, A. Faster differentially private convex optimization via second-order methods. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Ganesh, A., McMahan, B., and Thakurta, A. On design principles for private adaptive optimizers, 2025. URL https://arxiv.org/abs/2507.01129.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Granqvist, F., Song, C., Cahill, A., van Dalen, R., Pelikan, M., Chan, Y. S., Feng, X., Krishnaswami, N., Jina, V., and Chitnis, M. pfl-research: simulation framework for accelerating research in Private Federated Learning. *arXiv preprint arXiv:2404.06430*, April 2024. doi: 10.48550/arXiv.2404.06430.

Granziol, D. Hessformer: Hessians at foundation scale, 2025. URL https://arxiv.org/abs/2505.11564.

Gu, X., Kamath, G., and Wu, Z. S. Choosing public datasets for private machine learning via gradient subspace distance. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*, pp. 879–900. IEEE, 2025. doi: 10.1109/SATML64287.2025.00054. URL https://doi.org/10.1109/SaTML64287.2025.00054.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. URL https://arxiv.org/abs/1502.01852.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kairouz, P., Mcmahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5213–5225. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kairouz21b.html.

Kalinin, N. P., McKenna, R., Upadhyay, J., and Lampert, C. H. Back to square roots: An optimal bound on the matrix factorization error for multi-epoch differentially private sgd. *arXiv preprint arXiv:2505.12128*, 2025.

Kalinin, N. P., McKenna, R., Pagh, R., and Lampert, C. H. Dp-λcgd: Efficient noise correlation for differentially private model training, 2026. URL https://arxiv.org/abs/2601.22334.

Koloskova, A., McKenna, R., Charles, Z., Rush, K., and McMahan, B. Gradient descent with linearly correlated noise: theory and applications to differential privacy. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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

Kurakin, A., Song, S., Chien, S., Geambasu, R., Terzis, A., and Thakurta, A. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.

Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950. doi: 10.6028/jres.045.019.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lee, J. and Kifer, D. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget, 2018. URL https://arxiv.org/abs/1808.09501.

Li, T., Zaheer, M., Liu, K., Reddi, S. J., McMahan, H. B., and Smith, V. Differentially private adaptive optimization with delayed preconditioners. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=j1zQGmQQOX1.

Li, X., Tramèr, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

McMahan, H. B. and Pillutla, K. An inversion theorem for buffered linear toeplitz (blt) matrices and applications to streaming differential privacy. *arXiv preprint arXiv:2504.21413*, 2025.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Nocedal, J. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35 (151):773–782, 1980.

Papyan, V. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size, 2019. URL https://arxiv.org/abs/1811.07062.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

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

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *CVPR*, 2018.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3462–3471, 2017.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., and Mironov, I. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yekhanin, S., and Zhang, H. Differentially private fine-tuning of language models. *CoRR*, abs/2110.06500, 2021a. URL https://arxiv.org/abs/2110.06500.

Yu, D., Zhang, H., Chen, W., and Liu, T. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=7aogOj_VYO0.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yekhanin, S., and Zhang, H. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

Zhou, Y., Wu, S., and Banerjee, A. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7dpmlkBuJFC.

A Experiment Details and Results

We use PyTorch (Paszke et al., 2019) as the deep learning framework and datasets and models provided by its `torchvision` package (if exists) or HuggingFace. We implement DP-MF training as a new optimizer under Opacus (Yousefpour et al., 2021), a library that enables training PyTorch models with differential privacy, which is also used for our privacy accounting. Our implementation, especially the online noise vector generation, is highly inspired by Granqvist et al. (2024).

All the experiments in this paper are runnable on a single NVIDIA A5000 24GB GPU.

