
Adaptive Sigmoid Clipping for Balancing the Direction–Magnitude Mismatch Trade-off in Differentially Private Learning

Faeze Moradi Kalarde
ECE Department
University of Toronto
faeze.moradi@mail.utoronto.ca

Ali Bereyhi
ECE Department
University of Toronto
ali.bereyhi@utoronto.ca

Ben Liang
ECE Department
University of Toronto
liang@ece.utoronto.ca

Min Dong
ECSE Department
Ontario Tech University
Min.Dong@ontariotechu.ca

Abstract

Differential privacy (DP) limits the impact of individual training data samples by bounding their gradient norms through clipping. Conventional clipping operations assign unequal scaling factors to sample gradients with different norms, leading to a direction mismatch between the true batch gradient and the aggregation of the clipped gradients. Applying a smaller but identical scaling factor to all sample gradients alleviates this direction mismatch; however, it intensifies the magnitude mismatch by excessively reducing the aggregation norm. This work proposes a novel clipping method, termed adaptive sigmoid (AdaSig), which uses a sigmoid function with an adjustable saturation slope to clip the sample gradients. The slope is adaptively adjusted during the training process to balance the trade-off between direction mismatch and magnitude mismatch, as the statistics of sample gradients evolve over the training iterations. Despite AdaSig’s adaptive nature, our convergence analysis demonstrates that differentially private stochastic gradient descent (DP-SGD) with AdaSig clipping retains the best-known convergence rate under non-convex loss functions. Evaluating AdaSig on sentence and image classification tasks across different datasets shows that it consistently improves learning performance compared with established clipping methods.

1 Introduction

Deep learning models can expose individual data samples to privacy risks and are vulnerable to multiple types of practical attacks [2, 5, 26, 29]. To mitigate these risks, differential privacy (DP) [10] has emerged as a widely adopted standard in machine learning. Differentially private training constrains the impact of individual data samples on the training process, ensuring that the resulting model statistically resembles the one trained by the exclusion of an arbitrary data sample. As a result, inferring the presence of an individual data sample in the training set becomes challenging when observing the model during training. Differentially private stochastic gradient descent (DP-SGD) is a prominent privacy-preserving algorithm that incorporates DP into the SGD algorithm through two steps: (i) bounding sensitivity through clipping, and (ii) introducing uncertainty by adding Gaussian noise [1]. The first step limits the ℓ_2 -sensitivity of the batch gradient against the addition or removal of data samples by upper bounding the magnitude of the sample gradients using a clipping operation.

A widely used clipping operation, which we refer to as *vanilla clipping* [1, 8], scales each sample gradient $\mathbf{g}_{i,t}$ with a norm greater than C to constrain its norm to C , as

$$\tilde{\mathbf{g}}_{i,t} = \mathbf{g}_{i,t} \min \left(1, \frac{C}{\|\mathbf{g}_{i,t}\|} \right), \quad (1)$$

where the constant C is termed the *clipping threshold*. In the second step, a zero-mean Gaussian noise is added to the aggregation of the clipped gradients with a standard deviation that is proportional to the clipping threshold.

Applying the clipping operation to sample gradients introduces a deviation between the aggregation of clipped gradients, i.e., $\sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_{i,t}$, and the true batch gradient, i.e., $\sum_{i \in \mathcal{B}_t} \mathbf{g}_{i,t}$, which is referred to as *bias* [3, 13, 19]. The bias caused by clipping manifests in two forms: (i) *direction deviation*, and (ii) *magnitude deviation* from the true batch gradient. Direction deviation can be particularly severe in many classical clipping methods that are based on vanilla clipping operation in (1). This follows from the fact that the vanilla clipping maps all gradient norms greater than C to C , and hence neglects the information contained in the diversity of the sample gradient norms within the batch. This can be interpreted as assigning unequal weights to sample gradients, which causes a significant direction deviation. This misalignment can steer training steps towards undesired regions of the loss landscape and hinder training convergence [7, 11].

In response, the per-sample adaptive clipping (PSAC) method [36] was proposed to mitigate the direction deviation. Figure 1 shows the variation of the clipped gradient norm $\|\tilde{\mathbf{g}}_{i,t}\|$ versus the gradient norm $\|\mathbf{g}_{i,t}\|$ for the vanilla clipping and compares it with PSAC for the same threshold $C = 0.1$. We also include the curve for the automatic clipping method (Auto-S) [4], which is proposed to eliminate the need for tuning the clipping threshold. This figure shows the effectiveness of each method in reducing direction deviation: the more closely a curve can be approximated by a line passing through the origin, the better the clipping method approximates equal scaling of gradient samples with different magnitudes. We observe that the vanilla clipping method hard-clips the magnitude of sample gradients that are larger than C , resulting in a limited linear region¹ on the curve. Using Auto-S, the hard-thresholding behavior of the vanilla method is slightly reduced; nonetheless, its restricted linear region leads to a similar scaling as the vanilla method. In contrast, the PSAC clipping method has a wider linear region than vanilla clipping and Auto-S, as it preserves the same scaling factor across a broader range of sample gradient norms, thereby providing a more accurate approximation of equal scaling.

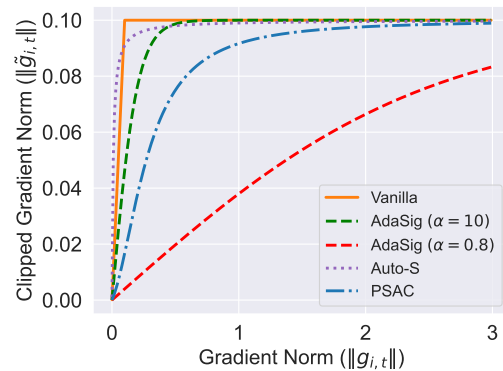


Figure 1: Clipped gradient norm vs. gradient norm. Comparing linear regions for different clipping methods. PSAC has a wider linear region compared with the vanilla clipping. The linear region of AdaSig can be adjusted by altering α .

While expanding the linear region can reduce direction deviation, it also decreases the norm of the aggregation of the clipped gradients compared with that of the true batch gradient (i.e., larger magnitude deviation). This occurs because the wide linear region of the clipping curve results in a small scaling factor, causing sample gradients with smaller magnitudes to have excessively small norms after scaling. This suggests that the span of the linear region on the clipping curve needs to be carefully adjusted to better balance the trade-off between direction deviation and magnitude deviation. Moreover, since the sample gradient statistics change across different training iterations, it is necessary to adaptively adjust the linear region to balance the trade-off throughout the training process.

Contributions Motivated by the above observation, we propose a novel clipping method called Adaptive Sigmoid (AdaSig), where the magnitude of the clipped gradient is determined from the original gradient magnitude using a scaled and shifted sigmoid function. AdaSig features a saturation slope parameter that adaptively adjusts the span of the linear region throughout training to balance the trade-off between direction and magnitude mismatches.

¹“Linear region” is used in contrast to the “lazy region” to refer to the part of the curve that can be well approximated by a straight line passing through the origin.

Figure 1 also shows the clipping curves for two choices of the saturation slope (denoted by α) in AdaSig. As α shrinks, the linear region of the clipping operation expands resulting in better approximation of equal gradient scaling and thus smaller *direction deviation*. Conversely, larger α preserves the magnitude of smaller sample gradients thus reducing the *magnitude deviation*. To adaptively adjust the clipping operation based on the gradient statistics throughout training, we treat α as a learnable parameter and update it in each iteration to minimize the empirical loss. To the best of our knowledge, this is the first study that designs a clipping method to balance the trade-off between the magnitude deviation and direction deviation. This work makes the following contributions:

- We introduce AdaSig, a novel clipping strategy that allows balancing the direction deviation and magnitude deviation by adjusting a parameter α . We treat α as a learnable parameter and derive its SGD-based update for empirical loss minimization.
- We develop a new algorithm for DP-SGD with AdaSig clipping (DP-SGD-AdaSig), by establishing an update rule for α that effectively preserves privacy.
- AdaSig’s unique clipping structure and varying slope complicate its convergence analysis such that existing analyses are inapplicable. However, we derive a convergence bound for DP-SGD-AdaSig in non-convex loss settings, showing that its privacy–utility trade-off matches the best-known bound in the literature.
- We conduct experiments on image and sentence classification tasks. Performance comparison with existing clipping methods shows the efficacy of AdaSig in learning enhancement through a proper balance between direction deviation and magnitude deviation.

2 Related Work

Extensions of Vanilla Clipping In vanilla clipping, the clipping threshold substantially impacts training performance [19]. Large C results in large privacy noise variance, while small C leads to aggressive gradient clipping, causing a significant bias. Several studies adaptively adjust C to improve the bias–variance trade-off. In [3], the optimal threshold is estimated using gradient quantiles during training. In [13], the clipping threshold is treated as a learnable parameter, allowing it to be optimized dynamically over the training iterations. However, these works focus on the bias–variance trade-off for vanilla clipping and cannot balance the magnitude and direction deviations introduced by the bias for a *fixed* variance level. This limitation arises from the inherent nature of vanilla clipping, where reducing the direction deviation (increasing C) inevitably increases the variance.

Some studies combine vanilla clipping with error feedback to mitigate the bias introduced by clipping [41, 16]. This technique is applicable to any clipping method, including ours, and is orthogonal to the focus of this study.

Other Clipping Methods Automatic clipping (Auto-S) [4] and normalized SGD (NSGD) [39] are similar approaches that aim to eliminate the need for tuning the clipping threshold. Their key idea is to normalize gradients as $\tilde{\mathbf{g}}_{i,t} = \frac{C\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|+r}$ for some small positive constant r . Although these methods are effective for their main goal, they suffer considerably from direction deviation, as they assign larger weights to gradients with smaller magnitudes. These small gradients are often opposite to the true batch gradient direction, as empirically shown in [36]. To address the direction deviation issue, PSAC [36] modifies the clipping operation to $\tilde{\mathbf{g}}_{i,t} = \frac{C\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\| + \frac{r}{\|\mathbf{g}_{i,t}\|} + r}$ for a positive r . This adjustment reduces the weights assigned to small gradients and thus mitigates the direction deviation. While PSAC effectively reduces the direction deviation, it cannot adaptively balance the trade-off between magnitude and direction deviations during training because its linear region span remains fixed throughout the process (Figure 1). To address this limitation, we propose AdaSig, which dynamically adjusts its saturation slope to better balance this trade-off.

3 Background and Preliminaries

Learning Model and SGD We consider a supervised setting, where the training dataset consists of N samples, denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, with $x_i \in \mathcal{X}$ representing the vector of input features and $y_i \in \mathcal{Y}$ denoting the corresponding target output. The goal is to learn a model $f_{\theta} : \mathcal{X} \rightarrow \mathcal{Y}$, with parameters $\theta \in \mathbb{R}^d$, that maps input features to the target outputs. The discrepancy between

the predicted outputs $\hat{y}_i = f_{\theta}(x_i)$ and the actual targets y_i is measured by the sample loss function, denoted as $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. Let $h^i(\theta)$ denote the loss of sample i as a function of the model parameters, i.e., $h^i(\theta) \triangleq \ell(f_{\theta}(x_i), y_i)$. The population loss is defined as $L(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{P}} [\ell(f_{\theta}(x), y)]$, where $\mathcal{P}$ denotes the unknown data distribution. In iteration t , stochastic gradient descent (SGD) updates the model parameters using the average gradient of the loss over the samples within a batch of data as

$$\theta_{t+1} = \theta_t - \frac{\lambda}{B} \sum_{i \in \mathcal{B}_t} \nabla_{\theta} h^i(\theta) \Big|_{\theta=\theta_t}, \quad (2)$$

where $\nabla_{\theta} h^i(\theta)$ denotes the gradient of the loss function with respect to (w.r.t.) the model parameters, θ_t denotes the model at iteration t , $\mathcal{B}_t \subseteq \mathcal{D}$ is the batch of data at iteration t , B denotes the expected batch size, and λ is the learning rate.

Differential Privacy The randomized mechanism $M : \mathcal{S} \rightarrow \mathcal{R}$ with domain $\mathcal{S}$ and range $\mathcal{R}$ satisfies (ϵ, δ) -differential privacy (DP) if, for any two *neighboring* datasets $S, S' \in \mathcal{S}$, i.e., S' is formed by adding or removing a single sample from S , and for any output set $R \subseteq \mathcal{R}$,

$$\Pr[M(S) \in R] \leq e^{\epsilon} \Pr[M(S') \in R] + \delta, \quad (3)$$

where $\Pr[A]$ denotes the probability of the event A . Inequality (3) implies that for small ϵ and δ , the output distributions of the mechanism under neighboring datasets approach each other, making it harder to detect the contribution of an individual sample from the mechanism's output.

Differentially Private SGD (DP-SGD) Let $\mathbf{g}_{i,t} \in \mathbb{R}^d$ denote the gradient of loss for sample i at iteration t , i.e., $\mathbf{g}_{i,t} \triangleq \nabla_{\theta} h^i(\theta) \Big|_{\theta=\theta_t}$. DP-SGD achieves privacy protection via two steps: (i) The gradient samples within the batch are first passed through a clipping method that returns a clipped gradient $\tilde{\mathbf{g}}_{i,t}$ whose ℓ_2 -norm is bounded by a constant C . Basic DP-SGD uses the vanilla clipping method in (1). The aim of clipping is to bound the ℓ_2 -sensitivity of the aggregation of the gradients by C . (ii) After aggregating the clipped gradients over the batch, zero-mean Gaussian privacy noise is added to introduce uncertainty. To ensure the privacy of training, the variance of the noise process is scaled with C^2 . Consequently, in iteration t of DP-SGD, we update the model parameters as

$$\theta_{t+1} = \theta_t - \frac{\lambda}{B} \left(\sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_{i,t} + \mathbf{n}_t \right), \quad (4)$$

where $\mathbf{n}_t \sim \mathcal{N}(0, C^2 \sigma^2 \mathbf{I}_d)$ is the Gaussian privacy noise with σ being the noise multiplier determined by the privacy budget (ϵ, δ) .

4 AdaSig Clipping

Vanilla clipping in (1) outputs a vector with the smallest error norm relative to the input vector. It however introduces a notable *direction deviation* between the true batch gradient and the sum of clipped gradients due to unequal scaling of different sample gradients. A naive solution to this issue is to scale all sample gradients by the same coefficient such that all resulting norms fall below the clipping threshold. This approach requires the scaling coefficient to be small enough so that the largest gradient norm is reduced to the clipping threshold. As a result, gradients with smaller norms are scaled down to very small norms, leading to an aggregation with excessively small magnitude (*magnitude deviation*). Since the privacy noise is proportional to the clipping threshold, this low-magnitude aggregation results in a very low signal-to-noise ratio, which hinders training performance.

