
GeoClip: Geometry-Aware Clipping for Differentially Private SGD

Atefeh Gilani
Arizona State University
Tempe, AZ, USA
agilani2@asu.edu

Naima Tasnim
Arizona State University
Tempe, AZ, USA
ntasnim2@asu.edu

Lalitha Sankar
Arizona State University
Tempe, AZ, USA
lsankar@asu.edu

Oliver Kosut
Arizona State University
Tempe, AZ, USA
okosut@asu.edu

Abstract

Differentially private stochastic gradient descent (DP-SGD) is the most widely used method for training machine learning models with provable privacy guarantees. A key challenge in DP-SGD is setting the per-sample gradient clipping threshold, which significantly affects the trade-off between privacy and utility. While recent adaptive methods improve performance by adjusting this threshold during training, they operate in the standard coordinate system and fail to account for correlations across the coordinates of the gradient. We propose GeoClip, a geometry-aware framework that clips and perturbs gradients in a transformed basis aligned with the geometry of the gradient distribution. GeoClip adaptively estimates this transformation using only previously released noisy gradients, incurring no additional privacy cost. We provide convergence guarantees for GeoClip and derive a closed-form solution for the optimal transformation that minimizes the amount of noise added while keeping the probability of gradient clipping under control. Experiments on both tabular and image datasets demonstrate that GeoClip consistently outperforms existing adaptive clipping methods under the same privacy budget.

1 Introduction

As machine learning models are increasingly trained on sensitive user data, ensuring strong privacy guarantees during training is essential to reduce the risk of misuse, discrimination, or unintended data exposure. Differential privacy (DP) [1, 2] offers a principled framework for protecting individual data, and has become a cornerstone of privacy-preserving machine learning. In deep learning, the most widely used approach for DP is differentially private stochastic gradient descent (DP-SGD) [3], which clips per-sample gradients and adds calibrated noise to their average.

Despite its widespread use, standard DP-SGD has a key limitation: it relies on a fixed clipping threshold to bound the sensitivity of individual gradients. Selecting this threshold poses a challenging privacy-utility tradeoff—setting it too low discards useful gradient information, while setting it too high increases sensitivity and necessitates injecting more noise, ultimately degrading model performance. This trade-off was observed empirically by McMahan et al. [4] and later analyzed theoretically and shown to be a fundamental limitation of differentially private learning by Amin et al. [5]. Moreover, the optimal threshold can vary over the course of training, across tasks, and between datasets, limiting the effectiveness of a fixed setting.

To address this, recent work has proposed adaptive strategies that dynamically adjust the clipping threshold during training. One class of methods uses decay schedules to reduce the threshold over time. Yu et al. [6] and Du et al. [7] propose linear and near-linear decay rules, respectively, where the schedule is predefined and does not depend on the dataset—hence, no privacy budget is required. Lin et al. [8] introduce a nonlinear decay schedule, along with a transfer strategy that leverages public data to guide threshold selection. Recently, methods have been introduced which set the clipping level based on the data during the training process. These include AdaClip [9], which applies coordinate-wise clipping based on estimated gradient variances. Yet another is quantile-based clipping [10], which sets the threshold using differentially private quantiles of per-sample gradient norms. Although [10] was designed for federated learning, it can be adapted to centralized DP-SGD. These adaptive strategies have been shown to improve model utility while preserving privacy guarantees.

Despite these advances, existing adaptive clipping methods remain agnostic to the geometry of the gradient distribution. They operate in the standard basis—treating each coordinate independently. This overlooks dependencies between coordinates, especially when gradients exhibit strong correlations across dimensions. In such cases, independently clipping and perturbing each coordinate can introduce redundant noise without improving privacy, ultimately degrading model utility. To address this, we propose *GeoClip*, a method that transforms gradients into a decorrelated basis that better reflects their underlying geometry. By applying DP mechanisms in this transformed space, GeoClip allocates noise more effectively, achieving a better privacy-utility tradeoff.

A different approach to correlations in gradients from the literature considers introducing correlations—dependencies across iterations and between entries of the noise vector—into the noise, rather than injecting i.i.d. Gaussian noise. This has been shown to improve the utility of private training [11, 12]. Choquette-Choo et al. [13] strengthen this direction by analytically characterizing near-optimal spatio-temporal correlation structures that lead to provably tighter privacy-utility tradeoffs. However, these approaches have been developed independently of adaptive clipping, and the effect of combining both methods remains unexplored. In addition, [13] uses pre-determined correlations in the noise, rather than being tailored to the data, as our approach is.

GeoClip is data-driven but does not require any additional privacy budget to compute the clipping transformation. Instead, it reuses the noisy gradients already released during training to estimate the mean and the covariance of the gradient distribution. By reusing these privatized gradients, GeoClip adapts its basis over time without accessing raw data or incurring additional privacy cost.

We list our main contributions below:

1. We propose GeoClip, a novel framework that applies differential privacy in a transformed basis rather than the standard coordinate system. To guide the choice of transformation, we derive a convergence theorem (Theorem 1) showing how the basis impacts convergence under DP-SGD, providing theoretical guidance for selecting transformations that improve utility.
2. Building on this insight, we formulate a convex optimization problem to find the transformation, and derive a closed-form solution (Theorem 2).
3. We introduce two algorithms to estimate the transformation using only previously released noisy gradients. The first is based on a moving average to estimate the gradient covariance matrix.
4. For large-scale models, the full gradient covariance matrix is prohibitively large to store. Thus, our second algorithm uses a streaming low-rank approximation of the covariance matrix. This second algorithm is thus suitable for deep models with large parameter counts.
5. We validate GeoClip through experiments on synthetic, tabular, and image datasets, showing that it consistently outperforms existing adaptive clipping methods under the same privacy budget.

Notation. We denote the $d \times d$ identity matrix by I_d . The notation $N \sim \mathcal{N}(0, \sigma^2 I_d)$ denotes a d -dimensional Gaussian with zero mean and covariance $\sigma^2 I_d$. We use $\|x\|_2$ for the Euclidean norm of a vector x , A^{-1} for the inverse of matrix A , and $A^\top$ for its transpose. The trace of a matrix A is denoted by $\text{Tr}(A)$, and $\text{Cov}(x \mid y)$ refers to the conditional covariance of x given y .

2 General Framework

Let $\mathcal{D} = \{x_k\}_{k=1}^N$ be a dataset of N examples, and let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ denote the empirical loss function defined as the average of per-sample losses:

$$f(\theta) = \frac{1}{N} \sum_{k=1}^N f_k(\theta), \quad (1)$$

where $\theta \in \mathbb{R}^d$ is the model parameter vector and each f_k corresponds to the loss on the k -th data point x_k . In DP-SGD, the algorithm updates θ using a noisy clipped stochastic gradient to ensure privacy. Let $g_t \in \mathbb{R}^d$ denote the stochastic gradient at iteration t . The update rule is:

$$\theta_{t+1} = \theta_t - \eta \tilde{g}_t, \quad (2)$$

where $\tilde{g}_t$ is the privatized version of g_t , obtained by clipping and adding noise.

GeoClip builds on the DP-SGD framework but changes how gradients are processed before clipping and adding noise. It begins by shifting and projecting the gradient into a new coordinate system:

$$\omega_t = M_t(g_t - a_t), \quad (3)$$

where $a_t \in \mathbb{R}^d$ is a reference point and $M_t \in \mathbb{R}^{d \times d}$ is a full-rank transformation matrix that defines the new basis. To enforce differential privacy, we clip the transformed gradient ω_t to unit norm and add Gaussian noise:

$$\tilde{\omega}_t = \frac{\omega_t}{\max(1, \|\omega_t\|_2)} + N_t, \quad N_t \sim \mathcal{N}(0, \sigma^2 I_d), \quad (4)$$

where σ is set based on the desired privacy guarantee. We then map the noisy, clipped gradient back to the original space:

$$\tilde{g}_t = M_t^{-1} \tilde{\omega}_t + a_t. \quad (5)$$

Remark 1. *GeoClip generalizes AdaClip of Pichapati et al. [9], which itself extends standard DP-SGD. AdaClip essentially assumes that M_t is diagonal for per-coordinate scaling, whereas GeoClip allows M_t to be any full-rank matrix. This added flexibility allows GeoClip to account for correlations between gradient components and inject noise along more meaningful directions.*

A key feature of our framework is that the privacy guarantees remain unaffected by the choice of M_t and a_t , as long as no privacy budget is used to compute them. This allows M_t and a_t to be chosen entirely based on utility, without compromising privacy. The main challenge, then, is how to select these parameters effectively. To address this, we first define a performance metric for GeoClip. Inspired by Pichapati et al. [9], we derive the following convergence bound for our framework and use it to guide the design of M_t and a_t , ultimately improving both convergence and training efficiency.