A.1 Experiments Setup

Datasets **Vision tasks:** We use a variety of datasets, including SVHN (Netzer et al., 2011), CIFAR-10 (Krizhevsky et al., 2009) as two canonical vision tasks. We also evaluate our methods on one medical image dataset, ChestX-ray14 (Wang et al., 2017), as it is closer to privacy-sensitive applications. When using ChestX-ray14, we perform a multi-label classification, where each image may be annotated with up to 14 thoracic disease categories (e.g., pneumonia, effusion, cardiomegaly, mass). Due to its class imbalance, we compute Area Under the ROC Curve (AUC) for each class and use the average of them (mean AUC) as the accuracy metric. Unless otherwise stated, we use 6,000 TinyImageNet (Le & Yang, 2015) data samples as the public dataset to pretrain the models (Figures 1 and 3, and the main results). We also use those data samples to estimate Hessian/eigenvalues of Hessian for our Problem P1 unless specified otherwise. Figure 2 used CIFAR-10 validation set for the Hessian estimation. **Language tasks:** We use MNLI (Wang et al., 2019) as the dataset for language tasks, which is a large-scale benchmark dataset for natural language inference. Each example contains a premise sentence and a hypothesis sentence, labeled as entailment, contradiction, or neutral, i.e., a 3-class classification task. To improve fine-tuning efficiency, we split 10% of its full data samples for training, and we surprisingly found that its noise-free accuracy is close to fine-tuning on 100% data samples as reported in Yu et al. (2021a). For the public dataset, we use WikiText (Merity et al., 2016), which is a collection of large-scale language-modeling datasets built from high-quality Wikipedia articles, designed to reflect natural, long-range text rather than heavily preprocessed sentences.

Network Architectures **Vision tasks:** Our evaluation uses both CNN-based (e.g., ResNet152 (He et al., 2016), VGG (Simonyan & Zisserman, 2015)) and Transformer-based (ViTs (Dosovitskiy et al., 2020)) architectures. For some experiments, we use small-scale CNNs (Table 3), whose full Hessian can be computed on our computers. When using a small CNN, except for Figure 5 where we use both, we use a CNN with GroupNorm (Table 3a). For ResNet152 and ViT (`vit_b_16`), we use the models (together with pretrained weights) provided by `torchvision`. For VGG, we use the network architecture from Choquette-Choo et al. (2023b), which is smaller than a standard VGG. **Language tasks:** For the classification task on MNLI, we use a pretrained RoBERTa model (`ROBERTA_BASE`) provided by `torchtext`.

Table 3: Network architectures of two small-scale CNN models.

(a) CNN with GroupNorm

Layer	Parameters
Conv2d + GroupNorm + ReLU	$3 \to 16$, kernel 3×3, stride 1
Conv2d + GroupNorm + ReLU	$16 \to 32$, kernel 3×3, stride 2
Conv2d + GroupNorm + ReLU	$32 \to 64$, kernel 3×3, stride 2
AdaptiveAvgPool2d	output 2×2
Linear + ReLU	$(256, 32)$
Linear	$(32, 10)$

(b) CNN with Maxpool

Layer	Parameters
Conv2d + ReLU + MaxPool2d	$3 \to 16$, kernel 3×3, stride 1; pool 2×2
Conv2d + ReLU + MaxPool2d	$16 \to 32$, kernel 3×3, stride 2; pool 2×2
Conv2d + ReLU	$32 \to 64$, kernel 3×3, stride 2
AdaptiveAvgPool2d	output 2×2
Linear + ReLU	$(256, 32)$
Linear	$(32, 10)$

Training/Finetuning Strategy We use full parameter training (for small CNNs and VGG), parameter-efficient fine-tuning mechanisms including freezing layer (for finetuning the last layer of ResNet152) and LoRA (Hu et al., 2022) (for ViT). For fine-tuning ViTs with LoRA, we use low rank $r = 8$, making the total number of finetuning parameters around 1M. For fine-tuning RoBERTa with LoRA, we use low rank $r = 16$, making the total number of finetuning parameters around 3M. Based on the recommendations of Gu et al. (2025), we finetune those parameters on public data first, then perform privacy-preserving training/finetuning on the private target datasets.

Hyperparameters Unless otherwise stated, we split 10% of data samples from training sets as validation sets and **tune hyperparameters on validation sets**. All models are trained for 20 epochs on CIFAR-10 with batch size 450, 5 epochs on ChestX-ray with batch size 150, and 3 epochs on MNLI with batch size 50. We use constant learning rate, no momentum, and no weight decay. We use Optuna (Akiba et al., 2019) for auto hyperparameter tuning and search within $[0.1, 1.]$ for the learning rate and $[0.1, 10.]$ for clip norm, for 30 completed trials. Additionally, for DP-BANDMF and NoiseCurve, we search within $[1, 20]$ for the band size.

