# Differentially Private Learning with Per-Sample Adaptive Clipping

**Tianyu Xia[4], Shuheng Shen[5], Su Yao[1,*], Xinyi Fu[5], Ke Xu[2,3,*], Xiaolong Xu[5], Xing Fu[5]**

[1]Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
[2]Department of Computer Science & Technology, Tsinghua University
[3]Zhongguancun Laboratory, Beijing
[4]School of Software & Microelectronics, Peking University
[5]Tiansuan Lab, Ant Group
xiatainyu@stu.pku.edu.cn, {yaosu, xuke}@tsinghua.edu.cn, {shuheng.ssh, fxy122992, yiyin.xxl, zicai.fx }@antgroup.com

## Abstract

Privacy in AI remains a topic that draws attention from researchers and the general public in recent years. As one way to implement privacy-preserving AI, differentially private learning is a framework that enables AI models to use differential privacy (DP). To achieve DP in the learning process, existing algorithms typically limit the magnitude of gradients with a constant clipping, which requires carefully tuned due to its significant impact on model performance. As a solution to this issue, latest works NSGD and Auto-S innovatively propose to use normalization instead of clipping to avoid hyperparameter tuning. However, normalization-based approaches like NSGD and Auto-S rely on a monotonic weight function, which imposes excessive weight on small gradient samples and introduces extra deviation to the update. In this paper, we propose a Differentially Private Per-Sample Adaptive Clipping (DP-PSAC) algorithm based on a non-monotonic adaptive weight function, which guarantees privacy without the typical hyperparameter tuning process of using a constant clipping while significantly reducing the deviation between the update and true batch-averaged gradient. We provide a rigorous theoretical convergence analysis and show that with convergence rate at the same order, the proposed algorithm achieves a lower non-vanishing bound, which is maintained over training iterations, compared with NSGD/Auto-S. In addition, through extensive experimental evaluation, we show that DP-PSAC outperforms or matches the state-of-the-art methods on multiple main-stream vision and language tasks.

## Introduction

Machine learning has substantially benefited from deep learning research and implementation. Unfortunately, the success of deep neural networks depends on a substantial amount of high-quality data, much of which typically contain sensitive personal data, making data-driven deep models vulnerable to privacy leaks (Zhu, Liu, and Han 2019). DP (Dwork, Roth et al. 2014) formally defines the influence of an individual sample on the final result and provides rigorous theoretical guarantees. Differentially Private stochastic gradient descent (DP-SGD) (Abadi et al. 2016), which first clips each stochastic gradient $g_t$ with a predetermined constant $C$ to constrain the privacy sensitivity and then adds Gaussian noise to the gradients to perturb the result, is a popularly used algorithm to defend deep learning models from differential attacks. Specifically, the iteration of DP-SGD at $x_t$ is:

$$x_{t+1} = x_t - \frac{\eta_t}{|B_t|}\left(\sum_{i\in B_t} g_{t,i}\min\left(\frac{C}{\|g_{t,i}\|},1\right) + \mathcal{N}(0, C^2\sigma^2)\right),$$

where $\eta_t$ is the learning rate, $B_t$ is the random batch and $\sigma$ is the standard deviation of Gaussian noise. Despite its considerable success, DP-SGD with constant clipping suffers from the following issues:

- The performance of the final model, as Kurakin et al. (2022) noted, will be significantly impacted by an incorrect $C$. It is really challenging to tune $C$.
- The search for $C$ itself incurs a extra privacy budget (Papernot and Steinke 2021).

In order to obtain an optimal clipping threshold to achieve higher model accuracy, Andrew et al. (2021) estimated the optimal clipping threshold through gradient quantiles, but this introduces a bigger hyperparameter search space and a large amount of extra computation. By using a public dataset sampled from the private dataset or partial statistics of the private dataset, Zhang, Ji, and Wang (2018) estimated the optimal clipping threshold during the learning process, but this may lead to new privacy leaking problems.