In light of the above discussion, a proper clipping operation should not only have a bounded range to constrain the ℓ_2 sensitivity, but it should also ensure that the output norm is monotonically increasing w.r.t. the input norm, with an adjustable linear region span that balances the trade-off between direction and magnitude deviations. Since the sigmoid function, with an adjustable saturation slope, satisfies both of these properties, we propose a clipping operation based on the sigmoid function, termed adaptive sigmoid (AdaSig). Further details on the choice of the sigmoid function for clipping are provided in Appendix A. This operation is illustrated in the sequel.

Clipping with Sigmoid Function For the clipping threshold C , we define AdaSig clipping as

$$\psi_{\alpha}(x) = C \left(\frac{2}{1 + e^{-\alpha x}} - 1 \right), \quad (5)$$

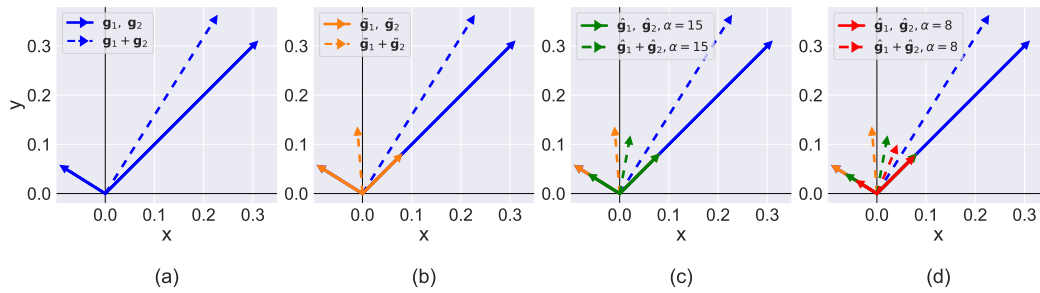


Figure 2: Illustrative example: sample gradients are shown by solid arrows and their aggregation is indicated by a dashed arrow. The horizontal and vertical axes correspond to the first and second entries of the two-dimensional vectors, respectively. (a) shows true gradients and their summation. (b) shows the gradients clipped by the vanilla method and their aggregation in orange. (c) and (d) show the gradients clipped by AdaSig for $\alpha = 15$ and $\alpha = 8$ in green and red, respectively. The adjustability of α enables AdaSig to balance the trade-off between direction and magnitude deviations.

where $\alpha > 0$ is the saturation slope. Using the AdaSig function, the sample gradient $\mathbf{g}_{i,t}$ is clipped as

$$\tilde{\mathbf{g}}_{i,t} = \psi_{\alpha}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}. \quad (6)$$

One can see that this clipping addresses the two mentioned desired properties: (i) since $\psi_{\alpha}(x) \in [0, C]$ for $x \geq 0$, the sum of the clipped gradients by AdaSig has a bounded ℓ_2 -sensitivity equal to C . (ii) The scaling is a monotonic function of the input norm and can adjust its linear region by varying α . Figure 1 shows the clipping curves of AdaSig for two choices of α . As the figure shows, the parameter α enables us to sketch an adjustable trade-off between the two extreme cases of clipping: by increasing α , the curve approaches the vanilla clipping method, which results in a larger magnitude after aggregating the clipped gradients at the expense of higher direction deviation. Conversely, decreasing α leads to a better approximation of equal scaling, and thus less direction deviation, while reducing the magnitude of the sum of the clipped gradients (i.e., larger magnitude deviation).

Illustrative Example Before developing our differentially private training algorithm via AdaSig, we provide a numerical example to illustrate AdaSig's capability to balance direction and magnitude deviations. Let's consider sample gradients as

$$\mathbf{g}_1 = \begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}, \mathbf{g}_2 = \begin{bmatrix} -0.08 \\ 0.05 \end{bmatrix}, \quad (7)$$

that are shown by solid blue arrows in Figure 2. The sum vector, i.e., true batch gradient, is shown with a dashed blue arrow. The clipped gradients using the vanilla method with $C = 0.1$, and their aggregation are shown in orange by solid and dashed arrows, respectively. Comparing the sum of the clipped gradients with the true batch gradient, we observe a notable direction deviation.

We next consider AdaSig clipping with the same threshold $C = 0.1$ and two different choices of the saturation slope, namely, $\alpha = 15$ and $\alpha = 8$. The result for $\alpha = 15$ is shown by green arrows with the dashed one indicating the sum of the clipped gradients. Evidently, the direction deviation is slightly reduced in this case compared with the vanilla method. As we reduce α to $\alpha = 8$, shown by red arrows, the direction deviation is further reduced. It is however observed that better alignment in this case is achieved at the cost of a smaller magnitude of the aggregated vector. This explains the natural trade-off between direction deviation and magnitude deviation. Appendix H.1 provides additional insights into the effect of α on this trade-off.

5 DP-SGD with AdaSig Clipping

We develop a DP-SGD algorithm with AdaSig clipping, which we refer to as DP-SGD-AdaSig. Nonetheless, AdaSig can be readily incorporated into other differentially private optimizers. DP-

SGD-AdaSig consists of two steps: (i) Differentially private update of the model parameters, and (ii) Differentially private update of the saturation slope α .

5.1 Differentially Private Update of Model Parameters

Rewriting AdaSig clipping in terms of α_t in (6), we have

$$\tilde{\mathbf{g}}_{i,t} = \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}, \quad (8)$$

where we add the subscript t to the slope, i.e., α_t , to indicate that it is an adjustable parameter and changes over iterations. We further define $s(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ as the aggregation of clipped gradients in iteration t , and use $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ to represent the private estimation of batch gradient, i.e.,

$$s(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) \triangleq \sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}, \quad (9a)$$

$$\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) \triangleq s(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) + \mathbf{n}_t^s, \quad (9b)$$

where $\mathbf{n}_t^s \sim \mathcal{N}(0, C^2 \sigma_s^2 \mathbf{I}_d)$ is privacy noise with per-coordinate variance $C^2 \sigma_s^2$. It is worth mentioning that $s(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ includes $\boldsymbol{\theta}_t$ as an argument, since it depends on $\boldsymbol{\theta}_t$ through sample gradient $\mathbf{g}_{i,t}$ computed at $\boldsymbol{\theta}_t$.

Using the above notation, the differentially private SGD update of model parameters $\boldsymbol{\theta}_t$ is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\lambda}{B} \hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t). \quad (10)$$

Note that in (10), the parameter C can be absorbed into the learning rate λ , since $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ linearly scales with C . In fact, the noise term $\mathbf{n}_t^s$ and all clipped gradients in $s(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ are scaled by C (see (5), and note that the variance of $\mathbf{n}_t^s$ is $C^2 \sigma_s^2$). We hence simplify the parameter space in the remainder of this work by setting $C = 1$.

5.2 Differentially Private Update of Saturation Slope

At iteration t , we aim to update α_t to minimize the empirical loss. To this end, we update α in the descent direction of the loss gradient. Developing a differentially private update rule for α_t therefore involves two main steps: (i) computing the derivative of the loss w.r.t. α , and (ii) obtaining a differentially private estimate of this derivative. Based on this estimate, a differentially private update rule can then be constructed using one-step gradient descent. We detail these steps in the subsequent sections.

Loss Derivative with Respect to α Let $\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t}$ denote the derivative of the loss on sample i w.r.t. α at iteration t . We approximate it as

$$\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t} \approx -\frac{\lambda}{B} \mathbf{g}_{i,t}^\top r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}), \quad (11)$$

where

$$r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) \triangleq \sum_{i \in \mathcal{B}_t} \frac{2e^{-\alpha_t \|\mathbf{g}_{i,t}\|} \mathbf{g}_{i,t}}{(1 + e^{-\alpha_t \|\mathbf{g}_{i,t}\|})^2}. \quad (12)$$

The derivation of the approximation in (11) is provided in Appendix C.

Let $\frac{\partial h(\boldsymbol{\theta}_t)}{\partial \alpha_t}$ denote the batch-averaged loss derivative w.r.t. α at iteration t . We approximate it as

$$\begin{aligned} \frac{\partial h(\boldsymbol{\theta}_t)}{\partial \alpha_t} &= \frac{1}{B} \sum_{i \in \mathcal{B}_t} \frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t} \\ &\stackrel{(a)}{\approx} -\frac{\lambda}{B^2} \left(\sum_{i \in \mathcal{B}_t} \mathbf{g}_{i,t} \right)^\top r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}), \end{aligned} \quad (13)$$

where (a) applies the approximation in (11).

Differentially Private Estimate of Loss Derivative We next compute a differentially private estimate of the loss derivative approximation in (13). The right-hand side (RHS) of (13) comprises two factors. For the first factor, i.e., $\sum_{i \in \mathcal{B}_t} \mathbf{g}_{i,t}$, a private estimate is given by $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$, which is computed for update of $\boldsymbol{\theta}_t$ in (10). We can hence obtain a differentially private estimate for $\frac{\partial h(\boldsymbol{\theta}_t)}{\partial \alpha_t}$ by finding a private estimate of the second factor, i.e., $r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$. The standard approach for differentially private estimation is to (i) restrict the ℓ_2 -sensitivity of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$, and (ii) add a privacy noise term, whose standard deviation is proportional to the bounded ℓ_2 -sensitivity. However, the former step is not required in this case, as the ℓ_2 -sensitivity of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ is bounded. This is shown in the following lemma.

Lemma 5.1. *The ℓ_2 -sensitivity of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ is bounded from above by $\Delta_{\alpha_t} = 0.448/\alpha_t$.*

Proof. See Appendix D. □

Using Lemma 5.1, we compute the private estimate of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ as

$$\hat{r}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) = r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) + \mathbf{n}_t^r, \quad (14)$$

where $\mathbf{n}_t^r \sim \mathcal{N}(0, \Delta_{\alpha_t}^2 \sigma_r^2 \mathbf{I}_d)$ is the Gaussian privacy noise with the noise multiplier σ_r . Replacing the first and second factors in (13) with their private estimates $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ and $\hat{r}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$, respectively, we obtain the differentially private update for α_t as

$$\alpha_{t+1} = \alpha_t + \frac{\lambda_\alpha \lambda}{B^2} \hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)^\top \hat{r}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}), \quad (15)$$

for some learning rate λ_α . Since both $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ and $\hat{r}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$ are noisy, their product can deviate notably from the derivative approximation in (13). To make the update rule robust, we consider only the sign of their product, which is more resilient to noise. Also, to ensure that the value of α_t remains positive, we adopt a standard exponential update, by rewriting (15) in the exponential form as

$$\alpha_{t+1} = \alpha_t \cdot e^{\lambda_\alpha \text{sign}(\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)^\top \hat{r}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}))}. \quad (16)$$

The DP-SGD-AdaSig algorithm is outlined in Algorithm 1. DP-SGD-AdaSig incurs negligible extra computational and memory costs compared with the vanilla clipping method. A complexity analysis is provided in Appendix B.

Algorithm 1 DP-SGD-AdaSig

```

1: Input:  $\mathcal{D}, T, \alpha_0 > 0, \lambda_\alpha, \lambda, B, \sigma_s, \sigma_r, \boldsymbol{\theta}_0$ 
2: Output:  $\{\boldsymbol{\theta}_t\}_{t=1}^T$ 
3:  $\hat{r}(\boldsymbol{\theta}_{-1}, \alpha_{-1}; \mathcal{B}_{-1}) = \mathbf{0}$  // Initialization
4: for  $t \in \{0, \dots, T-1\}$  do
5:   Form  $\mathcal{B}_t$  via Poisson sampling with rate  $B/N$ . // Poisson sampling
6:   for  $(x_i, y_i) \in \mathcal{B}_t$  in parallel do
7:      $\tilde{\mathbf{g}}_{i,t} = \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}$ 
8:      $\mathbf{p}_{i,t} = \frac{2e^{-\alpha_t \|\mathbf{g}_{i,t}\|} \mathbf{g}_{i,t}}{(1+e^{-\alpha_t \|\mathbf{g}_{i,t}\|})^2}$ 
9:   end for
10:   $\Delta_{\alpha_t} = \frac{0.448}{\alpha_t}$  // Sensitivity bound
11:   $\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) = \sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_{i,t} + \mathcal{N}(0, \sigma_s^2 \mathbf{I}_d)$ 
12:   $\hat{r}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) = \sum_{i \in \mathcal{B}_t} \mathbf{p}_{i,t} + \mathcal{N}(0, \Delta_{\alpha_t}^2 \sigma_r^2 \mathbf{I}_d)$ 
13:   $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\lambda}{B} \hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$  // Model update
14:   $\alpha_{t+1} = \alpha_t \cdot e^{\lambda_\alpha \text{sign}(\hat{s}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)^\top \hat{r}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}))}$  //  $\alpha$  update
15: end for

```

6 Privacy Analysis of DP-SGD-AdaSig

The privacy guarantee of the proposed algorithm is provided directly by extending the results for DP-SGD. To this end, we first recall that each iteration of the DP-SGD algorithm is a Gaussian

mechanism. The entire DP-SGD algorithm is then viewed as a composition of these Gaussian mechanisms, where each iteration's output serves as the input for the next iteration. The privacy accountant method is then used to obtain the noise multiplier σ in (4), which ensures the algorithm satisfies (ϵ, δ) -DP after T iterations [1]. We denote this guarantee as $\sigma = \text{Accountant}(q, T, \epsilon, \delta)$, where $q = B/N$ is the batch sampling probability.

Unlike the DP-SGD algorithm, DP-SGD-AdaSig in each iteration consists of two parallel Gaussian mechanisms: the first one returns the Gaussian approximation of the batch gradient, $\hat{s}(\theta_t, \alpha_t; \mathcal{B}_t)$, that is used for updating both the model in (10) and the saturation slope α_t in (16). The second mechanism returns the Gaussian approximation of $r(\theta_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$ denoted by $\hat{r}(\theta_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$, which is used to update α_t in (16). From a privacy perspective, these two parallel Gaussian mechanisms together behave as a single Gaussian mechanism [3, Theorem 1]. This is formalized in Proposition 6.1.