Theorem 1 (Convergence of GeoClip). *Assume f has an L -Lipschitz continuous gradient. Further, assume the stochastic gradients are bounded, i.e., $\|\nabla f_k(\theta)\| \leq G$, and have bounded variance, i.e., $\mathbb{E}_k \|\nabla f_k(\theta) - \nabla f(\theta)\|^2 \leq \sigma_g^2$. Let $\theta^* = \arg \min_{\theta \in \mathbb{R}^d} f(\theta)$ denote the optimal solution, and suppose the learning rate satisfies $\eta < \frac{2}{3L}$. Then, for the iterates $\{\theta_t\}_{t=0}^{T-1}$ produced by GeoClip with batch size 1 using the update rule $\theta_{t+1} = \theta_t - \eta \tilde{g}_t$, where $\tilde{g}_t$ is defined in (5), the average squared gradient norm satisfies:*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\theta_t)\|^2 &\leq \underbrace{\frac{f(\theta_0) - f(\theta^*)}{T \left(\eta - \frac{3L\eta^2}{2} \right)}}_{\text{Optimization gap}} + \underbrace{\frac{3L\eta}{2 - 3L\eta} \sigma_g^2}_{\text{Gradient variance term}} + \underbrace{\frac{L\eta\sigma^2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} \text{Tr} \left[(M_t^\top M_t)^{-1} \right]}_{\text{Noise-injection term}} \\ &\quad + \underbrace{\frac{2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} \left[\beta(a_t) \left(\text{Tr} (M_t^\top M_t \Sigma_t) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \right) \right]}_{\text{Clipping error term}}, \end{aligned} \quad (6)$$

where $\theta^t = (\theta_0, \dots, \theta_t)$ represents the history of parameter values up to iteration t , $\Sigma_t = \text{Cov}(g_t | \theta^t)$, and

$$\beta(a_t) = (G + \|a_t\|) \left(G + \frac{3L\eta}{2}(G + \|a_t\|) \right). \quad (7)$$

The above result generalizes Theorem 2 in Pichapati et al. [9]; proof details are in Appendix A.

Theorem 1 provides insight into how we should choose the transformation parameters a_t and M_t . In particular, we want to choose the transformation parameters a_t and M_t , for all t , to minimize the right-hand side of (6). The reference point a_t directly affects the clipping error by setting the center around which gradients are clipped, thereby influencing how much of each gradient is truncated. In contrast, the noise injection term remains independent of a_t , as noise is added regardless of the gradient's position relative to a_t . From Theorem 1, the clipping error at iteration t takes the form

$$\beta(a_t) \left(\text{Tr}(M_t^\top M_t \Sigma_t) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \right), \quad (8)$$

where $\beta(a_t)$ is a scale factor that grows with $\|a_t\|$. A natural choice is to set $a_t = \mathbb{E}[g_t | \theta^t]$ which eliminates the bias term $\|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2$. Given this choice, since gradients are norm-bounded by G , we can apply Jensen's inequality to also bound the norm of their mean, to obtain the bound

$$\beta(a_t) \leq 2G^2(1 + 3L\eta). \quad (9)$$

We now show that the remaining contribution to the clipping error term $\text{Tr}(M_t^\top M_t \Sigma_t)$ in fact serves as an upper bound on the probability that the gradient is clipped (i.e., $\|\omega\| > 1$). We do so using Markov's inequality as detailed below:

$$\text{Tr}(M_t^\top M_t \Sigma_t) = \mathbb{E}[\|M_t(g_t - \mathbb{E}[g_t | \theta^t])\|^2 | \theta^t] \quad (10)$$

$$= \mathbb{E}[\|\omega\|^2 | \theta^t] \quad (11)$$

$$\geq \Pr(\|\omega\| > 1 | \theta^t). \quad (12)$$

We can now interpret the clipping error term as the likelihood that clipping occurs—ideally, the lower the better as the gradient information will be better preserved. However, in setting a clipping level, we must also be aware of the amount of noise: if M_t is scaled down, there is effectively more noise, as captured by the noise-injection term in (6). We handle this tradeoff via the following optimization problem for the transformation matrix M_t :

$$\begin{aligned} & \underset{M_t}{\text{minimize}} && \text{Tr}(M_t^\top M_t)^{-1} \\ & \text{subject to} && \text{Tr}(M_t^\top M_t \Sigma_t) \leq \gamma. \end{aligned} \quad (13)$$

Theorem 2. Let $\Sigma_t = \text{Cov}(g_t | \theta^t)$ be a positive definite matrix. The optimal transformation matrix $M_t^* \in \mathbb{R}^{d \times d}$ at iteration t for the optimization problem in (13), along with its corresponding objective value, are given by:

$$M_t^* = \left(\frac{\gamma}{\sum_{i=1}^d \sqrt{\lambda_i}} \right)^{1/2} \Lambda_t^{-1/4} U_t^\top, \quad \text{Tr}(M_t^{*\top} M_t^*)^{-1} = \frac{\left(\sum_{i=1}^d \sqrt{\lambda_i} \right)^2}{\gamma}, \quad (14)$$

where $\Sigma_t = U_t \Lambda_t U_t^\top$ is the eigendecomposition of the covariance matrix, with $\Lambda_t = \text{diag}(\lambda_1, \dots, \lambda_d)$ containing its eigenvalues.

Remark 2. Applying the optimal transformation M_t^* to the gradient g_t at iteration t , the covariance of the transformed gradient $\tilde{\omega}_t = M_t^*(g_t - \mathbb{E}[g_t | \theta^t])$, conditioned on the history θ^t , is

$$\text{Cov}(\tilde{\omega}_t | \theta^t) = M_t^* \Sigma_t M_t^{*\top} = \frac{\gamma \Lambda_t^{1/2}}{\sum_{i=1}^d \sqrt{\lambda_i}} = \gamma \text{diag} \left(\frac{\sqrt{\lambda_1}}{\sum_{i=1}^d \sqrt{\lambda_i}}, \dots, \frac{\sqrt{\lambda_d}}{\sum_{i=1}^d \sqrt{\lambda_i}} \right).$$

Thus, the optimal transformation decorrelates and scales down the gradients while preserving the relative ordering of variance across directions, in contrast to traditional whitening, which eliminates all variance structure. Note that setting $a_t = \mathbb{E}[g_t | \theta^t]$ and choosing $M_t = \sqrt{\frac{\gamma}{d}} \Lambda_t^{-1/2} U_t^\top$

would amount to a whitening of the gradients and ensures that the constraint in (13) is active, i.e., $\text{Tr}(M_t^\top M_t \Sigma_t) = \gamma$. Under this choice, the objective becomes:

$$\text{Tr}(M_t^\top M_t)^{-1} = \frac{d}{\gamma} \text{Tr}(\Lambda_t) = \frac{d}{\gamma} \sum_{i=1}^d \lambda_i. \quad (15)$$

Comparing our objective in (14) with the objective resulting from the whitening transformation, we observe that applying the Cauchy–Schwarz inequality yields:

$$\frac{d}{\gamma} \sum_{i=1}^d \lambda_i \geq \frac{1}{\gamma} \left(\sum_{i=1}^d \sqrt{\lambda_i} \right)^2, \quad (16)$$

with equality if and only if $\lambda_1 = \dots = \lambda_d$. This shows that our solution achieves a strictly smaller objective than whitening in all non-isotropic cases, where the gradient distribution exhibits unequal variance along the eigenbasis directions.

The proof of Theorem 2 is provided in Appendix B.

3 Algorithm Overview

Algorithm 1 outlines our proposed GeoClip method. We explain its key steps below.

Moving average for mean and covariance estimation. While the theoretical results from Section 2 assume access to the true gradient distribution, in practice this distribution is unknown, and so it must be estimated using only privatized gradients. We estimate the mean and covariance using exponential moving averages computed from those privatized gradients. This enables us to estimate the geometry of the gradients without consuming additional privacy budget. Specifically, we maintain estimates a_t of the mean, and Σ_t of the covariance matrix, which are updated according to:

$$a_{t+1} \leftarrow \beta_1 a_t + (1 - \beta_1) \tilde{g}_t \quad (17)$$

$$\Sigma_{t+1} \leftarrow \beta_2 \Sigma_t + (1 - \beta_2) (\tilde{g}_t - a_t)(\tilde{g}_t - a_t)^\top \quad (18)$$

where β_1 and β_2 are constants close to 1 (e.g., $\beta_1 = 0.99$, $\beta_2 = 0.999$). The eigenvalues and eigenvectors used for the transformation are then computed from the estimated covariance.

Clamping eigenvalues. The covariance matrix is positive semi-definite and may contain zero eigenvalues, which can cause numerical instability. To address this, we clamp eigenvalues from below at a small threshold h_1 (e.g., 10^{-15}). Since we only observe privatized gradients, which may be noisy and unstable, we also clamp from above at h_2 to prevent extreme scaling.