To solve the aforementioned problems, two concurrent research (Bu et al. 2022; Yang et al. 2022) proposed to replace the clipping threshold with automatic clipping/normalizing, i.e. $\tilde{g} = g/(\|g\| + r)$, which can constrain the privacy sensitivity by normalizing all per-sample gradients to the same magnitude, but it actually assigns different weights to samples with various gradient norms. Consequently, the batch gradient becomes a weighted average of the per-sample gradients and the weighted gain is $1/(\|g\|+r)$, meaning smaller gradients are given larger weight. As shown in Figure 2, these techniques will increase the sample's weighted gain by up to $1/r$ times when its gradient norm moves toward 0, where $r$ is often set to 0.1 or a smaller value (Bu et al. 2022). Unfortunately, as illustrated in Figure 1, we observe that in the iterative process, small gradient samples frequently have a tendency to be practically orthogonal or even opposite to the true batch-averaged gradient. This means that the contribution of small gradient samples to the true batch gradient is negligible. Thus, giving small gradient samples large weight

[*]Su Yao and Ke Xu are co-corresponding authors.

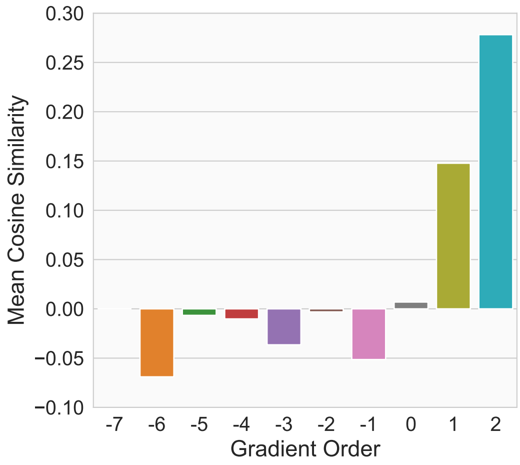

Figure 1: Average cosine similarity of single sample gradient and the batch-averaged gradient throughout training on MNIST dataset with DP-SGD under $(3, 10^{-5})$-DP.

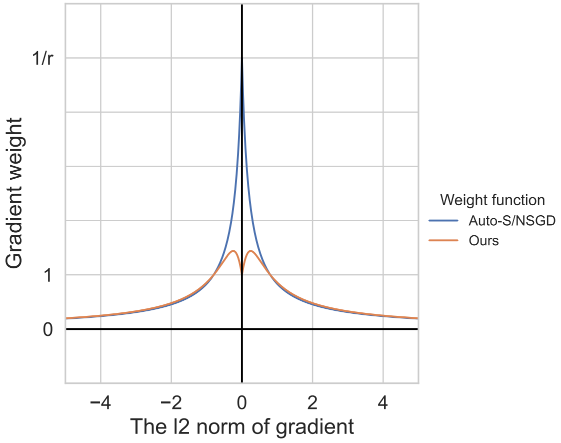

Figure 2: Gradient weight for calculating the batch-averaged gradient of our method and the Auto-S/NSGD method for different gradient norms.

results in an overwhelming deviation between the automatically clipped batch gradient and the actual batch gradient.

Intuitively, we hope that samples with different magnitudes of gradient norm will receive similar order of weights to preserve the average of clipped gradients as close to the original batch-averaged gradient as possible. Based on this, we propose **D**ifferentially **P**rivate **P**er-**S**ample **A**daptive **C**lipping (DP-PSAC) algorithm, by adopting a non-monotonous adaptive weight function. We summarize our contributions as follows:

- We propose a per-sample adaptive clipping algorithm, which is a new perspective and orthogonal to dynamic adaptive noise (Du et al. 2021) and coordinate clipping methods (Pichapati et al. 2019; Asi et al. 2021), and prove that it can be as private as currently used privacy-preserving optimization algorithms.
- We show how our algorithm converges in non-convex settings and provide a convergence error bound under DP. In addition, we demonstrate that DP-PSAC has a lower non-vanishing bound than Auto-S/NSGD.
- We demonstrate the empirical superiority of the proposed algorithm through extensive experiments while obtaining new state-of-the-art performance of differentially private learning on several datasets.