A.2 What if Hyperparameters are tuned on Test set (a common practice in the DP literature)

We re-evaluate the experiments in Figure 1 by training the model on the **full training set** and **tuning hyperparameters on the test set**. Under this setup, we are able to reproduce results comparable to those reported in Choquette-Choo et al. (2023a), and our method even achieves higher accuracy.

Table 4: Test accuracy comparison on CIFAR-10 under various ϵ ($\delta = 10^{-5}$). We use the **full** trainset for training data and tune hyperparameters on testset. Other setups follow Figure 1.

Non-private	ϵ	DP-SGD	DP-BANDMF	NoiseCurve
84.05	1	53.68	54.20	**56.81**
	2	58.01	58.16	**61.42**
	5	63.39	65.73	**68.77**
	8	65.06	68.19	**72.87**

A.3 Statistical Significance Tests

We performed both statistical significance tests (one-sided t-test with H1: "our method is more accurate than DP-BANDMF", and the permutation test) for each choice of ϵ in Figure 1. For one-sided t-test, its t-statistic and p-value can be computed using `scipy` package, and we calculate the p-value of the permutation test directly.

- $\epsilon = 1$: $t = 5.44, p = 0.0014$ for the one-sided t-test. For the permutation test of comparing NoiseCurve (6 seeds) to DP-BandMF (6 seeds), there are $12!/(6!6!) = 924$ orderings of their results. Thus, under the null hypothesis of no difference between the two methods, the probability that every run of NoiseCurve would have the higher accuracy is 1/924 for a p-value of $p \approx 0.001$.
- $\epsilon = 2$: $t = 9.31, p < 0.001$ for the one-sided t-test. The same analysis as for the $\epsilon = 1$ case yields $p \approx 0.001$ for the permutation test.
- $\epsilon = 5$: $t = 6.71, p < 0.001$ for the one-sided t-test. The same analysis as for the $\epsilon = 1$ case yields $p \approx 0.001$ for the permutation test.
- $\epsilon = 8$: $t = 28.78, p < 0.00001$ for the one-sided t-test. The same analysis as for the $\epsilon = 1$ case yields $p \approx 0.001$ for the permutation test.

A.4 In-distribution Public Data

Table 5: Comparison of test accuracy using TinyImageNet vs. Validation set as public data under various ϵ ($\delta = 10^{-5}$) across 1 run. CIFAR-10 experiments follow the settings of Figure 1 and ChestX-ray14 experiments follow Table 2a.

(a) CIFAR-10

Non-private	ϵ	DP-SGD	DP-BANDMF	Tiny	Val
84.05	5	63.07	63.08	66.3	66.85
	8	64.55	65.05	69.21	70.24

(b) ChestX-ray14

Non-private	ϵ	DP-SGD	DP-BANDMF	Tiny	Val
73.63	5	58.16	59.32	63.2	65.08
	8	59.21	62.55	65.18	67.49

In addition to using TinyImageNet as public data for eigenvalue estimation, we also evaluate accuracy when eigenvalues are estimated from the validation set of the target dataset, which can be regarded as in-distribution data relative to the training set.

We observe that using validation data eigenvalues yields around 1% improvement on CIFAR-10 and a modest gain of around 2% on ChestX-ray14. These results indicate the robustness of our method: even when only out-of-distribution public data such as TinyImageNet are available, the accuracy improvement remains.

A.5 Computing $\hat{H}$ using random initialization without random-label pretraining

We further evaluate the setting without random-label pretraining, where $\hat{\boldsymbol{H}}$ is computed from randomly-labeled data at a randomly chosen weight initialization.

Table 6: A comparison of test accuracy of VGG on CIFAR-10 with vs. without pretrain. Other setups follow Figure 1.

ϵ		DP-SGD	DP-BANDMF	NoiseCurve
5	w/o pretrain	62.81	63.18	64.72
	w/ pretrain	63.07	63.08	66.3
8	w/o pretrain	64.71	65.89	67.89
	w/ pretrain	64.55	65.05	69.21

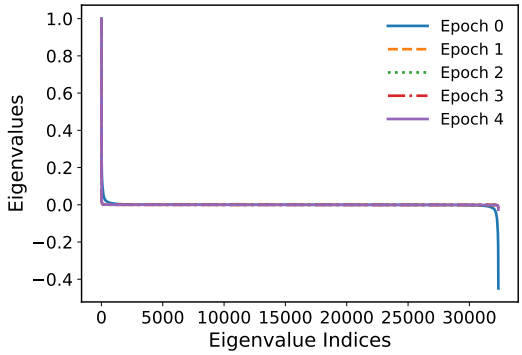


Figure 6: Change of the Hessian eigenvalues during training. Despite significant negative eigenvalues at initialization, positive eigenvalues dominate.