Proposition 6.1. *In each iteration of DP-SGD-AdaSig, the two parallel Gaussian mechanisms $\hat{s}(\cdot)$ and $\hat{r}(\cdot)$, with noise multipliers σ_s and σ_r , are equivalent to a single Gaussian mechanism with noise multiplier $\sigma = (\sigma_r^{-2} + \sigma_s^{-2})^{-1/2}$. Hence, each iteration of DP-SGD-AdaSig incurs the same privacy cost as an iteration of DP-SGD with noise multiplier σ .*

Compared with Theorem 1 in [3], Proposition 6.1 uses σ_r instead of $2\sigma_r$ for privatizing the second mechanism. This is because the privacy noise of the second mechanism in [3] privatizes a positive scalar, but in our algorithm, the noise privatizes $r(\theta_t, \alpha_t; \mathcal{B}_t)$ which may have non-positive entries.

7 Convergence Analysis of DP-SGD-AdaSig

Since the clipping structure of AdaSig is substantially different from established methods in the literature, existing convergence analyses are not directly applicable. Furthermore, the adaptive adjustment of α adds complexity to the analysis of training convergence. In this section, we provide theoretical guarantees on the convergence of DP-SGD-AdaSig. We consider the following assumptions on the population loss $L(\theta)$ and its gradient, denoted by $\mathbf{g}_t \triangleq \nabla L(\theta_t)$.

Assumption 7.1. (Lower bounded loss) There exists a scalar L^* such that $L(\theta) \geq L^*, \forall \theta \in \mathbb{R}^d$.

Assumption 7.2. (Smoothness) $L(\theta)$ is β -smooth, i.e., $\exists \beta > 0$ such that

$$L(\theta_1) \leq L(\theta_2) + \langle \nabla L(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\beta}{2} \|\theta_1 - \theta_2\|^2, \forall \theta_1, \theta_2 \in \mathbb{R}^d. \quad (17)$$

Assumption 7.3. (Bounded gradient). The ℓ_2 -norm of the gradient of $L(\theta)$ is bounded, i.e., there exists $G > 0$, such that $\|\mathbf{g}_t\| \leq G$ for any $\theta_t \in \mathbb{R}^d$.

Assumption 7.4. (Unbiased per-sample gradient). The per-sample gradients, i.e., $\mathbf{g}_{i,t}$ for $i \in \mathcal{B}_t$, are i.i.d. and are unbiased estimators of $\mathbf{g}_t$. This means $\mathbf{g}_{i,t} \sim \mathbf{v}_t, \forall i$, where $\mathbf{v}_t = \mathbf{g}_t + \Delta_t, \forall t$, with $\mathbb{E}[\Delta_t] = \mathbf{0}$. Additionally, $\mathbf{g}_{i,t}$ is distributed centrally symmetric around $\mathbf{g}_t$, i.e., $\Delta_t \stackrel{D}{=} -\Delta_t$.

Assumptions 7.1 and 7.2 are standard assumptions used for analyzing the convergence of first-order optimization algorithms [4, 7, 35, 41]. Assumption 7.3 is widely used for analysis of clipping in DP algorithms [32, 35, 41]. Assumption 7.4 (unbiasedness and symmetric distribution) is commonly used in both literature not concerned with DP [6, 30, 33, 38] and DP literature [4]. Specifically, the symmetric per-sample gradient distribution assumption is also empirically verified in [7, Figure 3].

We next present the main convergence result in Theorem 7.5, which shows that DP-SGD-AdaSig yields a bounded weighted average of the expected squared gradient norms. The bound is expressed explicitly in terms of the privacy budget and other parameters.

Theorem 7.5. *Let Assumptions 7.1–7.4 hold. For a constant $r \geq G$, let $\alpha_0 \propto 1/r$. Moreover, let*

$$\lambda_\alpha \propto 1/T \text{ and } \lambda = \sqrt{\frac{2N^2\epsilon^2(L(\theta_0) - L^*)}{\beta T(2N^2\epsilon^2 + d\nu_s^2 T \log(\frac{1}{\delta}))}}. \text{ Then, under the privacy budget } (\epsilon, \delta), \text{ there exist a}$$

constant $\nu_s > 0$ such that DP-SGD-AdaSig satisfies the following inequality:

$$\frac{1}{T} \sum_{t=0}^{T-1} \Pr \left\{ \|\Delta_t\| \leq r \right\} \mathbb{E} \|\mathbf{g}_t\|^2 \leq 2r\tilde{G}\sqrt{\beta\mathcal{J}}, \quad (18)$$

for some constant $\tilde{G} > 0$, where

$$\mathcal{J} \triangleq \frac{L(\theta_0) - L^*}{T} \left(1 + \frac{d\nu_s^2 T \log(\frac{1}{\delta})}{2N^2 \epsilon^2} \right). \quad (19)$$

Proof. See Appendix E.2. □

We note regarding the bound in Theorem 7.5: (i) The weight of expected gradient norm squared in the expression on the left-hand side (LHS) of (18) corresponds to the probability that the magnitude of the deviation of sample gradients from $\mathbf{g}_t$ does not exceed r . (ii) Tighter privacy guarantees, i.e., smaller values of (ϵ, δ) , lead to higher values on the RHS. This reflects the trade-off between privacy and learning performance. In Corollary 7.6 below, we observe that DP-SGD-AdaSig achieves the asymptotic convergence rate of $\mathcal{O}(\frac{\sqrt{d}}{N\epsilon})$, which matches the known privacy–utility trade-off bound in the literature for vanilla clipping that does not allow adaptation [7, 12].

Corollary 7.6. *As T approaches infinity, the RHS of (18) converges to $\mathcal{O}(\frac{\sqrt{d \log(1/\delta)}}{N\epsilon})$.*

When the sample gradients have a bounded variance, the weights in the LHS of the expression in Theorem 7.5 can be further simplified. This yields an upper bound for the average of the expected gradient norm squared, which is given in the following corollary.

Corollary 7.7. *Let $\mathbb{E}\|\Delta_t\|^2 \leq \zeta^2$ for any iteration t . Then, the bound in Theorem 7.5 reduces to*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\mathbf{g}_t\|^2 \leq 2R\tilde{G}\sqrt{\beta\mathcal{J}}, \quad (20)$$

where $R \triangleq \frac{r_*^2}{r_* - \zeta}$, and $r_* = \max\{2\zeta, G\}$.

Proof. See Appendix E.3. □

8 Experiments

We evaluate AdaSig on image and sentence classification tasks.² Under equal privacy budget (ϵ, δ) , we compare the performance of AdaSig against four baselines:³ (i) Vanilla method [1], (ii) method proposed in [3], (iii) Auto-S [4], and (iv) PSAC [36]. We further present an ablation study for AdaSig in Appendix H, where we compare the performance of AdaSig with sigmoid clipping using a fixed α .

8.1 Image Classification Task

We conduct experiments on five image classification datasets including MNIST [20], FashionMNIST [37], CIFAR-10 [18], ImageNette [15] (a 10-class subset of ImageNet [9]), and CelebA [23].

Setting For MNIST, FashionMNIST and CIFAR-10, we train the CNN architectures used in [27, 34, 4], i.e., a 4-layer CNN for MNIST and FashionMNIST, and an 8-layer CNN for CIFAR-10. The simulation setup and hyperparameters are adopted from [34]. For ImageNette, we use ResNet-9 with group normalization (instead of batch normalization) and *without* scale normalization, following the setup considered in [17, 4], with the minor exception that the learning rate does not decay. For CelebA, we use the same ResNet-9 architecture as for ImageNette, with the same simulation setup as in [4]. Since each image in CelebA dataset has 40 labels, we run two sets of experiments on this dataset: (i) We consider a binary classification problem that aims to predict the 'Smiling' label only. (ii) We consider the problem of multi-label classification with all the available 40 labels in the dataset. The detailed settings, including hyperparameters for each method, are provided in Appendix F.2.

Performance Comparison Table 1 (first six rows) shows the average test accuracy across five different runs of each dataset. AdaSig outperforms all baselines by achieving higher average accuracies

²The GitHub repository for our implementation is available at <https://github.com/faezemoradik/AdaptiveSigmoidClipping.git>.

³Methods such as [41] are not directly comparable, as they do not propose a new clipping strategy but instead incorporate error feedback into existing ones, which can be applied to any clipping method, including AdaSig.

Table 1: Average test accuracy (in percentages) with 95% confidence intervals on five runs. Bold numbers indicate p -value < 0.05 in the t-test performed to identify statistical significance.

DATASET	MODEL	(ϵ, δ)	VANILLA	METHOD IN [3]	AUTO-S	PSAC	AdaSig
MNIST	4-LAYER CNN	$(3, 10^{-5})$	98.11 ± 0.04	98.13 ± 0.03	98.04 ± 0.04	98.12 ± 0.04	98.14 ± 0.05
FASHIONMNIST	4-LAYER CNN	$(3, 10^{-5})$	86.27 ± 0.24	86.28 ± 0.19	86.20 ± 0.26	86.35 ± 0.19	86.68 ± 0.04
CIFAR-10	8-LAYER CNN	$(3, 10^{-5})$	61.49 ± 0.13	61.53 ± 0.11	61.21 ± 0.25	61.93 ± 0.14	62.54 ± 0.20
IMAGENETTE	RESNET-9	$(8, 10^{-4})$	62.02 ± 0.91	62.07 ± 0.78	61.44 ± 0.69	62.30 ± 0.42	64.65 ± 0.34
CELEBA (MULTI-LABEL)	RESNET-9	$(8, 5 \times 10^{-6})$	88.51 ± 0.03	88.53 ± 0.02	88.48 ± 0.03	88.46 ± 0.02	88.78 ± 0.03
CELEBA (SMILING)	RESNET-9	$(8, 5 \times 10^{-6})$	90.87 ± 0.28	90.88 ± 0.26	90.73 ± 0.19	90.90 ± 0.26	90.95 ± 0.14
SST-2	ROBERTA-BASE	$(3, \frac{1}{2N})$	87.43 ± 0.49	87.46 ± 0.47	87.18 ± 0.56	87.48 ± 0.32	88.44 ± 0.17
QNLI	ROBERTA-BASE	$(3, \frac{1}{2N})$	84.32 ± 0.16	84.36 ± 0.17	84.51 ± 0.13	84.58 ± 0.14	84.92 ± 0.12

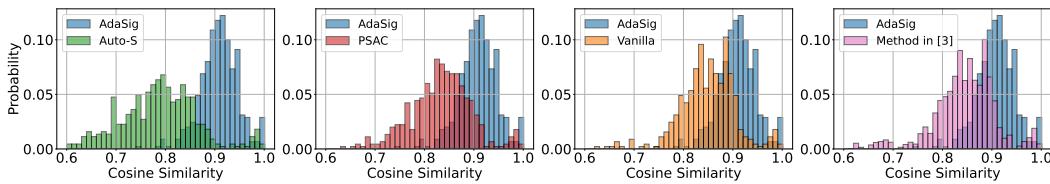


Figure 3: Comparison of cosine similarity during training ResNet-9 on ImageNette.

across all datasets. Conducting a statistical t-test further confirms the statistical significance of this improvement (p -value < 0.05) on 4 out of 6 datasets [14].

Cosine Similarity Analysis To assess AdaSig’s ability to balance the direction deviation and magnitude deviation, we measure the cosine similarity between the true batch gradient and the aggregation of clipped gradients. The results are shown in Figure 3, where we compare the histogram of the cosine similarity while training ResNet-9 on ImageNette. Notably, AdaSig’s histogram is concentrated at larger cosine similarities as compared with the baselines, which indicates the effectiveness of AdaSig clipping in reducing the direction deviation. Additional results are presented in Appendix G to elaborate on the trade-off between the direction and magnitude deviations.

8.2 Sentence Classification Task

We conduct sentence classification experiments on SST-2 [31], for sentiment classification, and QNLI [28], for question-answering inference both taken from the GLUE benchmark.

Setting We use the pre-trained RoBERTa-base model and conduct full parameter fine-tuning with DP consideration following the same setting as in [21, 4]. The detailed settings, including hyperparameters for each method, are provided in Appendix F.3.

Performance Comparison Table 1 (last two rows) shows the average test accuracy over five runs for each dataset. We observe that AdaSig achieves statistically significantly (p -value < 0.05) higher average test accuracies as compared with all baselines on both datasets.

9 Conclusions

We proposed AdaSig clipping, a novel clipping operation for differentially private training that balances the trade-off between direction and magnitude deviations incurred by clipping. It achieves this balance by adaptively adjusting its saturation slope throughout training based on information obtained from sample gradients. Our convergence analysis demonstrates that DP-SGD with AdaSig clipping retains the best-known convergence rate in the non-convex loss setting. Experiments on image and sentence classification tasks demonstrate that AdaSig clipping consistently improves training performance compared to existing clipping methods.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

- [1] 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 ACM SIGSAC Conference on Computer and Communications Security*, 2016.
- [2] Jason M Altschuler and Kunal Talwar. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.
- [3] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2021.
- [4] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023.
- [5] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium*, 2021.
- [6] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *Information Theory and Applications Workshop (ITA)*, 2018.
- [7] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.
- [8] Savelii Chezhegov, Aleksandr Beznosikov, Samuel Horváth, and Eduard Gorbunov. Convergence of clipped-SGD for convex (L_0, L_1) -smooth optimization with heavy-tailed noise. *arXiv preprint arXiv:2505.20817*, 2025.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [10] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers, Hanover, MA, USA, 2014.
- [11] Maria S Esipova, Atiyeh Ashari Ghomi, Yaqiao Luo, and Jesse C Cresswell. Disparate impact in differential privacy from gradient misalignment. In *International Conference on Learning Representations*, 2023.
- [12] Huang Fang, Xiaoyun Li, Chenglin Fan, and Ping Li. Improved convergence of differential privateSGD with gradient clipping. In *International Conference on Learning Representations*, 2023.
- [13] Filippo Galli, Catuscia Palamidessi, and Tommaso Cucinotta. Online sensitivity optimization in differentially private learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.
- [14] Sander Greenland, Stephen J Senn, Kenneth J Rothman, John B Carlin, Charles Poole, Steven N Goodman, and Douglas G Altman. Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations. *European journal of epidemiology*, 31(4):337–350, 2016.