Covariance update with mini-batch. Let B denote a mini-batch of training examples sampled at each iteration, with $|B|$ indicating the batch size. When using mini-batch gradient descent with $|B| > 1$, we must estimate the mean and covariance from the privatized batch averages of the per-sample gradients. Let g_i be the random variable representing the gradient of the i -th sample in the batch. Let $\bar{g} = \frac{1}{|B|} \sum_{i \in B} g_i$ be the batch average gradient. Assuming that the g_i are i.i.d., with the same distribution as g , the covariance of the average gradient $\bar{g}$ satisfies

$$\text{Cov}(\bar{g}) = \text{Cov}\left(\frac{1}{|B|} \sum_{i \in B} g_i\right) = \frac{1}{|B|} \text{Cov}(g). \quad (19)$$

The same principle applies when we observe only the privatized average gradient $\tilde{g}$. To account for this averaging effect, the covariance update is scaled by the batch size; i.e., line 11 of GeoClip in Algorithm 1 becomes

$$\Sigma_{t+1} \leftarrow \beta_2 \Sigma_t + |B|(1 - \beta_2) (\tilde{g}_t - a_t)(\tilde{g}_t - a_t)^\top. \quad (20)$$

Low-Rank PCA. When the dimensionality is high, computing and storing the full gradient covariance matrix becomes impractical. To address this, we propose a method to maintain a low-rank approximation using a simple and efficient procedure we refer to as *Streaming Rank- k PCA* (Algorithm 2). Specifically, we maintain an approximate eigendecomposition of the covariance in the form $U_t \Lambda_t U_t^\top$,

Algorithm 1 GeoClip

Require: Dataset $\mathcal{D}$, model f_θ , loss $\mathcal{L}$, learning rate η , noise scale σ , steps T , hyperparameters

- $h_1, h_2, \beta_1, \beta_2$
- 1: Initialize θ , mean vector $a_0 = 0$, covariance $\Sigma_0 = I_d$, transform $M_0 = M_0^{\text{inv}} = I_d$
 - 2: **for** $t = 0$ to T **do**
 - 3: Sample a data point (x_t, y_t)
 - 4: Compute gradient $g_t \leftarrow \nabla_\theta \mathcal{L}(f_\theta(x_t), y_t)$
 - 5: Center and transform: $\omega_t \leftarrow M_t(g_t - a_t)$
 - 6: Clip: $\bar{\omega}_t \leftarrow \omega_t / \max(1, \|\omega_t\|_2)$
 - 7: Add noise: $\tilde{\omega}_t \leftarrow \bar{\omega}_t + N$, where $N \sim \mathcal{N}(0, \sigma^2 I_d)$
 - 8: Map back: $\tilde{g}_t \leftarrow M_t^{\text{inv}} \tilde{\omega}_t + a_t$
 - 9: Update model: $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
 - 10: Update mean: $a_{t+1} \leftarrow \beta_1 a_t + (1 - \beta_1) \tilde{g}_t$
 - 11: Update covariance: $\Sigma_{t+1} \leftarrow \beta_2 \Sigma_t + (1 - \beta_2) (\tilde{g}_t - a_t)(\tilde{g}_t - a_t)^\top$
 - 12: Eigendecompose: $\Sigma_{t+1} = U_t \Lambda_t U_t^\top$
 - 13: Clamp eigenvalues: $\lambda_i \leftarrow \text{Clamp}(\lambda_i, \min = h_1, \max = h_2)$
 - 14: Set $M_{t+1} \leftarrow (\gamma / \sum_i \sqrt{\lambda_i})^{1/2} \Lambda_t^{-1/4} U_t^\top$
 - 15: Set $M_{t+1}^{\text{inv}} \leftarrow (\gamma / \sum_i \sqrt{\lambda_i})^{-1/2} U_t \Lambda_t^{1/4}$
 - 16: **end for**
 - 17: **return** Final parameters θ
-

Algorithm 2 STREAMING RANK- k PCA

Require: Eigenvectors $U_t \in \mathbb{R}^{d \times k}$, eigenvalues $\Lambda_t \in \mathbb{R}^{k \times k}$, gradient $\tilde{g}_t \in \mathbb{R}^d$, mean $a_{t+1} \in \mathbb{R}^d$, factor $\beta_3 \in \mathbb{R}$, rank k

- 1: Center: $z \leftarrow \tilde{g}_t - a_{t+1}$
 - 2: Form augmented matrix: $U_{\text{aug}} \leftarrow [U_t \ z]$
 - 3: Compute: $Z \leftarrow U_{\text{aug}} \text{diag}(\sqrt{\beta_3 \lambda_1}, \dots, \sqrt{\beta_3 \lambda_k}, \sqrt{1 - \beta_3})$
 - 4: Perform SVD: $Z = V S R^\top$
 - 5: Set $U_{t+1} \leftarrow$ first k columns of V
 - 6: Set $\Lambda_{t+1} \leftarrow$ squares of the first k singular values in S
 - 7: Return: U_{t+1}, Λ_{t+1}
-

where $U_t \in \mathbb{R}^{d \times k}$ contains the top- k eigenvectors and $\Lambda_t \in \mathbb{R}^{k \times k}$ the corresponding eigenvalues, where $k \ll d$. Upon receiving a new gradient $\tilde{g}_t$, we center it using the running mean a_{t+1} , yielding $z = \tilde{g}_t - a_{t+1}$, and perform a weighted update to the covariance:

$$\Sigma_{t+1} = \beta_3 U_t \Lambda_t U_t^\top + (1 - \beta_3) z z^\top = [U_t \ z] \begin{bmatrix} \beta_3 \Lambda_t & 0 \\ 0 & 1 - \beta_3 \end{bmatrix} [U_t \ z]^\top \quad (21)$$

Rather than forming this full matrix, we compute its square root:

$$Z = [U_t \ z] \cdot \text{diag}(\sqrt{\beta_3 \lambda_1}, \dots, \sqrt{\beta_3 \lambda_k}, \sqrt{1 - \beta_3}) \in \mathbb{R}^{d \times (k+1)}. \quad (22)$$

We then perform an SVD on Z and retain the top k singular vectors and squared singular values as the updated eigenvectors and eigenvalues. We note that this computation takes $\mathcal{O}(dk^2 + k^3)$ time [14], highlighting that it is only linear in d . The rest of the procedure follows Algorithm 1 by replacing the moving average over the full covariance matrix in line 11 of Algorithm 1 with the low-rank approximation described above. Another necessary change to the algorithm is the following: since U_t is no longer a square matrix, the transformation M_t takes the gradient into the lower k -dimensional space to clip and add noise, such that the resulting M_t^{inv} returns to the full d -dimensional space. The complete version of this variant is provided as Algorithm 3 in Appendix C.

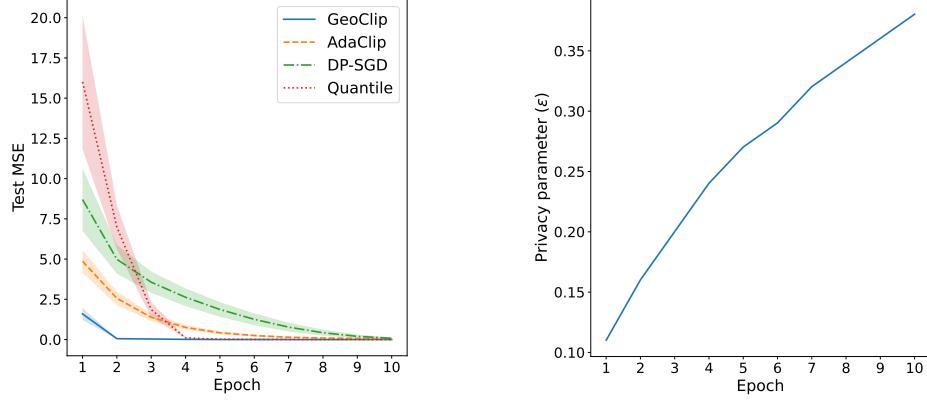


Figure 1: GeoClip results for the synthetic Gaussian dataset with 10 features. The left plot shows the average test MSE for each method over 10 epochs, with shaded regions representing the standard deviation across 20 random seeds. We observe that GeoClip achieves the fastest convergence and lowest average test MSE. The right plot shows the overall privacy budget ϵ expended for $\delta = 10^{-5}$. This plot applies to all four algorithms, as they are tuned to achieve the same privacy level for a given number of epochs.

4 Experimental Results

We present empirical results demonstrating the benefits of our proposed GeoClip framework compared to AdaClip, quantile-based clipping, and standard DP-SGD. For quantile-based clipping, we use the median of per-sample norms, which has been shown to perform well across various learning tasks [10]. All (ϵ, δ) -DP guarantees are computed using the Connect-the-Dots accountant [15]. All experiments were conducted on Google Colab using CPU resources. The code implementation is available on GitHub [16].