The results show that even without pretraining, our method still achieves a 1–3% accuracy improvement over DP-SGD and DP-BANDMF. We conjecture that this is because modern initialization schemes (e.g., He (He et al., 2015) or Xavier (Glorot & Bengio, 2010)) are designed to preserve the variance of activations and gradients across layers, which helps maintain stable signal propagation at the start of training and mitigates the risk of the parameters being placed in regions with pathological curvature (see Figure 6). Nevertheless, pretraining remains advantageous: it stabilizes the Hessian spectrum, yielding a nearly constant curvature that strengthens the effectiveness of our method and thereby leads to larger accuracy improvements.

B Proofs

Theorem 4.1 (Non-convex Convergence)**.** *Consider the DP-MF iteration updates: $\widetilde{\boldsymbol{w}}_t = \widetilde{\boldsymbol{w}}_{t-1} - \eta\left(\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_{t-1}) + \widetilde{\boldsymbol{z}}_{t-1}\right)$ until step T, where $\boldsymbol{C}$ is an arbitrary lower triangular mixing matrix whose column norms are all 1, $\boldsymbol{Z} \sim \mathcal{N}(0,\sigma^2)^{p\times T}$, and $\widetilde{\boldsymbol{z}}_t$ is the t-th column of $\boldsymbol{Z}\boldsymbol{C}^{-\top}$ (indexed from 0). Let $\mathcal{L}$ be a non-convex loss function with Hessian upper bound $\hat{\boldsymbol{H}}$ (satisfying Properties 1 and 2) such that $\mathcal{L}$ has a minimum value $\mathcal{L}^*$, let the learning rate $\eta\lambda_{\max}(\hat{\boldsymbol{H}}) \le 1$. Then:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 \le \frac{4\mathcal{L}(\boldsymbol{w}_0) - 4\mathcal{L}^*}{\eta T} + 10\eta^2\sigma^2\boxed{\mathrm{Tr}(\boldsymbol{W}\boldsymbol{X}^{-1})} \tag{1}$$

where $\boldsymbol{X} = \boldsymbol{C}^\top \boldsymbol{C}$ and the matrix $\boldsymbol{W}$ is defined as:

$$\boldsymbol{W}[i,j] = \frac{1}{T} \sum_{t=\max(i,j)+1}^{T-1} \mathrm{Tr}\left(\hat{\boldsymbol{H}}^2 (\boldsymbol{I} - \eta \hat{\boldsymbol{H}})^{2t-i-j-2}\right)$$

Proof of Theorem 4.1. We consider the following virtual iterates $\{\boldsymbol{v}_t\}_{t=0}^T$:

$$\begin{aligned}
\boldsymbol{v}_{t+1} &= (\boldsymbol{I} - \eta \hat{\boldsymbol{H}})\boldsymbol{v}_t - \eta(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \hat{\boldsymbol{H}}\widetilde{\boldsymbol{w}}_t) \\
&= \boldsymbol{v}_t - \eta\left(\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) + \hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\right)
\end{aligned}$$

Let $\boldsymbol{v}_0 = \widetilde{\boldsymbol{w}}_0 = \boldsymbol{w}_0$. For notation simplicity, let

$$\boldsymbol{b}_t := \nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) + \hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)$$

By the Hessian upper bound $\hat{\boldsymbol{H}}$ assumption on $\mathcal{L}$, we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v}_{t+1}) &\le \mathcal{L}(\boldsymbol{v}_t) - \eta\langle \nabla \mathcal{L}(\boldsymbol{v}_t), \boldsymbol{b}_t\rangle + \frac{\eta^2}{2}\|\boldsymbol{b}_t\|_{\hat{\boldsymbol{H}}}^2 \\
&= \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 - \frac{\eta}{2}\|\boldsymbol{b}_t\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 + \frac{\eta^2}{2}\|\boldsymbol{b}_t\|_{\hat{\boldsymbol{H}}}^2 \\
&= \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 + \left(-\frac{\eta}{2}\|\boldsymbol{b}_t\|_2^2 + \frac{\eta^2}{2}\|\boldsymbol{b}_t\|_{\hat{\boldsymbol{H}}}^2\right)
\end{aligned}$$