- [15] Jeremy Howard. ImageNette: A smaller subset of 10 easily classified classes from ImageNet, March 2019. URL <https://github.com/fastai/imagenette>. Dataset; fast.ai.
- [16] Rustem Islamov, Samuel Horvath, Aurelien Lucchi, Peter Richtarik, and Eduard Gorbunov. Double momentum and error feedback for clipping with fast rates and differential privacy. *arXiv preprint arXiv:2502.11682*, 2025.
- [17] Helena Klause, Alexander Ziller, Daniel Rueckert, Kerstin Hammernik, and Georgios Kaissis. Differentially private training of residual networks with scale normalisation. *arXiv preprint arXiv:2203.00324*, 2022.
- [18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Master’s thesis, University of Toronto, 2009.
- [19] Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.
- [20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- [21] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2021.
- [22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.
- [25] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.
- [26] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. SoK: Security and privacy in machine learning. In *IEEE European Symposium on Security and Privacy*, 2018.
- [27] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [28] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.
- [29] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.
- [30] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don’t decay the learning rate, increase the batch size. In *International Conference on Learning Representations*, 2018.
- [31] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2013.
- [32] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private GLMs. In *International Conference on Artificial Intelligence and Statistics*, 2021.
- [33] Mandt Stephan, Matthew D Hoffman, David M Blei, et al. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

- [34] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.
- [35] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2017.
- [36] Tianyu Xia, Shuheng Shen, Su Yao, Xinyi Fu, Ke Xu, Xiaolong Xu, and Xing Fu. Differentially private learning with per-sample adaptive clipping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [38] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021.
- [39] Xiaodong Yang, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Normalized/clipped SGD with perturbation for differentially private non-convex optimization. *arXiv preprint arXiv:2206.13033*, 2022.
- [40] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.
- [41] Xinwei Zhang, Zhiqi Bu, Steven Wu, and Mingyi Hong. Differentially private SGD without clipping bias: An error-feedback approach. In *International Conference on Learning Representations*, 2024.

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 main contribution of the work is proposing AdaSig, a clipping method for balancing direction–magnitude mismatch trade-off in differentially private learning. We explain this trade-off, limitations of existing methods, and how our AdaSig addresses them.

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: We clarify the existence of a trade-off between direction and magnitude deviation and discuss even though our method can balance this trade-off, these deviations cannot be eliminated simultaneously. We also clarify the scope of the work, which is differentially-private learning, making non differentially-private training beyond its scope.

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: We state all assumptions for our theoretical proofs and convergence analysis, which are all established assumptions that were used in previous works too. All theorems in the work are also supported by a proof.

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: We provide complete explanations on our experimental setup such as the datasets used, model architectures, etc. in Section 8. We also explain our hardware and software information.

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: [Yes]

Justification: We release all our implementation code to enable reproducibility. All datasets that were used in this work are established benchmarks that are publicly available.

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: We provide detailed explanations on our experimental settings in Section 8.

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: All our experiments were repeated five times and the confidence intervals for the five runs are also provided. Furthermore, the statistical significance of the improved performance offered by our method was validated for each dataset by conducting a statistical t-test.

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 provide detailed information on both software and hardware used to conduct experiments in this work.

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: This work is in complete alignment with the NeurIPS Code of Ethics. It does not involve any human studies, is in complete agreement with the data-related criteria, and there's no anticipated societal impact associated with it.

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: Our study proposes a method for improving privacy-preserving training of ML models, which we consider a positive societal impact. We do not anticipate any negative societal impact for our work.

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: We do not release any data or model that has a high risk of misuse.

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: Our work is licensed under CC BY-SA 4.0, and we are aware of its terms of use.

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: [\[Yes\]](#)

Justification: Our proposed methodology as well as the released software are well-documented.

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: This work 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: This work 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 methodology presented in this paper does not involve LLMs.

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.

APPENDIX

The appendix is structured as follows: Appendix A presents details on the choice of the sigmoid function for clipping. The complexity analysis for DP-SGD-AdaSig is given in Appendix B. An approximation of the sample loss derivative is provided in Appendix C. The proof of Lemma 5.1 is presented in Appendix D. The proof of Theorem 7.5 and the preliminary lemmas and theorems related to the convergence analysis are included in Appendix E. The detailed experimental setup is presented in Appendix F. A numerical analysis of the direction and magnitude deviations is provided in Appendix G. The ablation study of AdaSig is presented in Appendix H. Finally, Appendix I and Appendix J report the variation of α_t across training iterations and the numerical convergence results, respectively.

A Details on the Choice of Sigmoid Function for Clipping

The sigmoid function is simple yet well-suited to our goal of adaptively balancing the trade-off between direction and magnitude deviations. In particular, it enables control over the range of the linear span in Figure 1 through the parameter α . Although PSAC [36] and Auto-S [4] could be extended to variants that adjust their linear region span by varying the constant parameter r throughout training, the sigmoid function provides greater flexibility and control.

Specifically, in the PSAC clipping function, varying r alters the span of its linear region; however, the change in this span is confined to a limited range. This behavior can be illustrated by analyzing the PSAC clipping function, $\tilde{\mathbf{g}}_{i,t} = \frac{C\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\| + \frac{r}{\|\mathbf{g}_{i,t}\| + r}}$, in extreme cases. Due to the term $\frac{r}{\|\mathbf{g}_{i,t}\| + r}$ in the denominator, varying r from 0 to ∞ causes the clipped gradient norm curve $\|\tilde{\mathbf{g}}_{i,t}\|$ (shown in Figure 1) to transition only between two limiting curves: C and $\frac{C\|\mathbf{g}_{i,t}\|}{\|\mathbf{g}_{i,t}\| + 1}$. Consequently, the span of the linear region in PSAC is restricted to lie between these two curves, and adaptively updating r cannot effectively balance the trade-off between direction and magnitude deviations. In contrast, in the AdaSig clipping function of (6), the clipped gradient norm curve $\|\tilde{\mathbf{g}}_{i,t}\|$ (illustrated in Figure 1) can vary from C (as $\alpha \rightarrow \infty$) down to values arbitrarily close to 0 (as $\alpha \rightarrow 0$), thereby providing greater flexibility in adjusting the span of the linear region.

While the Auto-S clipping function, $\tilde{\mathbf{g}}_{i,t} = \frac{C\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\| + r}$, does not have the restricted variation of the linear region span seen in PSAC, it is still less flexible than the sigmoid function. Its linear region span is less responsive to changes in r than the sigmoid function is to α , so larger adjustments of r are required to produce a noticeable effect. In contrast, the sigmoid function, with its exponential dependence on α , offers more flexible control over the linear region and thus the direction–magnitude mismatch trade-off.

B Complexity Analysis

Compared with DP-SGD (with vanilla clipping), one can observe that DP-SGD-AdaSig (Algorithm 1) introduces a negligible increase in computational and memory costs. Specifically, with respect to computation, the algorithm performs lines 8, 12, and 14 in addition to the standard operations in vanilla clipping. Line 8 is carried out efficiently using the per-sample gradient norms computed in line 7, the computational cost of step 12 is identical to that of step 11, and line 14 incurs only a constant additional cost per iteration. Consequently, the overall computational cost introduced by these additional steps is dominated by other operations and the cost of per-sample gradient computation, making the additional overall computational cost minimal.

With respect to memory, DP-SGD-AdaSig requires extra memory to store $\hat{\mathbf{r}}(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ (line 12), which is required for the slope update in the next iteration (line 14). This additional memory cost is negligible as compared with the memory required for storing the model parameters and per-sample gradients.

C Approximation of the Sample Loss Derivative with Respect to α

Since the loss on sample i in iteration t , denoted by $h^i(\boldsymbol{\theta}_t)$, does not directly depend on α_t , to derive an update rule for α , we unroll the expression for $h^i(\boldsymbol{\theta}_t)$ and write it in terms of $\boldsymbol{\theta}_{t-1}$ and α_{t-1} . In particular, using (10), we obtain

$$h^i(\boldsymbol{\theta}_t) = h^i\left(\boldsymbol{\theta}_{t-1} - \frac{\lambda}{B} \hat{s}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})\right). \quad (21)$$

For sufficiently small learning rate for α (i.e., λ_α in (16)), α_{t-1} and α_t differ only slightly, and hence we adopt the approximation $\alpha_{t-1} \approx \alpha_t$. Consequently, the loss derivative at α_t , $\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t}$, can be approximated by the loss derivative at α_{t-1} , $\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_{t-1}}$, i.e.,

$$\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t} \approx \frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_{t-1}}. \quad (22)$$

We now apply the chain rule to (21) to compute $\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_{t-1}}$ as

$$\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_{t-1}} = (\nabla_{\boldsymbol{\theta}_t} h^i(\boldsymbol{\theta}_t))^\top \frac{\partial \boldsymbol{\theta}_t}{\partial \alpha_{t-1}}, \quad (23)$$

where the first factor in the RHS of (23) is the sample loss gradient w.r.t. model parameters, i.e., $\mathbf{g}_{i,t}$. To compute the second factor, we write

$$\begin{aligned} \frac{\partial \boldsymbol{\theta}_t}{\partial \alpha_{t-1}} &= \frac{\partial(\boldsymbol{\theta}_{t-1} - \frac{\lambda}{B} \hat{s}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}))}{\partial \alpha_{t-1}} \\ &\stackrel{(a)}{=} \frac{-\lambda}{B} \sum_{i \in \mathcal{B}_{t-1}} \frac{\partial \psi_{\alpha_{t-1}}(\|\mathbf{g}_{i,t-1}\|)}{\partial \alpha_{t-1}} \frac{\mathbf{g}_{i,t-1}}{\|\mathbf{g}_{i,t-1}\|} \\ &\stackrel{(b)}{=} \frac{-\lambda}{B} \sum_{i \in \mathcal{B}_{t-1}} \frac{2e^{-\alpha_{t-1}\|\mathbf{g}_{i,t-1}\|} \mathbf{g}_{i,t-1}}{(1 + e^{-\alpha_{t-1}\|\mathbf{g}_{i,t-1}\|})^2}, \end{aligned} \quad (24)$$

where (a) follows from the definition of $\hat{s}(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1})$ in (9b), and (b) is obtained by computing the derivative of the AdaSig function $\psi_{\alpha_{t-1}}(\cdot)$. Using the definition of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ in (12), we rewrite (24) as

$$\frac{\partial \boldsymbol{\theta}_t}{\partial \alpha_{t-1}} = \frac{-\lambda}{B} r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}). \quad (25)$$

Substituting (25) in (23), we finally get

$$\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_{t-1}} = -\frac{\lambda}{B} \mathbf{g}_{i,t}^\top r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}). \quad (26)$$

Using (22) together with (26), we have

$$\frac{\partial h^i(\boldsymbol{\theta}_t)}{\partial \alpha_t} \approx -\frac{\lambda}{B} \mathbf{g}_{i,t}^\top r(\boldsymbol{\theta}_{t-1}, \alpha_{t-1}; \mathcal{B}_{t-1}). \quad (27)$$

D Proof of Lemma 5.1: Sensitivity Bound

Proof. The ℓ_2 -sensitivity of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ w.r.t. the training samples in iteration t is defined as

$$\Delta r = \max_{\mathcal{B}_t, \mathcal{B}'_t} \left\| r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t) - r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}'_t) \right\|_2, \quad (28)$$

where $\mathcal{B}_t$ and $\mathcal{B}'_t$ are two adjacent batches of data that differ by exactly one data point. Without loss of generality, we assume that $\mathcal{B}'_t$ contains all elements of $\mathcal{B}_t$ together with an additional data point (x', y') , where x' is the input feature and y' is its corresponding target output. That is, $\mathcal{B}'_t = \mathcal{B}_t \cup \{(x', y')\}$. Moreover, let $\mathbf{g}'$ denote the gradient of sample loss at (x', y') , i.e., $\mathbf{g}' = \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}}(x'), y')$.

Considering the definition of $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ in (12), we can conclude that the terms under summation in $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}_t)$ and $r(\boldsymbol{\theta}_t, \alpha_t; \mathcal{B}'_t)$ are identical, except a single term that arises due to the extra element in $\mathcal{B}'_t$. The ℓ_2 -sensitivity in (28) can hence be upper bounded by the norm of this extra term, i.e.,

$$\Delta r \leq \max_{\mathbf{g}'} \left\| \frac{2e^{-\alpha_t} \|\mathbf{g}'\| \mathbf{g}'}{(1 + e^{-\alpha_t} \|\mathbf{g}'\|)^2} \right\|_2 \quad (29a)$$

$$= \max_{\mathbf{g}'} \frac{2e^{-\alpha_t} \|\mathbf{g}'\| \|\mathbf{g}'\|}{(1 + e^{-\alpha_t} \|\mathbf{g}'\|)^2} \quad (29b)$$

$$\stackrel{(a)}{=} \frac{1}{\alpha_t} \max_{z \geq 0} \frac{2e^{-z} z}{(1 + e^{-z})^2}, \quad (29c)$$

where (a) follows from the variable exchange $z \triangleq \alpha_t \|\mathbf{g}'\|$. We next find the solution to the maximization in (29c) by setting the derivative of the objective to zero, i.e., $\frac{\partial}{\partial z} \frac{2e^{-z} z}{(1 + e^{-z})^2} = 0$, which results in the following fixed-point equation as

$$z = \ln \frac{z + 1}{z - 1}. \quad (30)$$

Let's denote the solution for (30) by z_* . Substituting the solution, i.e., z_* , into objective (29c), we get the upper bound for Δr as

$$\Delta r \leq \frac{1}{\alpha_t} \frac{2e^{-z_*} z_*}{(1 + e^{-z_*})^2} = \frac{z_*^2 - 1}{2z_* \alpha_t}. \quad (31)$$

Solving (30) numerically for z and substituting z_* in (31) results in $\Delta r \leq 0.448/\alpha_t$, which completes the proof. $\square$

E Theoretical Convergence Analysis

We first introduce the preliminary lemmas and theorems required for proving the main theorem.

E.1 Preliminaries

Lemma E.1. *Suppose that Assumptions 7.2 and 7.4 hold. Under Algorithm 1, the expectation of the population loss difference in two consecutive iterations is upper-bounded by*

$$\mathbb{E} \left[L(\boldsymbol{\theta}_{t+1}) - L(\boldsymbol{\theta}_t) \middle| \boldsymbol{\theta}_t, \alpha_t \right] \leq -\lambda \mathbf{g}_t^\top \mathbb{E} \left[\left(\frac{2}{1 + e^{-\alpha_t} \|\mathbf{v}_t\|} - 1 \right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \middle| \boldsymbol{\theta}_t, \alpha_t \right] + \beta \lambda^2 \left(1 + \frac{d\sigma_s^2}{2B^2} \right), \quad (32)$$

where $\mathbf{g}_t = \nabla L(\boldsymbol{\theta}_t)$, $\mathbb{E}[\cdot | \boldsymbol{\theta}_t, \alpha_t]$ denote the expectation over the randomness in iteration t for given $\boldsymbol{\theta}_t$ and α_t , and $\mathbf{v}_t$ is the random process from which $\mathbf{g}_{i,t}$ is sampled, i.e., $\mathbf{g}_{i,t} \sim \mathbf{v}_t$ for $i \in \mathcal{B}_t$.