We start with a synthetic dataset to demonstrate how GeoClip accelerates convergence, reflecting its design motivation from Theorem 1, and then evaluate its performance on real-world datasets. All datasets are split into 80-10-10 train-validation-test sets for consistent evaluation. The results in Sections 4.1, 4.2, and 4.3 are derived from Algorithm 1. We present the results using the low-rank PCA variant in Section 4.4.

Our results indicate that our framework in Algorithm 1 is robust to the choice of hyperparameters β_1 , β_2 , and h_1 . Standard values commonly used in optimization, such as $\beta_1 = 0.99$ and $\beta_2 = 0.999$, work well in our setting. The parameter h_1 only needs to be a small positive constant (e.g., 10^{-15}) to ensure numerical stability. Since the eigenvalues are clamped to the range $[h_1, h_2]$ (line 13 in Algorithm 1), the trace term $\sum_i \sqrt{\lambda_i}$ is bounded between $d\sqrt{h_1}$ and $d\sqrt{h_2}$, where d is the dimensionality. This allows us to set the parameter γ to 1, as its effect can be absorbed by tuning h_2 . For h_2 , we have observed that the values 1 and 10 perform consistently well across datasets. Throughout all experiments, we tune only h_2 to select between these two options. A similar setup is used in Algorithm 2 with β_2 replaced by β_3 (set to 0.99), which is also robust across experiments.

4.1 Synthetic Dataset

GeoClip is designed to improve convergence, particularly in the presence of feature correlation. To empirically demonstrate this, we evaluate it on a synthetic Gaussian dataset with 20,000 samples and 10 features—five of which are correlated, while the remaining five are independent. To obtain the 5 correlated features, we first generate a $20,000 \times 5$ matrix Z and a 5×5 matrix A , each with entries drawn independently from the standard normal distribution. The correlated features are obtained by multiplying Z by A . The remaining 5 features are drawn independently from a standard multivariate normal distribution. The full feature matrix X is constructed by concatenating these two blocks. The target y is generated using a linear function with Gaussian noise: $y = Xw + b + \epsilon$, where $w \sim \mathcal{N}(0, I_{10})$ is a weight vector, $b \sim \mathcal{N}(0, 1)$ is a scalar bias term, and $\epsilon \sim \mathcal{N}(0, 0.01^2)$ is i.i.d. noise. We train a linear regression model using various private training methods for 10 epochs with a

batch size of 1024, tuning the learning rate for each method to ensure stable convergence. As shown in Figure 1 (left), GeoClip converges as early as epoch 2, while the next best method—quantile-based clipping—requires nearly twice as many epochs. Figure 1 (right) plots the privacy cost (ϵ) versus epoch, showing how faster convergence helps minimize overall privacy cost.

Although GeoClip introduces some computational and memory overhead, this cost is well justified by its empirical gains. GeoClip converges considerably faster than baseline methods, requiring fewer training iterations to reach comparable utility. While each iteration is slightly more expensive than in other baselines, this cost is partially offset by the reduced number of iterations. Combined with improved privacy, these benefits make the additional overhead a worthwhile trade-off.

As illustrated by the standard deviation bands in Figure 1, GeoClip not only achieves faster convergence but also demonstrates more stable training with reduced variance—an important property when privacy constraints limit training to a single run. In such scenarios, lower variability across runs enhances the reliability of the final model without requiring additional privacy budget.

4.2 Tabular Datasets

In addition to the synthetic dataset, we run experiments to compare the performance of different clipping strategies on three real-world datasets: Diabetes [17], Breast Cancer [18], and Android Malware [19]. We briefly describe each dataset and its corresponding learning task below.

The Diabetes dataset contains 442 samples and 10 standardized features. The continuous valued target variable indicates disease progression, making this a regression task. The Breast Cancer dataset contains 569 samples and 30 numerical features, which are standardized to zero mean and unit variance before training. The target is to predict a binary label indicating the presence or absence of cancer. The Android Malware dataset contains 4465 samples and 241 integer attributes. The target is to classify whether a program is malware or not.

For each of the three datasets, we perform a grid search over the relevant hyperparameters to identify the best model under a given (ϵ, δ) privacy budget. We train for 5 epochs using 20 random seeds and report the average performance along with the standard deviation in Tables 1, 2, and 3. We observe that our proposed GeoClip framework consistently outperforms all baseline methods across both regression and classification tasks. GeoClip achieves better performance with noticeably smaller standard deviations, indicating greater stability across random seeds.

Table 1: Diabetes dataset test MSE comparison for $\delta = 10^{-5}$, batch size = 32, and model dimension $d = 11$.

Framework	$\epsilon = 0.50$	$\epsilon = 0.86$	$\epsilon = 0.93$
GeoClip (ours)	0.073±0.015	0.044±0.003	0.039±0.009
AdaClip	0.077±0.027	0.062±0.028	0.055±0.014
Quantile	0.090±0.027	0.083±0.044	0.072±0.014
DP-SGD	0.108±0.040	0.095±0.047	0.072±0.040

Table 2: Breast Cancer dataset test accuracy (%) comparison for $\delta = 10^{-5}$, batch size = 64, and model dimension $d = 62$.

Framework	$\epsilon = 0.67$	$\epsilon = 0.8$	$\epsilon = 0.87$
GeoClip (ours)	87.87±3.32	88.57±3.37	93.63±1.63
AdaClip	84.90±5.91	85.42±5.34	87.71±5.87
Quantile	81.41±12.53	81.63±10.71	92.28±2.68
DP-SGD	77.32±6.17	79.42±9.71	85.95±4.45

Table 3: Malware dataset test accuracy (%) comparison for $\delta = 10^{-5}$, batch size = 512, and model dimension $d = 484$.

Framework	$\epsilon = 0.26$	$\epsilon = 0.49$	$\epsilon = 0.67$
GeoClip (ours)	90.77$\pm$1.83	91.64$\pm$1.26	92.67$\pm$1.63
AdaClip	88.35 $\pm$ 3.27	90.25 $\pm$ 1.33	90.23 $\pm$ 3.11
Quantile	77.84 $\pm$ 1.29	78.84 $\pm$ 1.27	81.86 $\pm$ 1.31
DP-SGD	88.04 $\pm$ 2.21	90.55 $\pm$ 1.55	90.57 $\pm$ 1.61

4.3 Final Layer Fine-Tuning

In many transfer learning scenarios, fine-tuning only the final layer is standard practice due to both its computational efficiency and minimal privacy cost. Last-layer fine-tuning is a well-suited application of GeoClip, as it involves a small number of trainable parameters, making covariance estimation more tractable. This setup also benefits from GeoClip’s faster convergence, which is particularly valuable in privacy-constrained settings where only limited training iterations are feasible.

To demonstrate this, we design an experiment where a convolutional neural network (CNN) is first trained on MNIST [20] using the Adam optimizer and then transferred to Fashion-MNIST [21] by freezing all layers except the final fully connected layer. The CNN consists of two convolutional and pooling layers followed by a linear compression layer that reduces the feature size to 50, resulting in a total of only 510 trainable parameters. We fine-tune this layer using different methods under varying privacy budgets, and present the results in Table 4.

Table 4: Final layer DP fine-tuning on Fashion-MNIST for 4 epochs and $\delta = 10^{-6}$ over 5 seeds.

Framework	$\epsilon = 0.6$	$\epsilon = 1$
GeoClip (Ours)	73.09$\pm$0.72	73.09$\pm$0.63
AdaClip	68.35 $\pm$ 0.41	69.24 $\pm$ 0.28
Quantile-based	71.78 $\pm$ 1.28	72.09 $\pm$ 1.12
DP-SGD	69.40 $\pm$ 0.82	69.83 $\pm$ 0.83

4.4 Low-Rank PCA Results

To evaluate our low-rank PCA algorithm (Algorithm 2), we construct a synthetic binary classification dataset with 20,000 samples and 400 Gaussian features, where 50 are correlated and 350 are uncorrelated. The feature matrix is constructed in the same manner as in Section 4.1. Labels are generated by applying a linear function to the features, adding Gaussian noise, and thresholding the sigmoid output. We train a logistic regression model with 802 parameters and compare our method against existing benchmarks using a rank-50 low-rank PCA approximation. As shown in the left panel of Figure 2, our approach converges faster than competing methods, even with this low-rank approximation.

We also evaluate our method on the USPS dataset [22] using logistic regression with 2,570 trainable parameters (256 input features $\times$ 10 classes + 10 biases). The USPS dataset contains 9,298 grayscale handwritten digit images (0–9), each of size 16×16 pixels. This compact benchmark is commonly used for evaluating digit classification models. For this dataset, we apply a low-rank PCA approximation with rank 100. Results are shown in the right panel of Figure 2. As with the synthetic dataset, GeoClip with low-rank PCA also achieves faster convergence on USPS compared to baseline methods. We provide ϵ -vs-iteration plots for both datasets, illustrating how faster convergence reduces overall privacy cost in Appendix D.