Since $\eta \le 1/\lambda_{\max}(\hat{\boldsymbol{H}})$, the last term is ≤ 0 and can be dropped. So,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v}_{t+1}) &\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 + \left(-\frac{\eta}{2}\|\boldsymbol{b}_t\|_2^2 + \frac{\eta^2}{2}\|\boldsymbol{b}_t\|_{\hat{\boldsymbol{H}}}^2\right) \\
&\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2
\end{aligned}$$

Now we work on creating a bound on the second term $-\frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2$:

$$\begin{aligned}
\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 &= \|\nabla \mathcal{L}(\boldsymbol{v}_t) + (\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t))\|_2^2 \\
&\le 2\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + 2\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 \\
\Rightarrow \|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 &\ge \frac{1}{2}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 - \|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 \\
\Rightarrow -\frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 &\le -\frac{\eta}{4}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2
\end{aligned}$$

Substitute back, we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v}_{t+1}) &\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 \\
&\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2
\end{aligned}$$

Using the consequence on $\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2$ of the $\hat{\boldsymbol{H}}$ upper bound, we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{v}_{t+1}) &\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \nabla \mathcal{L}(\boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 \\
&\le \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{\eta}{2}\|\hat{\boldsymbol{H}}(\widetilde{\boldsymbol{w}}_t - \boldsymbol{v}_t)\|_2^2 + \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2
\end{aligned}$$

Now we approach the last term $\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2$, we have:

$$\begin{aligned}
\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t &= \nabla \mathcal{L}(\boldsymbol{v}_t) - (\nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) + \hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)) \\
&= \nabla \mathcal{L}(\boldsymbol{v}_t) - \nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t) \\
\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 &\le 2\|\nabla \mathcal{L}(\boldsymbol{v}_t) - \nabla \mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + 2\|\hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\|_2^2 \\
&\le 4\|\hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\|^2 \quad \text{(consequence of } \hat{\boldsymbol{H}} \text{ upper bound)}
\end{aligned}$$

Substitute back, we have:

$$\begin{aligned}\mathcal{L}(\boldsymbol{v}_{t+1}) &\leq \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{\eta}{2}\|\hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\|^2 + \frac{\eta}{2}\|\nabla\mathcal{L}(\boldsymbol{v}_t) - \boldsymbol{b}_t\|_2^2 \\ &\leq \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{5\eta}{2}\|\hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\|^2\end{aligned}$$

Next, we see

$$\begin{aligned}\widetilde{\boldsymbol{w}}_{t+1} - \widetilde{\boldsymbol{w}}_t &= -\eta\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t) - \eta\widetilde{\boldsymbol{z}}_t \\ \boldsymbol{v}_{t+1} - \boldsymbol{v}_t &= -\eta\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t) + \eta\hat{\boldsymbol{H}}(\widetilde{\boldsymbol{w}}_t - \boldsymbol{v}_t) \\ \widetilde{\boldsymbol{w}}_{t+1} - \widetilde{\boldsymbol{w}}_t - (\boldsymbol{v}_{t+1} - \boldsymbol{v}_t) &= -\eta\hat{\boldsymbol{H}}(\widetilde{\boldsymbol{w}}_t - \boldsymbol{v}_t) - \eta\widetilde{\boldsymbol{z}}_t \\ \widetilde{\boldsymbol{w}}_{t+1} - \boldsymbol{v}_{t+1} &= (\boldsymbol{I} - \eta\hat{\boldsymbol{H}})(\widetilde{\boldsymbol{w}}_t - \boldsymbol{v}_t) - \eta\widetilde{\boldsymbol{z}}_t \\ &= -\eta\sum_{k=0}^{t}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k} \\ \widetilde{\boldsymbol{w}}_t - \boldsymbol{v}_t &= -\eta\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\end{aligned}$$