Proof. Based on Assumption 7.2, the population loss is β -smooth. Thus, we have

$$L(\boldsymbol{\theta}_{t+1}) - L(\boldsymbol{\theta}_t) \leq \mathbf{g}_t^\top (\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \frac{\beta}{2} \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|^2. \quad (33)$$

Taking expectation from both sides of (33) for given θ_t and α_t , we have

$$\begin{aligned} & \mathbb{E}[L(\theta_{t+1}) - L(\theta_t) | \theta_t, \alpha_t] \\ & \leq \mathbf{g}_t^\top \mathbb{E}[(\theta_{t+1} - \theta_t) | \theta_t, \alpha_t] + \frac{\beta}{2} \mathbb{E}[\|\theta_{t+1} - \theta_t\|^2 | \theta_t, \alpha_t] \end{aligned} \quad (34)$$

$$\begin{aligned} & \stackrel{(a)}{=} -\frac{\lambda}{B} \mathbf{g}_t^\top \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} + \mathbf{n}_t^s | \theta_t, \alpha_t\right] \\ & \quad + \frac{\beta\lambda^2}{2B^2} \mathbb{E}\left[\left\|\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} + \mathbf{n}_t^s\right\|^2 | \theta_t, \alpha_t\right] \end{aligned} \quad (35)$$

$$\begin{aligned} & \stackrel{(b)}{=} -\frac{\lambda}{B} \mathbf{g}_t^\top \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} | \theta_t, \alpha_t\right] \\ & \quad + \frac{\beta\lambda^2}{2B^2} \mathbb{E}\left[\left\|\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}\right\|^2 | \theta_t, \alpha_t\right] + \frac{\beta\lambda^2}{2B^2} \mathbb{E}[\|\mathbf{n}_t^s\|^2 | \theta_t, \alpha_t] \end{aligned} \quad (36)$$

$$\begin{aligned} & \stackrel{(c)}{\leq} -\frac{\lambda}{B} \mathbf{g}_t^\top \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} | \theta_t, \alpha_t\right] \\ & \quad + \frac{\beta\lambda^2}{2B^2} \mathbb{E}\left[\left(\sum_{i \in \mathcal{B}_t} \left\|\psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}\right\|\right)^2 | \theta_t, \alpha_t\right] + \frac{\beta\lambda^2}{2B^2} d\sigma_s^2 \end{aligned} \quad (37)$$

$$\stackrel{(d)}{\leq} -\frac{\lambda}{B} \mathbf{g}_t^\top \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} | \theta_t, \alpha_t\right] + \beta\lambda^2 \left(\frac{\mathbb{E}|\mathcal{B}_t|^2}{2B^2} + \frac{d\sigma_s^2}{2B^2}\right) \quad (38)$$

$$\stackrel{(e)}{\leq} -\frac{\lambda}{B} \mathbf{g}_t^\top \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} | \theta_t, \alpha_t\right] + \beta\lambda^2 \left(1 + \frac{d\sigma_s^2}{2B^2}\right), \quad (39)$$

where (a) follows from (10), (b) comes from the fact that $\mathbf{g}_{i,t}$ for $i \in \mathcal{B}_t$ are independent of the process $\mathbf{n}_t^s$, and the fact that $\mathbf{n}_t^s$ is zero-mean, (c) is obtained by applying the Triangle inequality to the second term in (36) and substituting the variance of $\mathbf{n}_t^s$, (d) results from the upper bound

$$\sum_{i \in \mathcal{B}_t} \left\|\psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|}\right\| \leq |\mathcal{B}_t|, \quad (40)$$

since $\psi_{\alpha_t}(x) \leq 1$ for any $x \in \mathbb{R}$, and (e) follows from the fact that, under Poisson sampling with rate B/N , we have

$$\frac{\mathbb{E}[|\mathcal{B}_t|^2]}{B^2} = \frac{B+1}{B} - \frac{1}{N} \leq 1 + \frac{1}{B} \leq 2, \quad (41)$$

where the last inequality holds since the expected batch size $B \geq 1$.

Based on Assumption 7.4, the sample gradients are identically distributed, i.e., $\mathbf{g}_{i,t} \sim \mathbf{v}_t, \forall i$. We can hence write

$$\frac{1}{B} \mathbb{E}\left[\sum_{i \in \mathcal{B}_t} \psi_{\alpha_t}(\|\mathbf{g}_{i,t}\|) \frac{\mathbf{g}_{i,t}}{\|\mathbf{g}_{i,t}\|} | \theta_t, \alpha_t\right] = \mathbb{E}\left[\psi_{\alpha_t}(\|\mathbf{v}_t\|) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} | \theta_t, \alpha_t\right]. \quad (42)$$

Substituting into (39) and replacing $\psi_{\alpha_t}(\|\mathbf{v}_t\|)$ with its definition, (32) is concluded. $\square$

Lemma E.2. Under Assumption 7.4, i.e., $\mathbf{v}_t = \mathbf{g}_t + \Delta_t$, the following equality holds:

$$\mathbf{g}_t^\top \mathbb{E}\left[\left(\frac{2}{1 + e^{-\alpha_t \|\mathbf{v}_t\|}} - 1\right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} | \theta_t, \alpha_t\right] = \frac{\|\mathbf{g}_t\|}{2} \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) | 0 \leq c \leq 1\right], \quad (43)$$

where $\gamma_t \triangleq \alpha_t \|\mathbf{g}_t\|$, $s \triangleq \frac{\|\Delta_t\|}{\|\mathbf{g}_t\|}$, $c \triangleq \frac{\mathbf{g}_t^\top \Delta_t}{\|\Delta_t\| \|\mathbf{g}_t\|}$, and

$$\begin{aligned} f(s, c, \gamma_t) & \triangleq \frac{1 + sc}{\sqrt{1 + s^2 + 2sc}} \left(\frac{2}{1 + e^{-\gamma_t \sqrt{1 + s^2 + 2sc}}} - 1\right) \\ & \quad + \frac{1 - sc}{\sqrt{1 + s^2 - 2sc}} \left(\frac{2}{1 + e^{-\gamma_t \sqrt{1 + s^2 - 2sc}}} - 1\right). \end{aligned} \quad (44)$$

Proof. We adopt an approach similar to that proposed in [4]. Let $\mathcal{H}^+$ and $\mathcal{H}^-$ denote the following two halfspaces:

$$\mathcal{H}^+ \triangleq \{\mathbf{o} \in \mathbb{R}^d | \mathbf{g}_t^\top \mathbf{o} \geq 0\}, \quad (45)$$

$$\mathcal{H}^- \triangleq \{\mathbf{o} \in \mathbb{R}^d | \mathbf{g}_t^\top \mathbf{o} \leq 0\}. \quad (46)$$

Using $\mathbf{v}_t = \mathbf{g}_t + \Delta_t$, we can write

$$\mathbf{g}_t^\top \mathbb{E} \left[\left(\frac{2}{1 + e^{-\alpha_t \|\mathbf{v}_t\|}} - 1 \right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \middle| \boldsymbol{\theta}_t, \alpha_t \right] \quad (47)$$

$$= \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \|\mathbf{g}_t + \Delta_t\|}} - 1 \right) \frac{\mathbf{g}_t^\top (\mathbf{g}_t + \Delta_t)}{\|\mathbf{g}_t + \Delta_t\|} \right] \quad (48)$$

$$\stackrel{(a)}{=} \frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t + \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^+ \right] \\ + \frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t + \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^- \right] \quad (49)$$

$$\stackrel{(b)}{=} \frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t + \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^+ \right] \\ + \frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 - 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t - \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 - 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^+ \right], \quad (50)$$

where (a) follows from conditioning the expectation w.r.t. Δ_t on two halfspaces, and using the fact that under the symmetric distribution assumption on Δ_t (i.e., $\Delta_t \stackrel{\mathcal{D}}{=} -\Delta_t$ in Assumption 7.4), we have

$$\Pr\{\Delta_t \in \mathcal{H}^+\} = \Pr\{\Delta_t \in \mathcal{H}^-\} = \frac{1}{2}, \quad (51)$$

and (b) follows again from symmetric distribution assumption on Δ_t and along with the variable exchange $\Delta_t = -\Delta_t$ in the second term of (49).

We now rewrite (50) in terms of the random variables s and $c \in [-1, 1]$ introduced in the lemma. Noting that the event $\Delta_t \in \mathcal{H}^+$ corresponds to $c \geq 0$, we can write

$$\frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t + \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 + 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^+ \right] \\ + \frac{1}{2} \mathbb{E}_{\Delta_t} \left[\left(\frac{2}{1 + e^{-\alpha_t \sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 - 2\mathbf{g}_t^\top \Delta_t}}} - 1 \right) \frac{\mathbf{g}_t^\top \mathbf{g}_t - \mathbf{g}_t^\top \Delta_t}{\sqrt{\|\mathbf{g}_t\|^2 + \|\Delta_t\|^2 - 2\mathbf{g}_t^\top \Delta_t}} \middle| \Delta_t \in \mathcal{H}^+ \right] \quad (52)$$

$$= \frac{\|\mathbf{g}_t\|}{2} \mathbb{E}_{s,c} \left[\frac{1 + sc}{\sqrt{1 + s^2 + 2sc}} \left(\frac{2}{1 + e^{-\gamma_t \sqrt{1 + s^2 + 2sc}}} - 1 \right) \middle| 0 \leq c \leq 1 \right] \\ + \frac{\|\mathbf{g}_t\|}{2} \mathbb{E}_{s,c} \left[\frac{1 - sc}{\sqrt{1 + s^2 - 2sc}} \left(\frac{2}{1 + e^{-\gamma_t \sqrt{1 + s^2 - 2sc}}} - 1 \right) \middle| 0 \leq c \leq 1 \right], \quad (53)$$

where the equality follows directly from the definitions of s and c . Expressing (53) in terms of $f(s, c, \gamma_t)$ yields the result. $\square$

Lemma E.3. The function $\frac{\psi_\alpha(x)}{x}$ with $\psi_\alpha(x)$ defined in (5) is a non-increasing function in x for $\alpha \geq 0$, $x \geq 0$.

Proof. Computing the derivative of the function w.r.t. x and simplifying the terms yields the following:

$$\frac{\partial(\psi_\alpha(x)/x)}{\partial x} = \frac{\partial \psi_\alpha(x)}{\partial x} \frac{1}{x} - \frac{1}{x^2} \psi_\alpha(x) \\ = \frac{2e^{-\alpha x}(\alpha x - \sinh \alpha x)}{x^2(1 + e^{\alpha x})^2}, \quad (54)$$

where the RHS is a non-positive term, since $\alpha x \leq \sinh \alpha x$ for $\alpha \geq 0, x \geq 0$. Thus, the derivative is non-positive, which shows that the function is non-increasing. $\square$

Lemma E.4. The function $\frac{\partial \psi_\alpha(x)}{\partial x}$ with $\psi_\alpha(x)$ defined in (5) is a non-increasing function in x for $\alpha \geq 0, x \geq 0$.

Proof. We compute the derivative of the function as

$$\begin{aligned} \frac{\partial(\partial \psi_\alpha(x)/\partial x)}{\partial x} &= \partial \left(\frac{2\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} \right) / \partial x \\ &= \frac{-2\alpha^2 e^{-\alpha x} (1 - e^{-\alpha x})}{(1 + e^{-\alpha x})^3}, \end{aligned} \quad (55)$$

which is non-positive for $\alpha \geq 0$ and $x \geq 0$, which leads to the desired conclusion. $\square$

Lemma E.5. For the function $\psi_\alpha(x)$ defined in (5), the following holds:

$$\frac{\partial \psi_\alpha(x)}{\partial x} \leq \frac{\psi_\alpha(x)}{x}, \quad \alpha \geq 0, x \geq 0. \quad (56)$$

Proof. We begin with the inequality $\alpha x \leq \sinh \alpha x$ for $x \geq 0$ and $\alpha \geq 0$. We have

$$\begin{aligned} \alpha x \leq \sinh \alpha x &\stackrel{(a)}{\Rightarrow} 2\alpha x e^{-\alpha x} \leq 1 - e^{-2\alpha x} \\ &\stackrel{(b)}{\Rightarrow} \frac{2\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} \leq \frac{1}{x} \left(\frac{2}{1 + e^{-\alpha x}} - 1 \right) \\ &\stackrel{(c)}{\Rightarrow} \frac{\partial \psi_\alpha(x)}{\partial x} \leq \frac{\psi_\alpha(x)}{x}, \end{aligned} \quad (57)$$

where (a) follows from multiplying both sides of the previous inequality by $2e^{-\alpha x}$, (b) follows from dividing both sides by $x(1 + e^{-\alpha x})^2$, and (c) results from writing both sides in terms of $\psi_\alpha(x)$ and $\frac{\partial \psi_\alpha(x)}{\partial x}$. $\square$

Theorem E.6. Let $f(s, c, \gamma_t)$ be the function defined in Lemma E.2. Then, the following properties hold:

1. $f(s, c, \gamma_t)$ is non-increasing in s for $0 \leq c \leq 1$ and $\gamma_t \geq 0$.
2. $f(s, c, \gamma_t) \geq 0$ for $0 \leq c \leq 1$ and $\gamma_t \geq 0$.
3. $f(s, c, \gamma_t)$ is non-increasing in c on the interval $0 \leq c \leq 1$, for $s \geq 1$ and $\gamma_t \geq 0$.