We compare the accuracy of GeoClip using Algorithm 1 and Algorithm 2 with $k = 50$ on the same synthetic dataset, with the results shown in Figure 3 in Appendix D. The results demonstrate that even with $k = 50$, Algorithm 2 achieves accuracy comparable to Algorithm 1, which computes the full covariance.

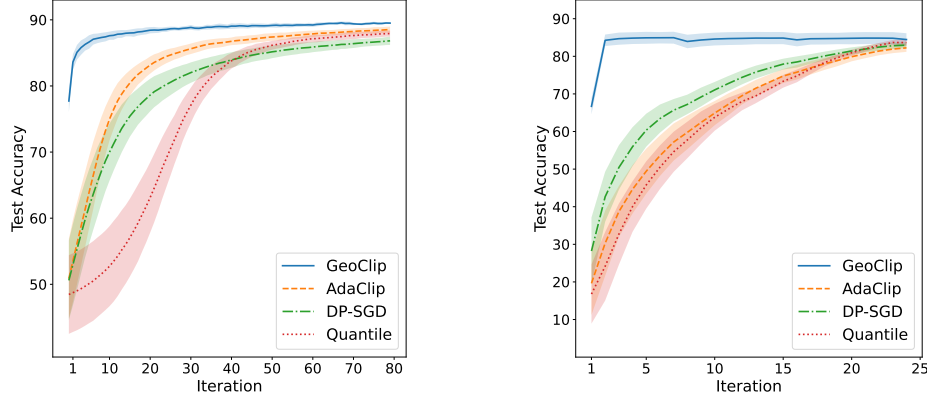


Figure 2: The left panel shows results on the synthetic Gaussian dataset with 400 features using a rank-50 PCA approximation for GeoClip. The plot displays average test accuracy (%) over 80 iterations with a batch size of 1024. GeoClip achieves the fastest convergence and highest average accuracy. The right panel shows results on the USPS dataset using a rank-100 approximation over 25 iterations with a batch size of 1024, where a similar convergence trend is observed. Shaded regions represent standard deviation across 20 random seeds.

We also conduct an ablation study on γ , h_1 , and h_2 using the USPS dataset under the same experimental settings. The results are presented in Appendix E. The findings show that varying γ results in only minor performance changes. Likewise, h_1 has no noticeable impact, and h_2 introduces small variations.

5 Conclusions

We have introduced GeoClip, a geometry-aware framework for differentially private SGD that leverages the structure of the gradient distribution to improve both utility and convergence. By operating in a basis adapted to the estimated noisy gradients, GeoClip injects noise more strategically, thereby reducing distortion without incurring additional privacy cost. We have provided a formal analysis of convergence guarantees which characterizes the optimal transformation. Our empirical results on synthetic and real-world datasets show that GeoClip consistently converges faster and outperforms existing adaptive clipping methods, improving both the mean and standard deviation of the performance metrics over multiple runs. Via low-rank approximation method, we have shown that GeoClip scales to the high-dimensional data setting, thus making it suitable for practical deployment in large, privacy-sensitive models.

Limitations. The linear transformation involved in GeoClip requires an additional computation via an eigendecomposition. Our low-rank approximation addresses that to some extent. The algorithm introduces additional hyperparameters (most notably h_2) compared to standard DP-SGD, which must be tuned for optimal performance. Our experiments have been performed on a limited collection of datasets; additional testing is needed to see how our algorithm performs in more generality.

Broader impact. As discussed in the Introduction, our work is motivated by societal concerns, with a focus on improving the theoretical limits of differentially private optimization.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback, which significantly improved the quality of this paper. This work was supported by the National Science Foundation under Grant Nos. CIF-1901243, CIF-2312666, and CIF-2007688.

References

- [1] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. Theory of Cryptography (TCC)*, pages 265–284, Berlin, Heidelberg, 2006. ISBN 3-540-32731-2, 978-3-540-32731-8.
- [2] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *EUROCRYPT*, pages 486–503, 2006.
- [3] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
- [4] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=BJ0hf1Z0b>.
- [5] Kareem Amin, Alex Kulesza, Andres Munoz, and Sergei Vassilvitskii. Bounding user contributions: A bias-variance trade-off in differential privacy. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 263–271. PMLR, 2019. URL <http://proceedings.mlr.press/v97/amin19a.html>.
- [6] Dongsheng Yu, Hao Zhang, and Wei Chen. Improve the gradient perturbation approach for differentially private optimization. In *NeurIPS 2018 Workshop on Privacy Preserving Machine Learning*, 2018. Workshop paper.
- [7] Jian Du, Song Li, Fengran Mo, and Siheng Chen. Dynamic differential-privacy preserving sgd. *ArXiv*, abs/2111.00173, 2021. URL <https://api.semanticscholar.org/CorpusID:240354620>.
- [8] Guanbiao Lin, Hongyang Yan, Guang Kou, Teng Huang, Shiyu Peng, Yingying Zhang, and Changyu Dong. Understanding adaptive gradient clipping in dp-sgd, empirically. *International Journal of Intelligent Systems*, 37(11):9674–9700, 2022. doi: <https://doi.org/10.1002/int.23001>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/int.23001>.
- [9] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.
- [10] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17455–17466. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/91cff01af640a24e7f9f7a5ab407889f-Paper.pdf.
- [11] Sergey Denisov, H Brendan McMahan, John Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. In *Advances in Neural Information Processing Systems*, volume 35, pages 5910–5924, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html.
- [12] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/kairouz21b.html>.
- [13] Christopher A. Choquette-Choo, Krishnamurthy Dvijotham, Krishna Pillutla, Arun Ganesh, Thomas Steinke, and Abhradeep Guha Thakurta. Correlated noise provably beats independent noise for differentially private learning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL <https://openreview.net/forum?id=xHmCdSArUC>.
- [14] Xiaocan Li, Shuo Wang, and Yinghao Cai. Tutorial: Complexity analysis of singular value decomposition and its variants, 2019. URL <https://arxiv.org/abs/1906.12085>.

- [15] Vadym Doroshenko, Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Connect the dots: Tighter discrete approximations of privacy loss distributions, 2022. URL <https://arxiv.org/abs/2207.04380>.
- [16] GeoClip Implementation. <https://github.com/atefeh-gilani/GeoClip>, May 2025.
- [17] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 2004.
- [18] William H. Wolberg, Olvi L. Mangasarian, and W. Nick Street. Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository, 1993. DOI: <https://doi.org/10.24432/C5DW2B>.
- [19] Parthajit Borah and Dhruba K. Bhattacharyya. TUANDROMD (Tezpur University Android Malware Dataset). UCI Machine Learning Repository, 2020. DOI: <https://doi.org/10.24432/C5560H>.
- [20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [22] Jonathan J Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

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 clearly state the paper's main contributions, including the GeoClip algorithm, its convergence guarantees, and empirical evidence of its improved privacy-utility trade-off.

Guidelines:

- The answer NA 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 NA 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: In the last part of the Conclusions section, we discuss the limitations of our work.

Guidelines:

- The answer NA 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: The assumptions are clearly stated within the theorem statements, and complete proofs are provided in the supplementary material.

Guidelines:

- The answer NA 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: For the reproducibility of the experimental results, the paper provides detailed steps to implement the proposed algorithms, specifies the setup for each dataset, and lists all relevant parameters.

Guidelines:

- The answer NA 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: [No]

Justification: Due to anonymity and other constraints, we are currently unable to release the code. However, the real-world datasets we used are publicly available, and the experimental procedures are described in sufficient detail for reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details including data splits and hyperparameters are discussed in the Experimental Results section. We either use custom optimizer updates as outlined in our proposed algorithms, or specify it otherwise.

Guidelines:

- The answer NA 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: For all our experiments, we report average performance with standard deviation over different seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: We state in the Experimental Results section that all experiments were conducted on Google Colab using only CPU resources. The computational demands are modest, and no specialized hardware (e.g., GPU or TPU) is required to reproduce the reported results. In addition, we discuss the order of the computation time of our algorithm in Section 3.

Guidelines:

- The answer NA 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 reviewed the NeurIPS Code of Ethics and confirm that our research fully complies with its principles and guidelines.

Guidelines:

- The answer NA 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 address the societal impacts of our work in the last part of the Conclusions section. We do not anticipate any negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not involve the release of models or datasets that pose a high risk of misuse. We use publicly available datasets and do not distribute any new generative or potentially harmful models.

Guidelines:

- The answer NA 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: All datasets used in the paper are publicly available and have been properly cited with their original sources.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not introduce any new datasets, models, or other assets for release.