Plug into the above:

$$\begin{aligned}\mathcal{L}(\boldsymbol{v}_{t+1}) &\leq \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{5\eta}{2}\|\hat{\boldsymbol{H}}(\boldsymbol{v}_t - \widetilde{\boldsymbol{w}}_t)\|^2 \\ &= \mathcal{L}(\boldsymbol{v}_t) - \frac{\eta}{4}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 + \frac{5\eta^3}{2}\|\hat{\boldsymbol{H}}\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\|^2\end{aligned}$$

Rearranging to move the gradient term to the left, then summing $t = 0, \ldots, T-1$ gets us:

$$\begin{aligned}\frac{\eta}{4}\sum_{t=0}^{T-1}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 &\leq \mathcal{L}(\boldsymbol{v}_0) - \mathcal{L}(\boldsymbol{v}_T) + \frac{5\eta^3}{2}\sum_{t=0}^{T-1}\|\hat{\boldsymbol{H}}\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\|^2 \\ \frac{1}{T}\sum_{t=0}^{T-1}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 &\leq \frac{4\mathcal{L}(\boldsymbol{v}_0) - 4\mathcal{L}(\boldsymbol{v}_T)}{\eta T} + \frac{10\eta^2}{T}\sum_{t=0}^{T-1}\|\hat{\boldsymbol{H}}\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\|^2\end{aligned}$$

Let $\boldsymbol{X} = \boldsymbol{C}^\top\boldsymbol{C}$. Now taking expectation w.r.t $\boldsymbol{Z}$ gets us:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2 &\leq \mathbb{E}\frac{4\mathcal{L}(\boldsymbol{v}_0) - 4\mathcal{L}(\boldsymbol{v}_T)}{\eta T} + \mathbb{E}\frac{10\eta^2}{T}\sum_{t=0}^{T-1}\|\hat{\boldsymbol{H}}\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\|^2 \\ &\leq \frac{4\mathcal{L}(\boldsymbol{w}_0) - 4\mathcal{L}^*}{\eta T} + \frac{10\eta^2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\hat{\boldsymbol{H}}\sum_{k=0}^{t-1}(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^k\widetilde{\boldsymbol{z}}_{t-k-1}\|^2 \\ &= \frac{4\mathcal{L}(\boldsymbol{w}_0) - 4\mathcal{L}^*}{\eta T} + \frac{10\eta^2}{T}\sum_{t=0}^{T-1}\sum_{i,j=0}^{t-1}\mathbb{E}\left[\widetilde{\boldsymbol{z}}_i^\top(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^{t-i-1}\hat{\boldsymbol{H}}^2(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^{t-j-1}\widetilde{\boldsymbol{z}}_j\right] \\ &= \frac{4\mathcal{L}(\boldsymbol{w}_0) - 4\mathcal{L}^*}{\eta T} + \frac{10\eta^2\sigma^2}{T}\sum_{t=0}^{T-1}\sum_{i,j=0}^{t-1}[\boldsymbol{X}^{-1}]_{ij}\operatorname{Tr}\left(\hat{\boldsymbol{H}}^2(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^{2t-i-j-2}\right) \\ &= \frac{4\mathcal{L}(\boldsymbol{w}_0) - 4\mathcal{L}^*}{\eta T} + \frac{10\eta^2\sigma^2}{T}\sum_{i,j=0}^{T-1}[\boldsymbol{X}^{-1}]_{ij}\sum_{t=\max(i,j)+1}^{T-1}\operatorname{Tr}\left(\hat{\boldsymbol{H}}^2(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^{2t-i-j-2}\right)\end{aligned}$$

Now we define $\boldsymbol{W} \in \mathbb{R}^{T\times T}$ entrywise as

$$\boldsymbol{W}_{i,j} := \frac{1}{T}\sum_{t=\max(i,j)+1}^{T-1}\operatorname{Tr}\left(\hat{\boldsymbol{H}}^2(\boldsymbol{I} - \eta\hat{\boldsymbol{H}})^{2t-i-j-2}\right)$$

Then we have

$$\sum_{i,j=0}^{T-1}[\boldsymbol{X}^{-1}]_{ij}\sum_{t=\max(i,j)+1}^{T-1}\operatorname{Tr}\left(\hat{\boldsymbol{H}}^2(\boldsymbol{I}-\eta\hat{\boldsymbol{H}})^{2t-i-j-2}\right)=T*\sum_{i,j=0}^{T-1}\boldsymbol{W}_{i,j}[\boldsymbol{X}^{-1}]_{i,j}$$
$$=T*\operatorname{Tr}(\boldsymbol{W}\boldsymbol{X}^{-1})$$