Proof. We first prove the first property, showing that $f(s, c, \gamma_t)$ is non-increasing in s . Based on (44) in Lemma E.2, we rewrite $f(s, c, \gamma_t)$ as

$$f(s, c, \gamma_t) = f_1(s, c) \psi_{\gamma_t}(x_1) + f_2(s, c) \psi_{\gamma_t}(x_2), \quad (58)$$

where we define $x_1 \triangleq \sqrt{1 + s^2 + 2sc}$, $x_2 \triangleq \sqrt{1 + s^2 - 2sc}$, and

$$f_1(s, c) \triangleq \frac{1 + sc}{\sqrt{1 + s^2 + 2sc}}, \quad (59)$$

$$f_2(s, c) \triangleq \frac{1 - sc}{\sqrt{1 + s^2 - 2sc}}. \quad (60)$$

Taking the derivative of $f(s, c, \gamma_t)$ w.r.t. s , we have

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial s} &= \frac{\partial f_1(s, c)}{\partial s} \psi_{\gamma_t}(x_1) + f_1(s, c) \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \frac{\partial x_1}{\partial s} \\ &\quad + \frac{\partial f_2(s, c)}{\partial s} \psi_{\gamma_t}(x_2) + f_2(s, c) \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \frac{\partial x_2}{\partial s}. \end{aligned} \quad (61)$$

Substituting the derivatives into the above expression results in

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial s} = & \frac{-s(1-c^2)}{(1+s^2+2sc)^{\frac{3}{2}}} \psi_{\gamma_t}(x_1) + \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \left(c + \frac{s(1-c^2)}{1+s^2+2sc} \right) \\ & + \frac{-s(1-c^2)}{(1+s^2-2sc)^{\frac{3}{2}}} \psi_{\gamma_t}(x_2) + \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \left(-c + \frac{s(1-c^2)}{1+s^2-2sc} \right). \end{aligned} \quad (62)$$

Rearranging the terms, we have

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial s} = & -c \left(\frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} - \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \right) \\ & - \frac{s(1-c^2)}{1+s^2+2sc} \left(\frac{\psi_{\gamma_t}(x_1)}{x_1} - \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \right) \\ & - \frac{s(1-c^2)}{1+s^2-2sc} \left(\frac{\psi_{\gamma_t}(x_2)}{x_2} - \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \right). \end{aligned} \quad (63)$$

The first term in the RHS of (63) is non-positive since, by Lemma E.4, the derivative of $\psi_{\alpha}(x)$ is a non-increasing function of x , and we have $x_2 \leq x_1$ for $c \geq 0, s \geq 0$. The second and third terms are also non-positive because $c \leq 1$ and, by Lemma E.5, it holds that

$$\frac{\psi_{\gamma_t}(x)}{x} \geq \frac{\partial \psi_{\gamma_t}(x)}{\partial x}, \quad \forall x \geq 0, \forall \gamma_t \geq 0. \quad (64)$$

Thus, we conclude that $\frac{\partial f(s, c, \gamma_t)}{\partial s} \leq 0$, implying that $f(s, c, \gamma_t)$ is non-increasing in s .

To prove the second property, we first evaluate the limit of the function as s approaches infinity:

$$\lim_{s \rightarrow \infty} f(s, c, \gamma_t) = 0, \quad 1 \geq c \geq 0, \gamma_t \geq 0. \quad (65)$$

By the first property, $f(s, c, \gamma_t)$ is non-increasing in s , which implies that its minimum value is zero.

To establish the third property, we compute the derivative of $f(s, c, \gamma_t)$ w.r.t. c . Based on (58), we have

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial c} = & \frac{\partial f_1(s, c)}{\partial c} \psi_{\gamma_t}(x_1) + f_1(s, c) \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \frac{\partial x_1}{\partial c} + \frac{\partial f_2(s, c)}{\partial c} \psi_{\gamma_t}(x_2) \\ & + f_2(s, c) \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \frac{\partial x_2}{\partial c}. \end{aligned} \quad (66)$$

Plugging the derivatives into the above expression and rearranging the terms, we obtain

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial c} = & \frac{\psi_{\gamma_t}(x_1)}{x_1} \frac{s^2(s+c)}{1+s^2+2sc} - \frac{\psi_{\gamma_t}(x_2)}{x_2} \frac{s^2(s-c)}{1+s^2-2sc} \\ & + \frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \frac{s(1+sc)}{1+s^2+2sc} - \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \frac{s(1-sc)}{1+s^2-2sc}. \end{aligned} \quad (67)$$

Next, we find an upper bound for $\frac{\partial f(s, c, \gamma_t)}{\partial c}$. Based on Lemma E.3, $\frac{\psi_{\gamma_t}(x)}{x}$ is a non-increasing function in x . Thus, we have

$$\frac{\psi_{\gamma_t}(x_1)}{x_1} \leq \frac{\psi_{\gamma_t}(x_2)}{x_2}, \quad (68)$$

which follows from the fact that $x_2 \leq x_1$ for $s \geq 0$ and $1 \geq c \geq 0$. Furthermore, by Lemma E.4, the function $\frac{\partial \psi_{\gamma_t}(x)}{\partial x}$ is non-increasing in x . Thus, since $x_2 \leq x_1$, it follows that

$$\frac{\partial \psi_{\gamma_t}(x_1)}{\partial x_1} \leq \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2}. \quad (69)$$

Substituting the upper bounds from (68) and (69) into (67), we obtain

$$\begin{aligned} \frac{\partial f(s, c, \gamma_t)}{\partial c} \leq & \frac{\psi_{\gamma_t}(x_2)}{x_2} \left(\frac{s^2(s+c)}{1+s^2+2sc} - \frac{s^2(s-c)}{1+s^2-2sc} \right) + \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \left(\frac{s(1+sc)}{1+s^2+2sc} - \frac{s(1-sc)}{1+s^2-2sc} \right) \\ = & \frac{\psi_{\gamma_t}(x_2)}{x_2} \frac{2cs^2(1-s^2)}{(1+s^2-2sc)(1+s^2+2sc)} - \frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \frac{2cs^2(1-s^2)}{(1+s^2-2sc)(1+s^2+2sc)}. \end{aligned} \quad (70)$$

The RHS of the upper bound in (70) is non-positive for $s \geq 1$ and $1 \geq c \geq 0$, since $\frac{\partial \psi_{\gamma_t}(x_2)}{\partial x_2} \leq \frac{\psi_{\gamma_t}(x_2)}{x_2}$ by Lemma E.5. $\square$

Lemma E.7. For any $r \geq \|\mathbf{g}_t\|$, the following inequality holds:

$$\mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1 \right] \geq f\left(\frac{r}{\|\mathbf{g}_t\|}, c = 1, \gamma_t\right) \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\}. \quad (71)$$

Proof. For any $r \in \mathbb{R}$, we have

$$\begin{aligned} \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1 \right] &\stackrel{(a)}{\geq} \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1, s \leq \frac{r}{\|\mathbf{g}_t\|} \right] \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\} \\ &\quad + \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1, s \geq \frac{r}{\|\mathbf{g}_t\|} \right] \Pr \left\{ s \geq \frac{r}{\|\mathbf{g}_t\|} \right\} \end{aligned} \quad (72)$$

$$\stackrel{(b)}{\geq} \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1, s \leq \frac{r}{\|\mathbf{g}_t\|} \right] \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\} \quad (73)$$

$$\stackrel{(c)}{\geq} \mathbb{E}_c \left[f\left(\frac{r}{\|\mathbf{g}_t\|}, c, \gamma_t\right) \middle| 0 \leq c \leq 1 \right] \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\}, \quad (74)$$

where (a) follows from conditioning the expectation on the events $s \geq \frac{r}{\|\mathbf{g}_t\|}$ and $s \leq \frac{r}{\|\mathbf{g}_t\|}$, (b) is concluded by dropping the second term in (72) and noting that $f(s, c, \gamma_t) \geq 0$ based on Theorem E.6, and (c) follows from the fact that $f(s, c, \gamma_t)$ is non-increasing in s for any $1 \geq c \geq 0$ and $\gamma_t \geq 0$ due to Theorem E.6.

We now restrict $r \geq \|\mathbf{g}_t\|$ and use the fact from Theorem E.6 that $f(s, c, \gamma_t)$ is non-increasing in c for any $s \geq 1$ to conclude that for any $r \geq \|\mathbf{g}_t\|$,

$$\mathbb{E}_c \left[f\left(\frac{r}{\|\mathbf{g}_t\|}, c, \gamma_t\right) \middle| 0 \leq c \leq 1 \right] \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\} \geq f\left(\frac{r}{\|\mathbf{g}_t\|}, c = 1, \gamma_t\right) \Pr \left\{ s \leq \frac{r}{\|\mathbf{g}_t\|} \right\}, \quad (75)$$

which completes the proof. $\square$

Lemma E.8. For $s \geq 1$, $f(s, c = 1, \gamma_t)$ is lower bounded as

$$f(s, c = 1, \gamma_t) \geq \frac{\gamma_t}{\cosh(s\gamma_t)}. \quad (76)$$

Proof. By substituting $c = 1$ into the definition of $f(s, c, \gamma_t)$ given in Lemma E.2 and simplifying the expression, we obtain

$$\begin{aligned} f(s, c = 1, \gamma_t) &= \frac{1+s}{\sqrt{1+s^2+2s}} \left(\frac{2}{1+e^{-\gamma_t\sqrt{1+s^2+2s}}} - 1 \right) + \frac{1-s}{\sqrt{1+s^2-2s}} \left(\frac{2}{1+e^{-\gamma_t\sqrt{1+s^2-2s}}} - 1 \right) \end{aligned} \quad (77)$$

$$= \frac{2}{1+e^{-\gamma_t(s+1)}} - \frac{2}{1+e^{-\gamma_t(s-1)}} \quad (78)$$

$$= \frac{2\sinh(\gamma_t)}{\cosh(\gamma_t) + \cosh(s\gamma_t)}. \quad (79)$$

We next note that $s \geq 1$. Using the fact that $\cosh(x)$ is an increasing function on $x \geq 0$, we can conclude that $\cosh(s\gamma_t) \geq \cosh(\gamma_t)$. We then use the lower bound $\sinh(x) \geq x$ for $x \geq 0$ to write

$$\frac{2\sinh(\gamma_t)}{\cosh(\gamma_t) + \cosh(s\gamma_t)} \geq \frac{\gamma_t}{\cosh(s\gamma_t)}, \quad (80)$$

which completes the proof. $\square$

Lemma E.9. Let $\lambda_\alpha = k_1/T$ and $\alpha_0 = k_2/r$ for some $k_1 > 0$ and $k_2 > 0$. Under Algorithm 1, the saturation slope is bounded on both sides as

$$\frac{\kappa_1}{r} \leq \alpha_t \leq \frac{\kappa_2}{r}, \quad 0 \leq t \leq T-1, \quad (81)$$

where $\kappa_1 \triangleq k_2 e^{-k_1}$ and $\kappa_2 \triangleq k_2 e^{k_1}$.

Proof. First, observe that for $t = 0$ we have $\alpha_0 = \frac{k_2}{r}$, which satisfies

$$\frac{\kappa_1}{r} \leq \alpha_0 \leq \frac{\kappa_2}{r},$$

since $k_1 > 0$. We next show that the same bound also holds for $1 \leq t \leq T-1$. Substituting $\lambda_\alpha = \frac{k_1}{T}$ and $\alpha_0 = \frac{k_2}{r}$ into the update rule (16), we obtain for any $t \geq 1$,

$$\alpha_t = \frac{k_2}{r} e^{\frac{k_1 q_t}{T}}, \quad (82)$$

where q_t is defined as

$$q_t \triangleq \sum_{\tau=0}^{t-1} \text{sign}(\hat{s}(\boldsymbol{\theta}_\tau, \alpha_\tau; \mathcal{B}_\tau)^\top \hat{r}(\boldsymbol{\theta}_{\tau-1}, \alpha_{\tau-1}; \mathcal{B}_{\tau-1})), \quad (83)$$

with $\hat{r}(\boldsymbol{\theta}_{-1}, \alpha_{-1}; \mathcal{B}_{-1}) = \mathbf{0}$. Since $-t \leq q_t \leq t$ and $1 \leq t \leq T-1$, we have

$$\alpha_t \geq \frac{k_2}{r} e^{\frac{-k_1(T-1)}{T}} \geq \frac{k_2}{r} e^{-k_1} = \frac{\kappa_1}{r} \quad (84)$$

$$\alpha_t \leq \frac{k_2}{r} e^{\frac{k_1(T-1)}{T}} \leq \frac{k_2}{r} e^{k_1} = \frac{\kappa_2}{r}, \quad (85)$$

which completes the proof. $\square$

Theorem E.10. *Let Assumptions 7.1–7.4 hold, and suppose that $\lambda_\alpha \propto 1/T$ and $\alpha_0 \propto 1/r$ for some constant $r \geq G$. Then, under the DP-SGD-AdaSig algorithm described in Algorithm 1, the following inequality holds:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \Pr \left\{ \|\Delta_t\| \leq r \right\} \mathbb{E} \|\mathbf{g}_t\|^2 \leq r \tilde{G} \left(\frac{L(\boldsymbol{\theta}_0) - L^*}{\lambda T} + \beta \lambda \left(1 + \frac{d\sigma_s^2}{2B^2} \right) \right), \quad (86)$$

where $\tilde{G} > 0$ is a constant.

Proof. Using the results of Lemmas E.1, E.2, E.7, E.8, and E.9, we have

$$\begin{aligned} & \frac{1}{\lambda} \mathbb{E} \left[L(\boldsymbol{\theta}_t) - L(\boldsymbol{\theta}_{t+1}) \middle| \boldsymbol{\theta}_t, \alpha_t \right] + \beta \lambda \left(1 + \frac{d\sigma_s^2}{2B^2} \right) \\ & \stackrel{\text{Lemma E.1}}{\geq} \mathbf{g}_t^\top \mathbb{E} \left[\left(\frac{2}{1 + e^{-\alpha_t \|\mathbf{v}_t\|}} - 1 \right) \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|} \middle| \boldsymbol{\theta}_t, \alpha_t \right] \end{aligned} \quad (87)$$

$$\stackrel{\text{Lemma E.2}}{=} \frac{\|\mathbf{g}_t\|}{2} \mathbb{E}_{s,c} \left[f(s, c, \gamma_t) \middle| 0 \leq c \leq 1 \right] \quad (88)$$

$$\stackrel{(a)}{\geq} \frac{\|\mathbf{g}_t\|}{2} f\left(\frac{r}{\|\mathbf{g}_t\|}, c = 1, \gamma_t\right) \Pr \left\{ \|\Delta_t\| \leq r \right\} \quad (89)$$

$$\stackrel{(b)}{\geq} \frac{\alpha_t \|\mathbf{g}_t\|^2}{2 \cosh(r\alpha_t)} \Pr \left\{ \|\Delta_t\| \leq r \right\} \quad (90)$$

$$\stackrel{\text{Lemma E.9}}{\geq} \frac{\kappa_1 \|\mathbf{g}_t\|^2}{2r \cosh(\kappa_2)} \Pr \left\{ \|\Delta_t\| \leq r \right\}, \quad (91)$$

where (a) is obtained by applying Lemma E.7, which requires the bound $r \geq \|\mathbf{g}_t\|, \forall t$. This bound follows directly from Assumption 7.3 (i.e., $\|\mathbf{g}_t\| \leq G, \forall t$) together with the assumption $r \geq G$. Inequality (b) is obtained by applying Lemma E.8, using the bound $r \geq \|\mathbf{g}_t\|, \forall t$. The last inequality follows from Lemma E.9, since under the assumptions $\lambda_\alpha \propto 1/T$ and $\alpha_0 \propto 1/r$, we can equivalently write $\lambda_\alpha = k_1/T$ and $\alpha_0 = k_2/r$ for some constants $k_1 > 0$ and $k_2 > 0$. Substituting these expressions into the lemma, the inequality holds with $\kappa_1 = k_2 e^{-k_1}, \kappa_2 = k_2 e^{k_1}$.