Guidelines:

- The answer NA 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: [NA]

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

Guidelines:

- The answer NA 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: [NA]

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

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The research does not use LLMs as an important, original, or non-standard component of the core methods.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Proof of Theorem 1

Theorem (Convergence of GeoClip). *Assume f has an L -Lipschitz continuous gradient. Further, assume the stochastic gradients are bounded, i.e., $\|\nabla f_k(\theta)\| \leq G$, and have bounded variance, i.e., $\mathbb{E}_k \|\nabla f_k(\theta) - \nabla f(\theta)\|^2 \leq \sigma_g^2$. Let $\theta^* = \arg \min_{\theta \in \mathbb{R}^d} f(\theta)$ denote the optimal solution, and suppose the learning rate satisfies $\eta < \frac{2}{3L}$. Then, for the iterates $\{\theta_t\}_{t=0}^{T-1}$ produced by GeoClip with batch size 1 using the update rule $\theta_{t+1} = \theta_t - \eta \tilde{g}_t$, where $\tilde{g}_t$ is defined in (5), the average squared gradient norm satisfies:*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\theta_t)\|^2 &\leq \underbrace{\frac{f(\theta_0) - f(\theta^*)}{T \left(\eta - \frac{3L\eta^2}{2} \right)}}_{\text{Optimization gap}} + \underbrace{\frac{3L\eta}{2 - 3L\eta} \sigma_g^2}_{\text{Gradient variance term}} + \underbrace{\frac{L\eta\sigma^2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} \text{Tr} \left[(M_t^\top M_t)^{-1} \right]}_{\text{Noise-injection term}} \\ &\quad + \underbrace{\frac{2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} [\beta(a_t) (\text{Tr} (M_t^\top M_t \Sigma_t) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2)]}_{\text{Clipping error term}}, \end{aligned} \quad (23)$$

where $\theta^t = (\theta_0, \dots, \theta_t)$ represents the history of parameter values up to iteration t , $\Sigma_t = \text{Cov}(g_t | \theta^t)$, and

$$\beta(a_t) = (G + \|a_t\|) \left(G + \frac{3L\eta}{2} (G + \|a_t\|) \right). \quad (24)$$

Proof. We can express the noisy gradient as:

$$\tilde{g}_t = M_t^{-1} \tilde{\omega}_t + a_t \quad (25)$$

$$= \frac{M_t^{-1} \omega_t}{\max\{1, \|\omega_t\|\}} + M_t^{-1} N_t + a_t \quad (26)$$

$$= \frac{g_t - a_t}{\max\{\|M_t(g_t - a_t)\|, 1\}} + M_t^{-1} N_t + a_t. \quad (27)$$

Thus, the parameter update takes the form

$$\theta_{t+1} = \theta_t - \eta \tilde{g}_t \quad (28)$$

$$= \theta_t - \eta \left[\frac{g_t - a_t}{\max\{\|M_t(g_t - a_t)\|, 1\}} + M_t^{-1} N_t + a_t \right]. \quad (29)$$

We define the following quantities:

$$c_t = \frac{g_t - a_t}{\max\{\|M_t(g_t - a_t)\|, 1\}}, \quad (30)$$

$$\Delta_t = c_t - (g_t - a_t). \quad (31)$$

c_t is the clipped version of $g_t - a_t$. Δ_t quantifies the distortion due to clipping, being zero when no clipping occurs and negative when clipping is applied. By the L -Lipschitz continuity of the gradient of $f(\theta)$, we have:

$$f(\theta_{t+1}) \leq f(\theta_t) + \langle \nabla f(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2} \|\theta_{t+1} - \theta_t\|^2. \quad (32)$$

Let $\theta^t = (\theta_0, \dots, \theta_t)$ describe the entire history of states. Taking the expectation conditioned on θ^t , we obtain:

$$\begin{aligned} \mathbb{E}[f(\theta_{t+1})|\theta^t] &\leq f(\theta_t) + \mathbb{E}[\langle \nabla f(\theta_t), \theta_{t+1} - \theta_t \rangle | \theta^t] + \frac{L}{2} \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2 | \theta^t] \end{aligned} \quad (33)$$

$$= f(\theta_t) - \eta \mathbb{E}[\langle \nabla f(\theta_t), c_t + M_t^{-1} N_t + a_t \rangle | \theta^t] + \frac{L\eta^2}{2} \mathbb{E}[\|c_t + M_t^{-1} N_t + a_t\|^2 | \theta^t] \quad (34)$$

$$= f(\theta_t) - \eta \mathbb{E}[\langle \nabla f(\theta_t), c_t + a_t \rangle | \theta^t] + \frac{L\eta^2}{2} \mathbb{E}[\|c_t + a_t\|^2 | \theta^t] + \frac{L\eta^2}{2} \mathbb{E}[\|M_t^{-1} N_t\|^2 | \theta^t] \quad (35)$$

$$= f(\theta_t) - \eta \mathbb{E}[\langle \nabla f(\theta_t), c_t + a_t \rangle | \theta^t] + \frac{L\eta^2}{2} \mathbb{E}[\|c_t + a_t\|^2 | \theta^t] + \frac{L\eta^2 \sigma^2}{2} \|M_t^{-1}\|_F^2 \quad (36)$$

$$= f(\theta_t) - \eta \mathbb{E}[\langle \nabla f(\theta_t), g_t + \Delta_t \rangle | \theta^t] + \frac{L\eta^2}{2} \mathbb{E}[\|g_t + \Delta_t\|^2 | \theta^t] + \frac{L\eta^2 \sigma^2}{2} \|M_t^{-1}\|_F^2 \quad (37)$$

$$\begin{aligned} &= f(\theta_t) - \eta \|\nabla f(\theta_t)\|^2 - \eta \mathbb{E}[\langle \nabla f(\theta_t), \Delta_t \rangle | \theta^t] \\ &\quad + \frac{L\eta^2}{2} \mathbb{E}[\|g_t - \nabla f(\theta_t) + \nabla f(\theta_t) + \Delta_t\|^2 | \theta^t] + \frac{L\eta^2 \sigma^2}{2} \|M_t^{-1}\|_F^2 \end{aligned} \quad (38)$$

where (35) follows from $\mathbb{E}[N_t] = 0$ and the independence of N_t from g_t (and thus from $c_t + a_t$). The equality (36) follows because $\mathbb{E}[NN^\top] = \sigma^2 I_d$, and $\|M_t^{-1}\|_F$ represents the Frobenius norm of M_t^{-1} . The equality (37) follows from $c_t + a_t = \Delta_t + g_t$. The last equality follows because

$$\mathbb{E}[\langle \nabla f(\theta_t), g_t + \Delta_t \rangle | \theta^t] = \mathbb{E}[\langle \nabla f(\theta_t), \Delta_t \rangle | \theta^t] + \mathbb{E}[\langle \nabla f(\theta_t), g_t \rangle | \theta^t] \quad (39)$$

$$= \mathbb{E}[\langle \nabla f(\theta_t), \Delta_t \rangle | \theta^t] + \nabla f(\theta_t)^\top \mathbb{E}[g_t | \theta^t] \quad (40)$$

$$= \mathbb{E}[\langle \nabla f(\theta_t), \Delta_t \rangle | \theta^t] + \|\nabla f(\theta_t)\|^2, \quad (41)$$

where (41) follows because g_t is an unbiased estimator of the true gradient, i.e., $\mathbb{E}[g_t | \theta^t] = \nabla f(\theta_t)$. From Jensen's inequality, we have:

$$\begin{aligned} &\mathbb{E}[\|g_t - \nabla f(\theta_t) + \nabla f(\theta_t) + \Delta_t\|^2 | \theta^t] \\ &\leq 3 \left(\mathbb{E}[\|g_t - \nabla f(\theta_t)\|^2 | \theta^t] + \|\nabla f(\theta_t)\|^2 + \mathbb{E}[\|\Delta_t\|^2 | \theta^t] \right). \end{aligned} \quad (42)$$

From the Cauchy-Schwarz inequality, we have:

$$\mathbb{E}[\langle \nabla f(\theta_t), \Delta_t \rangle | \theta^t] \leq \|\nabla f(\theta_t)\| \mathbb{E}[\|\Delta_t\| | \theta^t]. \quad (43)$$

Plugging (42) and (43) into (38), we obtain:

$$\begin{aligned} \mathbb{E}[f(\theta_{t+1}) | \theta^t] &\leq f(\theta_t) - \eta \|\nabla f(\theta_t)\|^2 + \eta \|\nabla f(\theta_t)\| \mathbb{E}[\|\Delta_t\| | \theta^t] \\ &\quad + \frac{3L\eta^2}{2} \left[\mathbb{E}[\|g_t - \nabla f(\theta_t)\|^2 | \theta^t] + \|\nabla f(\theta_t)\|^2 + \mathbb{E}[\|\Delta_t\|^2 | \theta^t] \right] + \frac{L\eta^2 \sigma^2}{2} \|M_t^{-1}\|_F^2. \end{aligned} \quad (44)$$