Therefore, finally we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\widetilde{\boldsymbol{w}}_t)\|_2^2\leq\frac{4\mathcal{L}(\boldsymbol{w}_0)-4\mathcal{L}^*}{\eta T}+10\eta^2\sigma^2\operatorname{Tr}(\boldsymbol{W}\boldsymbol{X}^{-1})$$

That completes the proof. □

Theorem 5.1 (Dominance Certificate)**.** *Let $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ be two symmetric real matrices, with $\hat{\boldsymbol{H}}$ positive definite. Define (using the convention 0/0=0) two dominance certificates $\rho(\boldsymbol{H},\hat{\boldsymbol{H}})$ as*

$$\rho_1(\boldsymbol{H},\hat{\boldsymbol{H}}):=\max_{\boldsymbol{v}\neq\boldsymbol{0}}\frac{\|\boldsymbol{H}\boldsymbol{v}\|_2}{\|\hat{\boldsymbol{H}}\boldsymbol{v}\|_2}\quad\textit{and}\quad\rho_2(\boldsymbol{H},\hat{\boldsymbol{H}}):=\max_{\boldsymbol{v}\neq\boldsymbol{0}}\frac{\boldsymbol{v}^\top\boldsymbol{H}\boldsymbol{v}}{\boldsymbol{v}^\top\hat{\boldsymbol{H}}\boldsymbol{v}}$$

Then, $\hat{\boldsymbol{H}}$ upper bounds $\boldsymbol{H}\iff\rho(\boldsymbol{H},\hat{\boldsymbol{H}}):=\max(\rho_1,\rho_2)\leq 1$.

Proof of Theorem 5.1. Because $\boldsymbol{H}$ and $\hat{\boldsymbol{H}}$ are symmetric,

$$\|\boldsymbol{H}\boldsymbol{v}\|_2^2=\boldsymbol{v}^\top\boldsymbol{H}^2\boldsymbol{v}\qquad\text{and}\qquad\|\hat{\boldsymbol{H}}\boldsymbol{v}\|_2^2=\boldsymbol{v}^\top\hat{\boldsymbol{H}}^2\boldsymbol{v}$$

Therefore,

$$\rho_1(\boldsymbol{H},\hat{\boldsymbol{H}})\leq 1\iff\|\boldsymbol{H}\boldsymbol{v}\|_2\leq\|\hat{\boldsymbol{H}}\boldsymbol{v}\|_2,\forall\boldsymbol{v}\iff\boldsymbol{v}^\top\boldsymbol{H}^2\boldsymbol{v}\leq\boldsymbol{v}^\top\hat{\boldsymbol{H}}^2\boldsymbol{v},\forall\boldsymbol{v}$$

and,

$$\boldsymbol{v}^\top\boldsymbol{H}^2\boldsymbol{v}\leq\boldsymbol{v}^\top\hat{\boldsymbol{H}}^2\boldsymbol{v}\quad\text{for every }\boldsymbol{v}\iff\boldsymbol{H}^2\preceq\hat{\boldsymbol{H}}^2.$$

Similarly,

$$\rho_2(\boldsymbol{H},\hat{\boldsymbol{H}})\leq 1\iff\boldsymbol{v}^\top\boldsymbol{H}\boldsymbol{v}\leq\boldsymbol{v}^\top\hat{\boldsymbol{H}}\boldsymbol{v},\forall\boldsymbol{v}\iff\boldsymbol{v}^\top(\hat{\boldsymbol{H}}-\boldsymbol{H})\boldsymbol{v}\geq 0,\forall\boldsymbol{v}$$

and

$$\boldsymbol{v}^\top(\hat{\boldsymbol{H}}-\boldsymbol{H})\boldsymbol{v}\geq 0,\forall\boldsymbol{v}\iff\boldsymbol{H}\preceq\hat{\boldsymbol{H}}$$