## Related Work

Deep learning based on gradient clipping and the Gaussian mechanism has become the most popular differentially private learning scheme. Constant clipping was firstly adopted in (Abadi et al. 2016) to equip SGD with privacy protection, called DP-SGD. Subsequentially, it was well studied in a series of works (Wang, Ye, and Xu 2017; Li et al. 2022; Wang, Chen, and Xu 2019; Kuru et al. 2022; Mangold et al. 2022; Bassily, Guzmán, and Menart 2021; Yu et al. 2021a; Wang et al. 2022; Wu et al. 2021; Esipova et al. 2022) to apply DP to other optimization algorithms, such as DP-AdaGrad, DP-SVRG, and ApolySFW. From a theoretical perspective, Zhang et al. (2020a) and Zhang et al. (2020b) analyzed the convergence of clipped SGD. From the perspective of application, DP-Lora (Yu et al. 2022) and RGP (Yu et al. 2021b) enabled differential privacy learning for large-scale model fine-tuning through methods such as low-rank compression.

Nevertheless, it is shown that the optimal threshold is always changing during the optimization process (van der Veen et al. 2018). Numerous studies are proposed to dynamically adjust the threshold in training in order to lessen the impact of a fixed threshold on the performance of DP-based algorithms. Among them, Andrew et al. (2021) predicted the optimal clipping threshold using extra privacy budget during the optimization process. Du et al. (2021) proposed to dynamically decrease the clipping threshold and noise magnitude along with the iteration round $t$. More fine-grained, some works (Pichapati et al. 2019; Asi et al. 2021) proposed axis-level adaptive clipping and noise addition methods, giving different clipping thresholds and non-homogeneous noise to the gradient components on a different axis. Despite the great success of these algorithms, the initial threshold still needs to be manually set, and the final performance is sensitive to the initial threshold.

To get rid of the dependence of differentially private learning on the clipping threshold, Bu et al. (2022) and Yang et al. (2022) concurrently proposed to constrain the gradient sensitivity with normalization, called Automatic Clipping (Auto-S) or Normalized SGD (NSGD). They showed that when normalizing all gradients to the same magnitude, the learning rate and the clipping hyperparameter can be coupled, thus only the one hyperparameter need to be tuned. However, this method suffers from a large deviation between their normalized batch-averaged gradient and the unnormalized one when some gradient norms in a batch are tiny. The proposed algorithm in this paper alleviates the above problem by reducing the size of deviation and achieves better theoretical and experimental results.

# Preliminary

## Notations and Definitions

Throughout the paper, we will let $\|\cdot\|$ denote the $\ell_2$ norm of a vector and $\langle\cdot,\cdot\rangle$ denote the inner product of two vectors. The gradient of $f(x)$ is represented by $\nabla f(x)$. The training dataset for the optimization problem is represented by $D$. The probability that event $z$ occurs is represented by Pr[z]. A random variable's mathematical expectation is denoted by $\mathbb{E}(\cdot)$. We consider the following empirical risk minimization problem:

$$\underset{x\in R^d}{min}\, f(x) := \frac{1}{|D|}\sum_{\xi_i\in D} f(x,\xi_i),$$

where $f(x,\xi_i)$ is the loss function with respect to data point $\xi_i$. In addition, we use $x^*$ to indicate the optimal solution to the above problem.

DP (Dwork, Roth et al. 2014) provides a formal definition of individual privacy, with the intuition that the result of a random algorithm on a dataset should not be different too much with or without one data point:

**Definition 1** ($(\epsilon,\delta)$-DP)**.** *A randomized mechanism $\mathcal{M}$ : $\mathcal{D}\to\mathcal{R}$ offers $(\epsilon,\delta)$-differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$ differing by a single data point and any $S\subset\mathcal{R}$ it satisfies that:*

$$\Pr[\mathcal{M}(D)\in S] \le e^{\epsilon}\Pr[\mathcal{M}(D')\in S] + \delta.$$

In deep learning training, $(\epsilon,\delta)$-DP is the most widely employed type of DP. It mainly relies on the Gaussian mechanism, which involves introducing Gaussian noise to gradients. Its privacy budget can calculated by means of the moments accountant (Abadi et al. 2016), Rényi-DP (Mironov 2017) or $f$-DP (Dong, Roth, and Su 2019).

## Assumptions

In this paper, we formulate the following assumptions, all of which are common and basic in past works (Ghadimi and Lan 2013; Bu et al. 2022; Yang et al. 2022).