To bound $\mathbb{E}[\|\Delta_t\| | \theta^t]$ and $\mathbb{E}[\|\Delta_t\|^2 | \theta^t]$, we first bound $\Pr(\|\Delta_t\| > 0 | \theta^t)$ and $\|\Delta_t\|$ given θ^t and $\|\Delta_t\| > 0$. Using Markov's inequality, we obtain:

$$\Pr(\|\Delta_t\| > 0 | \theta^t) = \Pr(\|M_t(g_t - a_t)\|^2 > 1 | \theta^t) \quad (45)$$

$$\leq \Pr(\|M_t(g_t - a_t)\|^2 \geq 1 | \theta^t) \quad (46)$$

$$\leq \mathbb{E}[\|M_t(g_t - a_t)\|^2 | \theta^t] \quad (47)$$

$$= \mathbb{E}[\|M_t(g_t - \mathbb{E}[g_t | \theta^t] + \mathbb{E}[g_t | \theta^t] - a_t)\|^2 | \theta^t] \quad (48)$$

$$= \mathbb{E}[\|M_t(g_t - \mathbb{E}[g_t | \theta^t])\|^2 | \theta^t] + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \quad (49)$$

$$= \text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2. \quad (50)$$

Using (30) and (31), we obtain:

$$\|\Delta_t\| = \left\| \left(\frac{1}{\max\{\|M_t(g_t - a_t)\|, 1\}} - 1 \right) (g_t - a_t) \right\| \quad (51)$$

$$\leq \|g_t - a_t\| \quad (52)$$

$$\leq \|g_t\| + \|a_t\| \quad (53)$$

$$\leq G + \|a_t\|, \quad (54)$$

where the final inequality follows from $\|\nabla f_k(\theta)\| \leq G$. Therefore,

$$\mathbb{E}[\|\Delta_t\| | \theta^t] = \Pr(\|\Delta_t\| > 0 | \theta^t) \mathbb{E}[\|\Delta_t\| | \|\Delta_t\| > 0, \theta^t] \quad (55)$$

$$\leq (G + \|a_t\|) \left(\text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \right). \quad (56)$$

Similarly, we obtain the following bound:

$$\mathbb{E}[\|\Delta_t\|^2 | \theta^t] \leq (G + \|a_t\|)^2 \left(\text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \right). \quad (57)$$

By plugging (56) and (57) into (44) and rearranging the terms, we obtain

$$\begin{aligned} & \mathbb{E}[f(\theta_{t+1}) | \theta^t] \\ & \leq f(\theta_t) + \left(\frac{3L\eta^2}{2} - \eta \right) \|\nabla f(\theta_t)\|^2 + \eta(G + \|a_t\|) \left(G + \frac{3L\eta}{2}(G + \|a_t\|) \right) \\ & \quad \left[\text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2 \right] + \frac{3L\eta^2}{2} \mathbb{E}[\|g_t - \nabla f(\theta_t)\|^2 | \theta^t] + \frac{L\eta^2\sigma^2}{2} \|M_t^{-1}\|_F^2. \end{aligned} \quad (58)$$

Above, we have used the fact that $\|\nabla f(\theta_t)\| \leq G$, which itself follows from Jensen's inequality and the assumption that $\|\nabla f_k(\theta)\| \leq G$. Let

$$\beta(a_t) = (G + \|a_t\|) \left(G + \frac{3L\eta}{2}(G + \|a_t\|) \right). \quad (59)$$

Rearranging the terms, applying the law of total expectation (now taking expectation over the entire history of states θ^t), and using the bound $\mathbb{E}_k[\|\nabla f_k(\theta) - \nabla f(\theta)\|^2] \leq \sigma_g^2$, we obtain:

$$\begin{aligned} \left(\eta - \frac{3L\eta^2}{2} \right) \mathbb{E} \|\nabla f(\theta_t)\|^2 & \leq \mathbb{E} f(\theta_t) - \mathbb{E} f(\theta_{t+1}) + \frac{3L\eta^2\sigma_g^2}{2} + \frac{L\eta^2\sigma^2}{2} \mathbb{E} \|M_t^{-1}\|_F^2 \\ & \quad + \eta \mathbb{E} [\beta(a_t) (\text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2)]. \end{aligned} \quad (60)$$

Summing over $t = 0$ to $T - 1$ and applying the telescoping sum, we get:

$$\begin{aligned} \left(\eta - \frac{3L\eta^2}{2} \right) \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\theta_t)\|^2 & \leq f(\theta_0) - \mathbb{E} f(\theta_T) + \frac{3L\eta^2\sigma_g^2}{2} T + \frac{L\eta^2\sigma^2}{2} \sum_{t=0}^{T-1} \mathbb{E} \|M_t^{-1}\|_F^2 \\ & \quad + \eta \sum_{t=0}^{T-1} \mathbb{E} [\beta(a_t) (\text{Tr}(\text{Cov}(M_t g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2)] \end{aligned} \quad (61)$$

Dividing by T , assuming $\eta \leq \frac{2}{3L}$, applying the identity $\|M_t^{-1}\|_F^2 = \text{Tr}((M_t^\top M_t)^{-1})$, and using the fact that $\mathbb{E} f(\theta_T) \geq f(\theta^*)$, we obtain:

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\theta_t)\|^2 \\ & \leq \frac{f(\theta_0) - f(\theta^*)}{T \left(\eta - \frac{3L\eta^2}{2} \right)} + \frac{3L\eta}{2 - 3L\eta} \sigma_g^2 + \frac{L\eta\sigma^2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} [\text{Tr}(M_t^\top M_t)^{-1}] \\ & \quad + \frac{2}{T(2 - 3L\eta)} \sum_{t=0}^{T-1} \mathbb{E} [\beta(a_t) (\text{Tr}(M_t^\top M_t \text{Cov}(g_t | \theta^t)) + \|M_t(\mathbb{E}[g_t | \theta^t] - a_t)\|^2)]. \end{aligned} \quad (62)$$

□

B Proof of Theorem 2

We would like to solve the following optimization

$$\begin{aligned} & \underset{M_t}{\text{minimize}} \quad \text{Tr}(M_t^\top M_t)^{-1} \\ & \text{subject to} \quad \text{Tr}(M_t^\top M_t \Sigma_t) \leq \gamma. \end{aligned} \quad (63)$$

Defining A_t as the Gram matrix of M_t , i.e., $A_t = M_t^\top M_t$, we can reformulate the optimization problem as:

$$\begin{aligned} & \underset{A_t}{\text{minimize}} \quad \text{Tr}(A_t^{-1}) \\ & \text{subject to} \quad \text{Tr}(A_t \Sigma_t) \leq \gamma. \end{aligned} \quad (64)$$

Both the objective and the constraint are convex in A , so to solve the problem, we introduce the Lagrangian function:

$$\ell(A_t, \mu) = \text{Tr}(A_t^{-1}) + \mu (\text{Tr}(A_t \Sigma_t) - \gamma), \quad (65)$$

Taking the derivative of $\ell(A_t, \mu)$ with respect to A_t and setting the derivative to zero for optimality, we get

$$A_t^{-2} = \mu \Sigma_t. \quad (66)$$

and so

$$A_t = \frac{1}{\sqrt{\mu}} \Sigma_t^{-\frac{1}{2}}. \quad (67)$$

Substituting A_t in the constraint $\text{Tr}(A_t \Sigma_t) = \gamma$, we get

$$\frac{1}{\sqrt{\mu}} \text{Tr}(\Sigma_t^{\frac{1}{2}}) = \gamma. \quad (68)$$

Solving for μ :

$$\sqrt{\mu} = \frac{1}{\gamma} \text{Tr}(\Sigma_t^{\frac{1}{2}}). \quad (69)$$

Thus, the optimal A_t is:

$$A_t = \frac{\gamma}{\text{Tr}(\Sigma_t^{\frac{1}{2}})} \Sigma_t^{-\frac{1}{2}}. \quad (70)$$

Using the eigen decomposition, we write Σ_t as:

$$\Sigma_t = U_t \Lambda_t U_t^\top, \quad (71)$$

where U_t is an orthogonal matrix whose columns are the eigenvectors of Σ_t , and

$$\Lambda_t = \text{diag}(\lambda_1, \dots, \lambda_d)$$

is a diagonal matrix containing the corresponding eigenvalues. We now have

$$\Sigma_t^{\frac{1}{2}} = U_t \Lambda_t^{\frac{1}{2}} U_t^\top, \quad \Sigma_t^{-\frac{1}{2}} = U_t \Lambda_t^{-\frac{1}{2}} U_t^\top, \quad (72)$$