Let us now define $\tilde{G} \triangleq 2 \cosh(\kappa_2)/\kappa_1$. After taking expectation from both sides of the last inequality, summing over iterations from 0 to $T-1$, and dividing by T , we have

$$\begin{aligned} \frac{1}{r\tilde{G}T} \sum_{t=0}^{T-1} \Pr\{\|\Delta_t\| \leq r\} \mathbb{E}\|\mathbf{g}_t\|^2 \\ \leq \frac{1}{\lambda T} \sum_{t=0}^{T-1} \mathbb{E}[L(\boldsymbol{\theta}_t) - L(\boldsymbol{\theta}_{t+1})] + \frac{1}{T} \sum_{t=0}^{T-1} \beta\lambda \left(1 + \frac{d\sigma_s^2}{2B^2}\right) \end{aligned} \quad (92)$$

$$\stackrel{(a)}{\leq} \frac{L(\boldsymbol{\theta}_0) - L^*}{\lambda T} + \beta\lambda \left(1 + \frac{d\sigma_s^2}{2B^2}\right), \quad (93)$$

where (a) follows from Assumption 7.1. Rearranging the terms yields the inequality in (86). $\square$

E.2 Proof of Theorem 7.5

Proof. We use the result in Theorem E.10 and set the noise multiplier σ_s such that privacy is guaranteed. To ensure the privacy guarantee, we use the result in Theorem 1 in [1]. For clarity, we first restate Theorem 1 in [1].

Theorem E.11 (Theorem 1 of [1]). *There exist constants u and ν such that, given the sampling probability $q = B/N$, for any $\epsilon \leq uq^2T$, the composition of T Gaussian mechanisms, each with noise multiplier σ , satisfies (ϵ, δ) -DP if*

$$\sigma^2 \geq \frac{\nu^2 q^2 T \log(1/\delta)}{\epsilon^2}. \quad (94)$$

According to Theorem E.11, the composition of T Gaussian mechanisms ensures (ϵ, δ) -DP provided that the noise multiplier σ is set to

$$\sigma^2 = \frac{\nu^2 B^2 T \log(1/\delta)}{N^2 \epsilon^2}. \quad (95)$$

Note that our proposed algorithm (DP-SGD-AdaSig, described in Algorithm 1) requires two noise multipliers, σ_s and σ_r , since each iteration involves two Gaussian mechanisms, $\hat{s}(\cdot)$ and $\hat{r}(\cdot)$. According to Proposition 6.1, these two parallel Gaussian mechanisms are equivalent to a single Gaussian mechanism with noise multiplier σ , where

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_s^2} + \frac{1}{\sigma_r^2}. \quad (96)$$

To ensure (ϵ, δ) -DP for the overall algorithm after T iterations, we set σ_s and σ_r such that the equivalent noise multiplier σ in (96) satisfies (95). This can be achieved by setting σ_s and σ_r as follows:

$$\sigma_s^2 = \frac{\nu_s^2 B^2 T \log(1/\delta)}{N^2 \epsilon^2}, \quad (97)$$

$$\sigma_r^2 = \frac{\nu_r^2 B^2 T \log(1/\delta)}{N^2 \epsilon^2}, \quad (98)$$

with constants ν_s and ν_r satisfying

$$\frac{1}{\nu^2} = \frac{1}{\nu_s^2} + \frac{1}{\nu_r^2}. \quad (99)$$

The choices of σ_s and σ_r in (97) and (98), together with constants ν_s and ν_r that satisfy the equality in (99), ensure that (96) holds and thereby guarantee (ϵ, δ) -DP for the DP-SGD-AdaSig algorithm.

Substituting σ_s^2 from (97) into (86) in Theorem E.10, we then optimize its LHS w.r.t. the learning rate λ , which yields

$$\lambda = \sqrt{\frac{2N^2 \epsilon^2 (L(\boldsymbol{\theta}_0) - L^*)}{\beta T (2N^2 \epsilon^2 + d\nu_s^2 T \log(\frac{1}{\delta}))}}. \quad (100)$$

Substituting this learning rate into the LHS concludes the result. $\square$

E.3 Proof of Corollary 7.7

Proof. The probability $\Pr \left\{ \|\Delta_t\| \leq r \right\}$ can be lower bounded as

$$\Pr \left\{ \|\Delta_t\| \leq r \right\} \stackrel{(a)}{\geq} 1 - \frac{\mathbb{E} \|\Delta_t\|}{r} \quad (101)$$

$$\stackrel{(b)}{\geq} 1 - \frac{\sqrt{\mathbb{E} \|\Delta_t\|^2}}{r} \quad (102)$$

$$\stackrel{(c)}{\geq} 1 - \frac{\zeta}{r}, \quad (103)$$

where (a) follows from Markov's inequality, (b) follows from Jensen's inequality, and (c) is due to the bounded variance assumption.

Using the derived probability lower bound in (103) on the LHS of (18) in Theorem 7.5, and considering $r \geq \max\{\zeta, G\}$, we divide both sides by $1 - \frac{\zeta}{r}$ to obtain:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\mathbf{g}_t\|^2 \leq \frac{2\tilde{G}r^2}{r - \zeta} \sqrt{\frac{\beta(L(\theta_0) - L^*)}{T} \left(1 + \frac{d\nu_s^2 T \log(\frac{1}{\delta})}{2N^2 \epsilon^2}\right)}. \quad (104)$$

We next set r such that the RHS of the above inequality is minimized. Note that when $r \geq \zeta$, the function $\frac{r^2}{r - \zeta}$ achieves its minimum value at $r = 2\zeta$, and for $r \geq 2\zeta$, the function is increasing. Noting that in Theorem 7.5 $r \geq G$, two cases may occur: namely, $G \leq 2\zeta$ or $G \geq 2\zeta$. By considering these two cases, separately, we can conclude that

$$r_* = \underset{r \geq \max\{\zeta, G\}}{\operatorname{argmin}} \frac{r^2}{r - \zeta} = \begin{cases} 2\zeta, & \text{if } G \leq 2\zeta, \\ G, & \text{if } G \geq 2\zeta \end{cases} = \max\{2\zeta, G\}. \quad (105)$$

Substituting r_* into (104) completes the proof. $\square$

F Experiments Settings

This section provides further information on the experimental settings considered in the paper. We present more details on the datasets and architectures used for the numerical experiments presented in the paper.

F.1 General Settings

Privacy Settings In all experiments, we fix the DP budget (ϵ, δ) , and compute the noise multiplier, i.e., σ , numerically using the Opcaus library [40], such that the DP budget spent after T iterations (or equivalently Tq epochs) equals to the fixed budget (ϵ, δ) . Considering Proposition 6.1, for a given σ , different pairs of σ_s and σ_r can be obtained, where decreasing one increases the other. To simplify the parameter space, we set $\sigma_s = 1.01\sigma$ in all AdaSig experiments, making σ_s smaller than σ_r . This choice prioritizes a more accurate update of the model over the saturation slope, as roughly updating α in the descent direction is sufficient.

Hyperparameters To ensure fair comparison, we use the same batch size and number of epochs across all methods. For AdaSig, we tune the learning rates λ , λ_α , and the initial saturation slope α_0 . For vanilla clipping, we tune C and λ . For Auto-S and PSAC, we use the tuned value of C for vanilla clipping following their original papers [4, 36], and tune hyperparameters r and learning rate λ . For the method in [3], we tune the initial clipping threshold C^0 , clipping threshold learning rate η_C , learning rate λ , σ_b , and γ .

F.2 Settings for Image Classification

MNIST and FashionMNIST We use the 4-layer CNN with tanh activation proposed in [27] and described in Table 6 of [34], with cross-entropy loss and the DP-SGD optimizer. The DP budget is set to $(\epsilon, \delta) = (3, 10^{-5})$. For batch size, number of epochs, and momentum, which are common across different methods, we use the values reported in [34]. These values are summarized in Table 2. Table 3 presents the best values of other hyperparameters for each method.

Table 2: Common hyperparameters across different methods used for training CNN on MNIST, FashionMNIST, and CIFAR-10.

Parameter	MNIST	FashionMNIST	CIFAR-10
Batch size (B)	512	2048	4096
Number of epochs (Tq)	40	40	60
Momentum	0.9	0.9	0.9

Table 3: Hyperparameters selected for each method for training CNN on MNIST, FashionMNIST, and CIFAR-10.

Method	Parameter	MNIST	FashionMNIST	CIFAR-10
Vanilla	C	0.1	0.1	0.1
	λ	0.5	4.0	3.0
Auto-S	r	0.01	0.01	0.01
	C	0.1	0.1	0.1
	λ	0.5	4.0	3.0
PSAC	r	0.1	0.1	0.1
	C	0.1	0.1	0.1
	λ	0.5	4.0	3.0
Method in [3]	C^0	0.1	0.1	0.1
	λ	0.5	4.0	3.0
	η_C	0.05	0.01	0.01
	σ_b	40.0	30.0	25.0
	γ	0.5	0.5	1.0
AdaSig	α_0	5.0	1.0	1.0
	λ	0.05	0.4	0.25
	λ_α	0.01	0.01	0.01

CIFAR-10 We use the 8-layer CNN with tanh activation from [27], as detailed in Table 7 of [34], with cross-entropy loss and DP-SGD optimizer. The DP budget is set to $(\epsilon, \delta) = (3, 10^{-5})$. We use the same batch size, number of epochs, and momentum across different methods as reported in Table 2.⁴ The best values of other hyperparameters for different methods are reported in Table 3.

ImageNette We use ImageNette, a 10-class subset of ImageNet [9], with an image size of 160×160 . We consider ResNet-9 architecture (about 2.5 million parameters), with the Mish activation function [25] and cross-entropy loss. For training, the DP-Nesterov-accelerated Adam (DP-NAdam) optimizer is utilized.⁵ We consider a DP budget of $(\epsilon, \delta) = (8, 10^{-4})$. We follow [17] by using group normalization instead of batch normalization without scale normalization. The only difference is that we do not apply the learning rate decay schedule. The ResNet-9 architecture can be found at [17].⁶ All methods use the same batch size and number of epochs, which are set according to the values reported in [4, 36], as shown in Table 4. Table 5 lists the best values of the remaining hyperparameters for each method.

CelebA We use the same ResNet-9 architecture as for the ImageNette dataset, with the DP-Adam optimizer. The CelebA dataset contains 40 labels per image and we use it for both single-label and multi-label classification tasks.

⁴Note that the batch size and the number of epochs differ from those reported in [34] and used in [4, 36]. Our experiments showed that these hyperparameter changes yield better performance across all baselines compared with those reported in the references.

⁵For ImageNette and CelebA, we use more advanced optimizers than DP-SGD, namely, DP-NAdam and DP-Adam, respectively, to achieve improved performance for these more challenging datasets. These experiments further demonstrate the wide applicability of the AdaSig approach with different optimizers.

⁶Check also <https://gist.github.com/gkaissis/6db6b7271f93d3459263b6978cfd4146>.

Table 4: Common hyperparameters across different methods used for training Resnet-9 on ImageNette and CelebA.

Parameter	ImageNette	CelebA (Multi-label/Smiling)
Batch size (B)	1024	512
Number of epochs (Tq)	50	10

Table 5: Hyperparameters selected for each method for training ResNet-9 on ImageNette, and CelebA.

Method	Parameter	ImageNette	CelebA(Multi-label)	CelebA (Smiling)
Vanilla	C	1.5	0.1	0.1
	λ	5×10^{-4}	10^{-3}	10^{-3}
Auto-S	r	0.01	0.01	0.01
	C	1.5	0.1	0.1
	λ	5×10^{-4}	10^{-3}	10^{-3}
PSAC	r	0.1	0.1	0.1
	C	1.5	0.1	0.1
	λ	5×10^{-4}	10^{-3}	10^{-3}
Method in [3]	C^0	1.5	0.1	0.1
	λ	5×10^{-4}	5×10^{-4}	2×10^{-3}
	η_C	0.01	0.01	0.05
	σ_b	30.0	20.0	25.0
	γ	3.0	1.0	1.0
AdaSig	α_0	5.0	1.0	1.0
	λ	10^{-3}	5×10^{-4}	5×10^{-4}
	λ_α	0.02	0.001	0.02

- For the single-label task, we perform binary classification considering the label 'Smiling', where we use the binary cross-entropy loss for training. In this scenario, the output layer of ResNet-9 consists of a single neuron.
- For the multi-label classification task, all available 40 labels are considered for prediction. Here, the output layer contains 40 neurons, and we use a scalar loss function that averages the 40 binary cross-entropy losses from each label.

In both cases, we set the DP budget to $(\epsilon, \delta) = (8, 5 \times 10^{-6})$.

The batch size and the number of epochs are set the same across all methods, following the settings in [4, 36], and are reported in Table 4. The best values of other hyperparameters for each method are presented in Table 5.

F.3 Setting for Sentence Classification

SST-2 and QNLI We use a pre-trained RoBERTa-base model (about 125 million parameters) [22] and perform full parameter fine-tuning. We use the cross-entropy loss function. The AdamW optimizer [24] without weight decay is applied. A learning rate scheduler is used to linearly reduce the learning rate from its initial value to zero throughout the training process. The DP budget is set to $(\epsilon, \delta) = (3, \frac{1}{2N})$, where N is the number of training samples for each dataset. It is worth mentioning that SST-2 and QNLI contain 67,349 and 104,743 training samples, respectively.

The batch size, number of epochs, and maximum sequence length for all methods are set according to [21, 4], and are given in Table 6. Table 7 lists the best value of other hyperparameters for each method.