**Assumption 1** ($(L_0, L_1)$-generalized smooth)**.** *We assume that $f(x)$ is $(L_0, L_1)$-generalized smooth, this is, for all $x, y \in \mathbb{R}^d$, there exist constants $L_0 > 0$ and $L_1 \ge 0$ such that $\|\nabla f(x) - \nabla f(y)\| \le (L_0 + L_1\|\nabla f(x)\|)\,\|x-y\|$.*

**Assumption 2** (Bounded variance)**.** *For all $x\in\mathbb{R}^d$, there exist constants $\tau_0 > 0$ and $0\le\tau_1<1$, such that $\|g(x,\xi_i) - \nabla f(x)\| \le \tau_0 + \tau_1\|\nabla f(x)\|$ with probability 1.*

## Review: Normalized/Automatic DP Training

The fundamental method of Normalized/Automatic differentially private training (Bu et al. 2022; Yang et al. 2022) is to limit the magnitude of each gradient by using normalization rather than clipping. Specifically, it normalizes all per-sample gradients to the same size:

$$\widetilde{g} = \mathrm{Clip}(g) = g/\|g\|.$$

The algorithm called Auto-S/NSGD (Bu et al. 2022; Yang et al. 2022) jumps out of the original gradient clipping framework, so that the gradient clipping parameter and the learning rate are coupled:

$$\begin{aligned} x_t - x_{t+1} &= \frac{\eta_t}{|B_t|}\left(\sum_{i\in B_t}\frac{Cg_{t,i}}{\|g_{t,i}\|} + \mathcal{N}(0, C^2\sigma^2)\right) \\ &= \frac{\eta_t C}{|B_t|}\left(\sum_{i\in B_t}\frac{g_{t,i}}{\|g_{t,i}\|} + \mathcal{N}(0,\sigma^2)\right). \end{aligned}$$

As a result, it is unnecessary to tune the hyperparameter $C$. Additionally, a regularization term $r$ is added to the scaling factor to enhance training stability:

$$\widetilde{g} = \mathrm{Clip}(g) = g/\left(\|g\|+r\right),$$

where $r$ is usually set to 0.1 or less (Bu et al. 2022).

On the one hand, Auto-S/NSGD outperforms standard clipping-based techniques on numerous vision and language tasks. On the other hand, it eliminates reliance on the clipping threshold and reduces the searching space for hyperparameters. The algorithm proposed in this paper is a refinement of Auto-S/NSGD.

# Motivation

## Small Gradients Should not Get Huge Gains

**The contribution of small gradients are negligible.** The gradients of the samples in the batch are mathematically averaged to produce the update for each iteration of batch SGD without clipping. The gradient sizes for various samples within a batch may differ over orders of magnitude. Therefore, small gradient samples have little impact on the batch-averaged gradient for the entire batch. We calculate the cosine similarity between each sample's gradient and the actual batch-averaged gradient to determine how much each sample contributed to the final update. Giving very large weights to small gradient samples will result in a significant difference between the normalized batch-averaged gradient and the unnormalized gradient, as shown in Figure 1 where larger individual gradients maintain higher cosine similarity to the true batch average while smaller gradient samples are almost orthogonal or even negative to it. Additional datasets have produced similar results (Appendix D).

**Monotonic weights bring larger convergence errors.** Recall that the update in Auto-S/NSGD is equivalent to using a weighted average of per-sample gradients:

$$G_{\mathrm{batch}} = \frac{1}{|B_t|}\sum_{i\in B_t}\tilde{g}_{t,i} = \frac{1}{|B_t|}\sum_{i\in B_t} w_{t,i}g_{t,i},$$

where $w_{t,i}$ is monotonically decreasing with respect to $\|g_{t,i}\|$, i.e. $w_{t,i} = 1/(\|g_{t,i}\| + r)$. This leads to a larger learning rate for a smaller individual gradient. As a result, in the later stages of the optimization process, the magnitude of the majority of individual gradients tends to zero, but the size of the update is still in the same order as that in the beginning, making steady convergence more challenging. This intuition is also reflected in its theoretical analysis. The norm of the gradient in Auto-S/NSGD has an $O(r^{-1})$ non-vanishing upper bound, which cannot be reduced as the number of iterations increases.