Thus, the final expression for A_t is:

$$A_t = \frac{\gamma}{\text{Tr}(\Lambda_t^{\frac{1}{2}})} U_t \Lambda_t^{-\frac{1}{2}} U_t^\top. \quad (73)$$

Therefore, we have

$$M_t^\top M_t = \frac{\gamma}{\text{Tr}(\Lambda_t^{\frac{1}{2}})} U_t \Lambda_t^{-\frac{1}{2}} U_t^\top, \quad (74)$$

and

$$M_t = \left(\frac{\gamma}{\text{Tr}(\Lambda_t^{\frac{1}{2}})} \right)^{1/2} \Lambda_t^{-\frac{1}{4}} U_t^\top. \quad (75)$$

Since $\text{Tr}(\Lambda_t^{1/2}) = \sum_{i=1}^d \sqrt{\lambda_i}$, we can simplify M_t as follows:

$$M_t = \left(\frac{\gamma}{\sum_{i=1}^d \sqrt{\lambda_i}} \right)^{1/2} \Lambda_t^{-1/4} U_t^\top. \quad (76)$$

Algorithm 3 GEOCLIP WITH RANK- k PCA

Require: Dataset $\mathcal{D}$, model f_θ , loss $\mathcal{L}$, learning rate η , noise scale σ , steps T , rank k , hyperparameters $h_1, h_2, \beta_1, \beta_3$

- 1: Initialize θ , mean vector $a_0 = 0$, $U_0 = [e_1, \dots, e_k]$ where $e_i \in \mathbb{R}^d$ is the i -th standard basis vector, $\Lambda_0 = I_k$
- 2: Compute transform $M_0 \leftarrow (\gamma/k)^{1/2} \Lambda_0^{-1/4} U_0^\top$ and $M_0^{\text{inv}} \leftarrow (\gamma/k)^{-1/2} U_0 \Lambda_0^{1/4}$
- 3: **for** $t = 0$ to T **do**
- 4: Sample a data point (x_t, y_t)
- 5: Compute gradient $g_t \leftarrow \nabla_\theta \mathcal{L}(f_\theta(x_t), y_t)$
- 6: Center and transform: $\omega_t \leftarrow M_t(g_t - a_t)$
- 7: Clip: $\bar{\omega}_t \leftarrow \omega_t / \max(1, \|\omega_t\|_2)$
- 8: Add noise: $\tilde{\omega}_t \leftarrow \bar{\omega}_t + N$, where $N \sim \mathcal{N}(0, \sigma^2 I_k)$
- 9: Map back: $\tilde{g}_t \leftarrow M_t^{\text{inv}} \tilde{\omega}_t + a_t$
- 10: Update model: $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
- 11: Update mean: $a_{t+1} \leftarrow \beta_1 a_t + (1 - \beta_1) \tilde{g}_t$
- 12: Update eigenspace: $(U_{t+1}, \Lambda_{t+1}) \leftarrow \text{STREAMING RANK-}k \text{ PCA}(U_t, \Lambda_t, \tilde{g}_t, a_{t+1}, \beta_3, k)$
- 13: Clamp eigenvalues: $\lambda_i \leftarrow \text{CLAMP}(\lambda_i, \min = h_1, \max = h_2)$
- 14: Set $M_{t+1} \leftarrow (\gamma / \sum_i \sqrt{\lambda_i})^{1/2} \Lambda_{t+1}^{-1/4} U_{t+1}^\top$
- 15: Set $M_{t+1}^{\text{inv}} \leftarrow (\gamma / \sum_i \sqrt{\lambda_i})^{-1/2} U_{t+1} \Lambda_{t+1}^{1/4}$
- 16: **end for**
- 17: **return** Final parameters θ

C GeoClip with Low-Rank PCA

Computing and storing the full gradient covariance matrix becomes infeasible in high dimensions. To address this, we propose a low-rank approximation method in Algorithm 3, which incorporates a rank- k PCA step described in the [STREAMING RANK- \$k\$ PCA](#) algorithm.

To compute M_0 and M_0^{inv} in line 2 of Algorithm 3, we use the simplification $\text{Tr}(\Lambda_0^{1/2}) = \sum_{i=1}^k 1 = k$. Also, in line 13, each λ_i depends on t , but we omit the subscript for notational simplicity.

STREAMING RANK- k PCA

Require: Eigenvectors $U_t \in \mathbb{R}^{d \times k}$, eigenvalues $\Lambda_t \in \mathbb{R}^{k \times k}$, gradient $\tilde{g}_t \in \mathbb{R}^d$, mean $a_{t+1} \in \mathbb{R}^d$, factor $\beta_3 \in \mathbb{R}$, rank k

- 1: Center: $z \leftarrow \tilde{g}_t - a_{t+1}$
- 2: Form augmented matrix: $U_{\text{aug}} \leftarrow [U_t \ z]$
- 3: Compute: $Z \leftarrow U_{\text{aug}} \text{diag}(\sqrt{\beta_3 \lambda_1}, \dots, \sqrt{\beta_3 \lambda_k}, \sqrt{1 - \beta_3})$
- 4: Perform SVD: $Z = V S R^\top$
- 5: Set $U_{t+1} \leftarrow$ first k columns of V
- 6: Set $\Lambda_{t+1} \leftarrow$ squares of the first k singular values in S
- 7: Return: U_{t+1}, Λ_{t+1}

D Additional Plots Related to Section 4.4

The middle and left panels of Figure 3 show the privacy cost (ε) versus iteration curves corresponding to the accuracy-iteration plots in Figure 2 (Section 4.4, Low-Rank PCA Results). These plots highlight how faster convergence reduces overall privacy cost. For example, on the USPS dataset (Figure 2), GeoClip achieves high accuracy within the first few iterations, requiring only $\varepsilon \approx 0.15$, whereas quantile-based clipping takes about 24 iterations to reach similar accuracy, incurring a higher privacy cost of $\varepsilon \approx 0.5$.

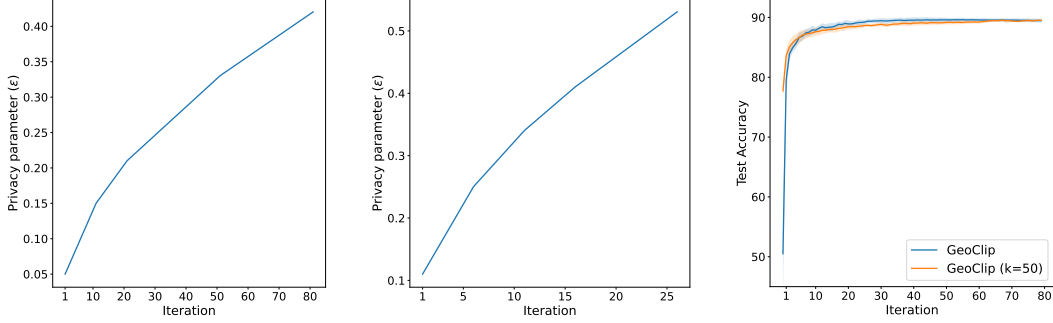


Figure 3: The left plot shows the overall privacy budget ϵ spent on training the synthetic Gaussian dataset with 400 features for $\delta = 10^{-5}$, while the middle plot shows the same for the USPS dataset with $\delta = 10^{-5}$. These plots apply to all four algorithms, which are tuned to achieve the same privacy level for a given number of iterations. The right panel compares GeoClip’s performance using the full-covariance and low-rank approximation algorithms.

We compare the accuracy of GeoClip using Algorithm 1 and Algorithm 2 with $k = 50$ on the same synthetic dataset described in Section 4.4, as shown in the right panel of Figure 3. The results show that even with $k = 50$, Algorithm 2 achieves accuracy comparable to Algorithm 1, which computes the full covariance.

E Ablation Study

We conduct an ablation study on γ , h_1 , and h_2 using the USPS dataset, under the same settings as in Section 4.4. As shown, varying γ does not significantly affect performance. The results indicate no noticeable impact from h_1 , and only minor variations with h_2 .

Table 5: Test accuracy (%) for different values of γ on the USPS dataset with $h_2 = 10$ and $h_1 = 10^{-15}$

Method	$\gamma = 0.2$	$\gamma = 0.6$	$\gamma = 1.0$
GeoClip ($k = 50$)	85.537	86.182	85.752

Table 6: Test accuracy (%) for different values of h_2 on the USPS dataset with $h_1 = 10^{-15}$ and $\gamma = 1$

Method	$h_2 = 1$	$h_2 = 10$	$h_2 = 100$	$h_2 = 1000$
GeoClip ($k = 50$)	86.021	85.752	85.752	85.752

Table 7: Test accuracy (%) for different values of h_1 on the USPS dataset with $h_2 = 10$ and $\gamma = 1$

Method	$h_1 = 10^{-6}$	$h_1 = 10^{-10}$	$h_1 = 10^{-15}$
GeoClip ($k = 50$)	85.752	85.752	85.752