Combining the two equivalences gives

$$\max\{\rho_1(\boldsymbol{H},\hat{\boldsymbol{H}}),\rho_2(\boldsymbol{H},\hat{\boldsymbol{H}})\}\leq 1\iff\boldsymbol{H}^2\preceq\hat{\boldsymbol{H}}^2\text{ and }\boldsymbol{H}\preceq\hat{\boldsymbol{H}}$$

which proves the result. □

C Algorithm Pseudocode

We present the full pseudocode of the DP-MF training algorithm, adapted from Choquette-Choo et al. (2023a). To ensure completeness, we additionally present Algorithm 3 to show how to do privacy accounting under DP-MF, together with how DP-BandMF forms training data batches (Algorithm 4), and generates correlated noise vectors online (Algorithm 5). To simplify the algorithm presentation and remain consistent with Choquette-Choo et al. (2023a), we assume that all relevant quantities are evenly divisible.

Our method, NoiseCurve, modifies it by providing a different the noise correlation $\boldsymbol{C}$ as shown in Algorithm 1.

Algorithm 3 DP-BANDMF Privacy Accounting

Input: Privacy budget (ϵ, δ), Privacy accountant `PRVAccountant`, Clip norm ζ, Batch size B, Total number of data samples n, Training number of epochs E, Number of bands b
Output: Noise scale σ

1: Total number of training steps: $T \leftarrow \frac{En}{B}$
2: Sample rate for DP-BANDMF: $q \leftarrow \frac{Bb}{n}$
3: Total number of compositions for DP-BANDMF: $T \leftarrow \frac{T}{b}$
4: Compute noise multiplier: $C \leftarrow \texttt{PRVAccountant}(\epsilon, \delta, q, T)$
5: Noise scale: $\sigma \leftarrow C \cdot \zeta$

Algorithm 4 DP-BANDMF Data Batch Sampling

Input: Dataset $\mathcal{D}$, Batch size $|B|$, Current step t
Output: Data batch B_t

1: $D_1, \ldots, D_b \leftarrow$ arbitrary partition of $\mathcal{D}$
2: $j = t (\text{mod } b) + 1$
3: $B_t \leftarrow$ random size $|B|$ subset of D_j

Algorithm 5 DP-BANDMF Noise Generation Online

Input: Noise correlation $\boldsymbol{C} \in \mathbb{R}^{T \times T}$ (b-banded and lower triangular), Noise scale σ, Current step t
Output: $(\boldsymbol{C}^{-1}\boldsymbol{Z})_{[t,:]}$

1: **for** $i = 1, 2, \ldots, T$ **do**
2: $\quad \boldsymbol{z}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_p)$
3: $\quad \boldsymbol{x}_i = (\boldsymbol{z}_i - \sum_{j=i-b+1}^{i-1} \boldsymbol{C}_{i,j}\boldsymbol{x}_j)/\boldsymbol{C}_{i,i}$
4: **end for**
5: **Return** $\boldsymbol{x}$

Note that in Algorithm 5, Choquette-Choo et al. (2023a) simplify the presentation by using the convention that out-of-bounds indexing into a matrix or vector returns 0.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction provide a clear overview of the paper and accurately represent its main contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 7 specifically discusses the limitations of the work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Each theoretical result clearly states its assumptions within the theorem, and the corresponding complete and correct proof is presented in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide all necessary experimental setups in the captions and surrounding text of tables and figures to clarify the results. Detailed experimental procedures are included in the Appendix to enable full reproduction of our findings.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We have uploaded the code and provided detailed instructions in the supplemental material to enable faithful reproduction of the main experimental results. All datasets used in this paper are publicly accessible.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: All the necessary details are provided in Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We evaluated the statistical significance of our experiments using both permutation tests and one-sided t-tests, as detailed in the main text and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments described in this paper can be reproduced on a single NVIDIA A5000 (24GB) GPU within a reasonable execution time.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read and understood the NeurIPS Code of Ethics, and confirm that the research described in this paper fully complies with it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We consider improving the accuracy of privacy-preserving neural networks to be a positive broader impact. This theme is the basis of our paper, and the goal of improving the accuracy of privacy-preserving training is discussed in the introduction.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: All data and models used in this paper are well-established benchmarks and foundation models that have been widely adopted by the research community for many years. Therefore, this paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We confirm that all creators or original owners of the assets used (e.g., code, data, models) are properly credited, and their licenses and terms of use are fully respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provided detailed documents in the supplemental material to enable faithful reproduction of the main experimental results.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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