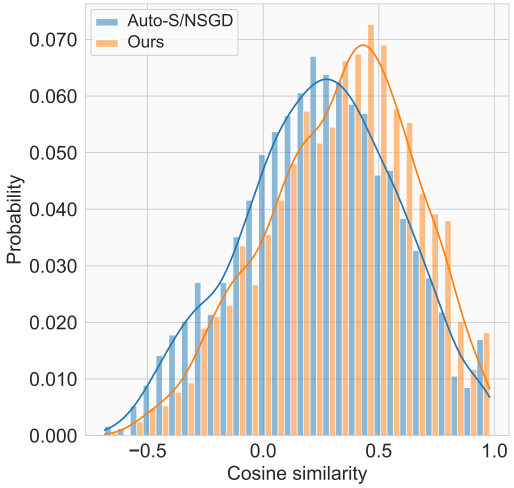

Figure 3: Cosine similarity histogram between the weighted batch-averaged gradients used in different methods and the real batch-averaged gradients.

### Non-Monotonous Adaptive Weight Function

We provide a non-monotonic adaptive weight function that gives small gradient samples a weight near 1 while weighting large gradients similarly to $1/\|g_{t,i}\|$:

$$w(g_{t,i}) = 1/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right).$$

Our weight function can provide weights that are closer to automatic clipping when the gradient is large, as in Figure 2. Additionally, we restrict the gradient's weight to a certain order of magnitude when it is small in order to lessen the overall deviation. We offer both theoretical and experimental evidence of the benefits of our adaptive weight function.

We describe our algorithmic pipeline and theoretical contributions in more detail in the following section.

## Per-Sample Adaptive Clipping Training

Here, we formally define the differentially private training algorithm DP-PSAC based on the per-sample adaptive clipping method. In the $k$-th iteration, The $i$-th gradient $g_{t,i}$ is clipped as

$$\widetilde{g}_{t,i} = \mathrm{Clip}(g_{t,i}) = Cg_{t,i}/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right),$$

where $C$ is the hyperparameter for clipping. Then, we can define the clipping weight (scaling factor) as

$$w(g_{t,i}) = \frac{\widetilde{g}_{t,i}}{Cg_{t,i}} = 1/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right).$$

Algorithm 1: DP-PSAC

**Input**: initial weights $x_0$ ,learning rate $\eta_t$ , batch size $B$, dataset $\mathcal{S} = (z_1, ..., z_N)$, privacy budget $(\epsilon, \delta)$, max clipping threshold $C$, the number of iterations $T$

```
Compute the standard deviation $\sigma$ of noise based on Theorem 1
for iteration $t = 0, ..., T-1$ do
    Sample a batch $\mathcal{D}_t := \{z_i^t\}_{i=1}^b$ from $\mathcal{S}$ uniformly with replacement
    Compute the gradient $g_{t,i}$ for each sample
    $\widetilde{g}_{t,i} = Cg_{t,i}/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right)$
    $\hat{g}_t = \sum_{i=1}^{b} \widetilde{g}_{t,i} + \mathcal{N}(0, C^2\sigma^2)$
    $x_{t+1} = x_t - \frac{\eta_t}{B}\hat{g}_t$
end for
```

As the result, the model increment in the $t$-th iteration can be formulated as below:

$$\begin{aligned}
&\Delta x_t \\
= &-\frac{\eta_t}{B}\left(\sum_{i=1}^{B} \widetilde{g}_{t,i} + \mathcal{N}(0, C^2\sigma^2)\right) \\
= &-\frac{\eta_t}{B}\left(\sum_{i=1}^{B} Cg_{t,i}/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right) + \mathcal{N}(0, C^2\sigma^2)\right) \\
= &-\frac{\eta_t C}{B}\left(\sum_{i=1}^{B} g_{t,i}/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\| + r}\right) + \mathcal{N}(0, \sigma^2)\right).
\end{aligned}$$

The clipping parameter $C$ does not require adjustment because it is coupled with the learning rate $\eta_t$, as can be seen from this equality. The entire procedure of per-sample adaptive gradient clipping-based differential privacy training is summarized in Algorithm 1.