Table 6: Common hyperparameters across different methods used for RoBERTa-base full parameter fine-tuning on SST-2 and QNLI.

Parameter	SST-2	QNLI
Batch size (B)	1000	2000
Number of epochs (Tq)	3	6
Max sequence length	256	256

Table 7: Hyperparameters selected for each method for fine-tuning RoBERTAa-base on SST-2 and QNLI.

Method	Parameter	SST-2	QNLI
Vanilla	C	0.1	0.1
	λ	5×10^{-4}	5×10^{-4}
Auto-S	r	0.01	0.01
	C	0.1	0.1
	λ	5×10^{-4}	5×10^{-4}
PSAC	r	0.1	0.1
	C	0.1	0.1
	λ	5×10^{-4}	5×10^{-4}
Method in [3]	C^0	0.1	0.1
	λ	5×10^{-4}	5×10^{-4}
	η_C	0.01	0.01
	σ_b	35.00	25.0
	γ	1.0	1.0
AdaSig	α_0	1.0	1.0
	λ	5×10^{-4}	5×10^{-4}
	λ_α	0.005	0.01

F.4 Hardware and Software Information

All experiments are performed on a server equipped with Intel Xeon E5-2683 v4 CPUs, NVIDIA V100 GPUs, and 251 GiB of memory. The operating system used is AlmaLinux 9.3, and the CUDA Toolkit version is 12.2. The implementation of all training procedures is based on PyTorch 2.3.0 and Opacus 1.4.1.

G Numerical Analysis of Direction Deviation and Magnitude Deviation

In this section, we provide additional figures to further assess the direction deviation and magnitude deviation achieved by the proposed AdaSig clipping method and the baselines.

G.1 Defining Metrics

Cosine Similarity To evaluate the direction deviation, we compute the cosine similarity between the aggregation of the clipped gradients, i.e., $\tilde{\mathbf{g}}_t \triangleq \sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_{i,t}$, and the true batch gradient, i.e., $\mathbf{g}_t \triangleq \sum_{i \in \mathcal{B}_t} \mathbf{g}_{i,t}$, as $\cos \phi_t = \frac{\langle \tilde{\mathbf{g}}_t, \mathbf{g}_t \rangle}{\|\tilde{\mathbf{g}}_t\| \cdot \|\mathbf{g}_t\|}$, which measures the cosine of the angle between $\tilde{\mathbf{g}}_t$ and $\mathbf{g}_t$.

SNR To characterize the magnitude deviation, we define the signal-to-noise-ratio (SNR) as a normalized magnitude of the aggregation of the clipped gradients. Let us denote by $\|\tilde{\mathbf{g}}_t\|$ the magnitude of the aggregation of the clipped gradients after clipping, i.e., $\|\tilde{\mathbf{g}}_t\| = \left\| \sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_{i,t} \right\|$. We define the SNR to be the ratio of this magnitude to the standard deviation of the added privacy noise denoted by σ_n , i.e., $\text{SNR} = \frac{\|\tilde{\mathbf{g}}_t\|}{\sigma_n}$. For AdaSig, $\sigma_n = \sigma_s$. For vanilla clipping, Auto-S, and PSAC,

$\sigma_n = C\sigma$. For the method in [3], $\sigma_n = C^t z_\Delta$. This metric can describe the magnitude deviation against various approaches with potentially different privacy noise variances: as the SNR decreases, the performance degradation caused by the privacy noise becomes more severe.

G.2 Performance Comparison

ImageNette Figure 4 shows the histogram of the SNR during training on the ImageNette dataset. The results depict that using AdaSig, the SNR is concentrated around larger values. We further recall the earlier observations reported in Figure 3 in the main paper. There, we observe that using AdaSig, direction deviation is reduced compared with the baselines. Considering these two results, i.e., Figure 3 and Figure 4, we conclude that using AdaSig, the deteriorating impact of clipping is reduced with respect to *both* direction deviation and magnitude deviation. This demonstrates that, for ImageNette, AdaSig provides highly effective clipping that preserves closeness to the true batch gradient.

CIFAR-10 We next present the results of similar experiments on CIFAR-10. Figures 5 and 6 show the histogram of cosine similarity and SNR while training the 8-layer CNN on CIFAR-10, respectively. As observed in these figures, the cosine similarity distribution of AdaSig is skewed toward higher values relative to the baselines. This means that AdaSig incurs less direction deviation. Nevertheless, the histogram of the SNR achieved by AdaSig is close to those of the baselines. Thus, for CIFAR-10, AdaSig clipping is still able to improve the trade-off between the direction deviation and the magnitude deviation.

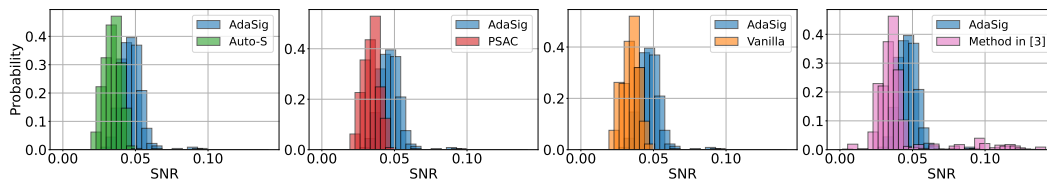


Figure 4: Comparison of normalized magnitude for different approaches during training ResNet-9 on ImageNette.

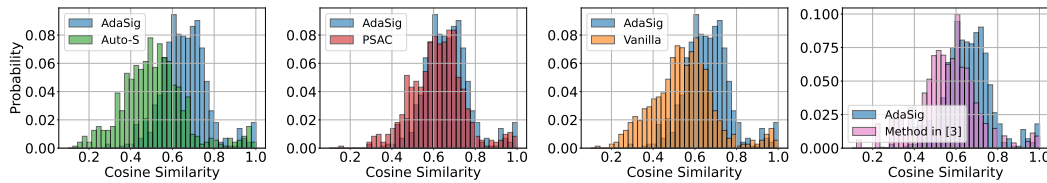


Figure 5: Comparison of cosine similarity for different approaches during training 8-layer CNN on CIFAR-10.

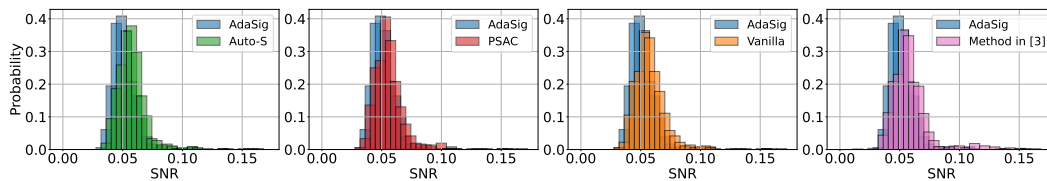


Figure 6: Comparison of normalized magnitude for different approaches during training 8-layer CNN on CIFAR-10.

H Ablation Study: Sigmoid Clipping with Constant α

An ablation method for the AdaSig approach is to use a fixed α across all iterations, i.e., $\alpha_t = \alpha, \forall t$. We examine sigmoid clipping with various fixed α values for training an 8-layer CNN on the CIFAR-10 dataset. The experimental setup is based on Appendix F.2.

H.1 Impact of α on Direction Deviation and Magnitude Deviation

In this section, we study the impact of α on the *trade-off* between direction deviation and magnitude deviation. Figure 7 shows the histogram of cosine similarity and SNR during the training for three fixed values of α . As seen, decreasing the α value results in an increase in cosine similarity, while reducing the SNR. This occurs because decreasing α expands the linear region of the sigmoid function, providing equal scaling for different gradient samples during clipping (see Figure 1 in the main paper), which reduces the direction deviation. This observation also aligns with the numerical example presented in Section 4 of the main paper.

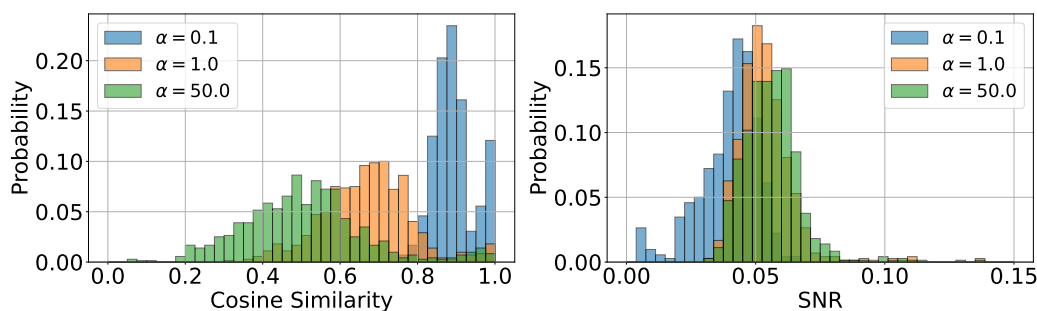


Figure 7: Trade-off between direction deviation and magnitude deviation for several fixed α values during the training of an 8-layer CNN on CIFAR-10. left: Cosine similarity histogram, right: SNR histogram.

H.2 Impact of α on Test Accuracy

Next, we present the average final test accuracies for sigmoid clipping with several fixed α values in Table 8. As observed, very low values of α (less than 0.1) result in low accuracy due to a low SNR, while very large α values (greater than 5) also degrade accuracy due to direction deviation. However, for α in the middle range, we achieve the highest accuracy, resulting from a balance between direction deviation and magnitude deviation. It is worth noting that the highest accuracy achieved by $\alpha = 1$ is still lower than the accuracy attained by the AdaSig method, as shown in Table 1 in the main paper, which highlights that adaptively adjusting α over iterations further improves accuracy.

Table 8: Average test accuracies with 95% confidence intervals over five runs for sigmoid clipping with various fixed α values.

	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 5.0$	$\alpha = 50.0$	$\alpha = 100.0$
Average test accuracy (%)	45.27 ± 0.67	61.74 ± 0.21	62.13 ± 0.12	61.64 ± 0.27	60.62 ± 0.38	60.39 ± 0.31

I Variation of α_t During Training

In this section, we illustrate how α_t evolves during training on CIFAR-10 for three different random seeds. As shown in Figure 8, the general trend across seeds is that α_t initially increases from its starting value 1.0 up to a peak in the middle of training, and then gradually decreases toward the end. The exact trajectory varies across seeds since each seed corresponds to a different batch sampling process, leading to distinct gradient statistics in each run.

When examining α_t over training iterations across different datasets and models, we do not observe a consistent pattern. This behavior arises from the dependence of α_t on individual sample gradients, which vary substantially across datasets and model architectures.

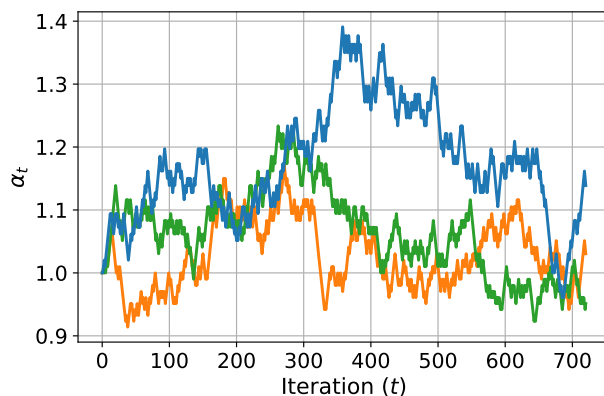


Figure 8: Evolution of α_t across training iterations on CIFAR-10 for three random seeds.

J Convergence Plots

In this section, we present convergence plots for several experiments. Figure 9 illustrates the average test accuracy and test loss over five runs versus epochs during training on the CIFAR-10 dataset. The shaded regions around the curves represent the 95% confidence intervals. As shown, the increasing test accuracy in the left plot and the decreasing test loss in the right plot demonstrate convergence for all methods, including AdaSig. Additionally, the figure highlights that AdaSig achieves a lower final test loss and higher final test accuracy compared with the baselines.

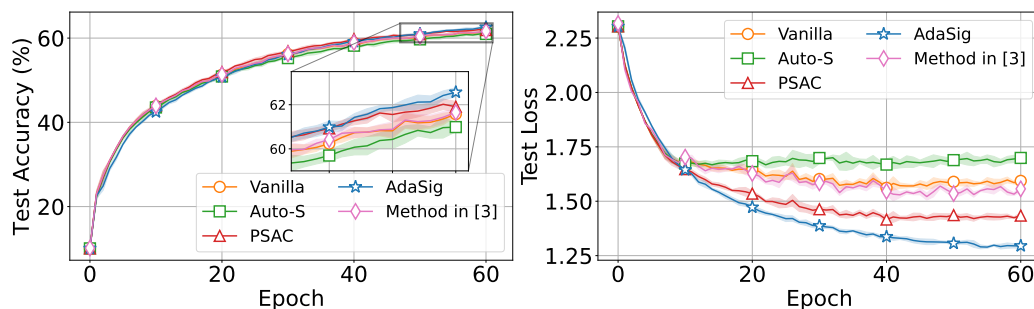


Figure 9: Convergence plot for CIFAR-10 dataset, left: Test accuracy vs. epoch, right: Test loss vs. epoch.

Additionally, Figure 10 shows the average test loss and accuracy throughout the training on the SST-2 dataset. As observed, all methods progressively reduce the test loss while increasing accuracy, with AdaSig outperforming the other methods.

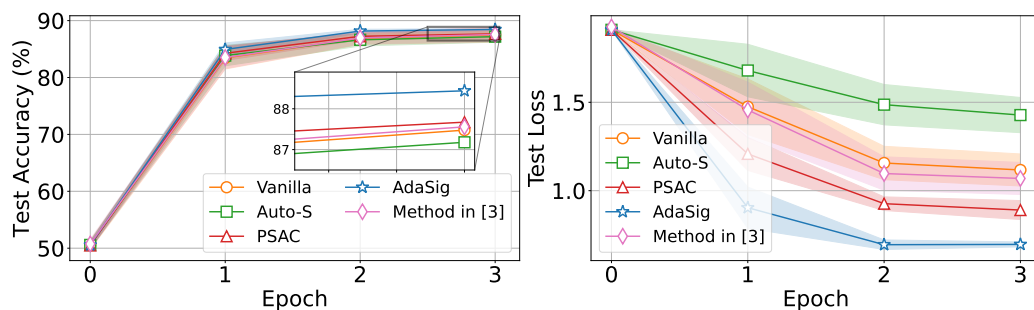


Figure 10: Convergence plot for SST-2 dataset, left: Test accuracy vs. epoch, right: Test loss vs. epoch.