We compute the cosine similarity between the batch-averaged gradient that was weighted using different functions and the true batch-averaged gradient in the same iteration in order to determine the deviation between the two gradients. The greater the cosine similarity, the closer the two gradients are. We run both our weight function and that in Auto-S/NSGD five times each on the Fashion-MNIST dataset, measuring the cosine similarity between the weighted batch-averaged gradient and the true batch-averaged gradient every 10 iterations. As shown in Figure 3, compared with Auto-S/NSGD, our method has a higher percentage of gradients with larger similarity, which demonstrates that our method is statistically closer to the true batch-averaged gradient than Auto-S/NSGD. Besides, for the "lazy region" problem of Auto-V (Bu et al. 2022), we show that our method can solve this problem better than Auto-S through simulation experiments under the same setting (Appendix C).

It should be highlighted that our method applies an adaptive norm constraint depending on the properties of each gradient sample, which is a novel and unexplored viewpoint. Although Auto-S and NSGD are incredibly close to this per-

| Method | Clipping threshold | Additional assumption | Non-vanishing bound |
|---|---|---|---|
| DP-SGD (Yang et al. 2022) | Yes | $c > \frac{2\tau_0}{1-\tau_1}$ | / |
| Auto-S/NSGD (Bu et al. 2022; Yang et al. 2022) | No | $p(\Delta) = p(-\Delta)$ or $r > \tau_0$ | $\mathcal{O}(r^{-1})$ |
| DP-PSAC (Ours) | No | / | $\mathcal{O}(r^{-1/2})$ |

Table 1: Comparison of theoretical results of DP-SGD, Auto-S/NSGD and DP-PSAC.

spective, they focus on scaling all gradient norms to the same or similar size, which limits their adaptability.

## Privacy Guarantee of DP-PSAC

To achieve privacy protection, existing learning methods with DP such as DP-SGD mainly adopt two techniques, namely clipping gradients and adding Gaussian noise. The first technique is used to limit the privacy sensitivity of gradients such that $\|g\| \leq C$ and the second technique is used to apply the Gaussian mechanism (Dong, Roth, and Su 2019) to achieve DP. We observe the per sample adaptive clipped gradient in DP-PSAC satisfies $\widetilde{g}_{t,i} = Cg_{t,i}/\left(\|g_{t,i}\| + \frac{r}{\|g_{t,i}\|+r}\right) \leq C$, which means that DP-PSAC can achieve the same privacy-sensitivity constraint for gradients as DP-SGD. Furthermore, this means that the privacy analysis on DP-SGD still can be applied on DP-PSAC.

**Theorem 1.** *There exist constants $c_1$ and $c_2$ so that given the sampling probability $q = B/N$ and the number of iterations $T$, for any $\epsilon \leq c_1 q^2 T$ and $\delta > 0$, Algorithm 1 is $(\epsilon, \delta)$-differentially private if we choose*

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\epsilon}.$$

## Convergence Guarantee of DP-PSAC

Without losing generality, we prove that DP-PSAC converges to the stationary point, i.e. $\lim_{t\to+\infty} \|\nabla f(x_t)\| = 0$, which is widely adopted criterion for general non-convex optimization (Ghadimi and Lan 2013). All detailed proofs are deferred to the appendix due to the page limitation. We first give Theorem 2 to bound the expected gradient norm with the number of iteration $T$ and the variance of the Gaussian noise $\sigma^2$.

**Theorem 2.** *For $f(x)$ satisfying Assumptions 1, 2. Given an arbitrary noise multiplier $\sigma$ and constant $r \in (0, 1]$, we run DP-PSAC for the number of iterations $T \geq A(L, \tau, d, r, \sigma, B)$ (Lemma 6 in Appendix. B) with a constant learning rate*

$$\eta = \sqrt{\frac{2B^2}{d\sigma^2 T(L_0 + L_1(\tau_0 + 1))}}.$$

*We can observe that the gradient norm can be bounded by the following inequality:*

$$\mathbb{E}(\underset{0\leq t<T}{min} \|\nabla f(x_t)\|) \leq \mathcal{O}\left(\sqrt[4]{\frac{d\sigma^2}{TB^2}} + \sqrt[4]{\frac{B^2}{Td\sigma^2}}\right) + \frac{8\tau_0^2(1+\tau_0)}{3N(\tau_0,\tau_1,r)(1-\tau_1)^3(\tau_0 + \frac{r(1-\tau_1)}{2\tau_0+r(1-\tau_1)})(2\sqrt{r}-r)},$$

*where*

$$N(\tau_0,\tau_1,r) = \min\left(\frac{\tau_0}{1-\tau_1}, \frac{2\tau_0^2 + r\tau_0(1-\tau_1)}{4\tau_0^2 + 2r\tau_0(1-\tau_1) + r(1-\tau_1)^2}\right).$$

It can be inferred from Theorem 1 that the noise multiplier $\sigma$ depends on the privacy parameters $(\epsilon, \delta)$ and the number of iterations $T$. In order to achieve the DP guarantee, Theorem 2 can be extended to observe the following Corollary by properly setting $\sigma$.

**Corollary 1.** *With the same setting as Theorem 2, we set $T \geq \mathcal{O}(N^2\epsilon^2/(d\log(1/\delta)))$. To achieve $(\epsilon, \delta)$ DP guarantees with a sufficient number of samples $N \geq L_1 A'(\epsilon, \delta, \tau, L, d, r)$ (Lemma 8 in Appendix. B), the expected gradient norm can be bounded as:*

$$\mathbb{E}(\underset{0\leq t<T}{min} ||\nabla f(x_t)||) \leq \mathcal{O}\left(\sqrt{\frac{\sqrt{d\log(1/\delta)}}{N\epsilon}}\right) + \frac{8\tau_0^2(1+\tau_0)}{3N(\tau_0,\tau_1,r)(1-\tau_1)^3(\tau_0 + \frac{r(1-\tau_1)}{2\tau_0+r(1-\tau_1)})(2\sqrt{r}-r)}.$$

From Theorem 2 and Corollary 1, it can be observed that when we choose a suitable learning rate, DP-PSAC can achieve the convergence rate of $\mathcal{O}(\sqrt{\frac{\sqrt{d\log(1/\delta)}}{N\epsilon}})$, which is consistent with the latest results of the differentially private non-convex optimization (Bu et al. 2022; Yang et al. 2022).

**Remark 1.** *There are no additional assumptions to limit the hyperparameters or distribution of gradients in the convergence proof of Theorem 2.*

In previous work, the convergence results of (Bu et al. 2022) rely on the assumption that the gradient distribution is symmetric. The convergence results of Yang et al. (2022) depend on the assumption that the regularization term satisfies $r > \tau_0$, but $\tau_0$ is difficult to observe. DP-PSAC does not rely on extra-assumed properties because its weight function is non-monotonic and there is a strict upper bound that does not depend on $\|\nabla f(x)\|$. We summarize the theoretical comparison of different algorithms in Table 1. We demonstrate the theoretical superiority of this weight function by briefly introducing our proof procedure.

Similar to conventional non-convex optimization based on $(L_0, L_1)-$generalized smooth, our convergence analysis is developed by the following lemma:

**Lemma 1.** *Under the premise of Assumption 1, for each iteration $t$, letting $w_{t,i} = w(g_{t,i})$ indicate the sample weight*

| Task | Model | $(\epsilon, \delta)$ | DP-SGD(%) | Auto-S/NSGD(%) | DP-PSAC(%) |
|---|---|---|---|---|---|
| MNIST | CNN | $(3, 1e-5)$ | $98.12 \pm 0.07$ | $98.17 \pm 0.07$ | $\mathbf{98.24 \pm 0.07}$ |
| FashionMNIST | CNN | $(3, 1e-5)$ | $86.22 \pm 0.29$ | $86.30 \pm 0.21$ | $\mathbf{86.56 \pm 0.16}$ |
| CIFAR10 | SimCLRv2 | $(2, 1e-5)$ | $92.47 \pm 0.07$ | $92.72 \pm 0.16$ | $\mathbf{92.78 \pm 0.13}$ |
| imagenette | ResNet9 | $(8, 1e-4)$ | $63.66 \pm 0.05$ | $63.44 \pm 0.24$ | $\mathbf{64.00 \pm 0.16}$ |
| CelebA [Smiling] | ResNet9 | $(8, 5e-6)$ | $91.17 \pm 0.06$ | $91.10 \pm 0.02$ | $\mathbf{91.41 \pm 0.02}$ |
| CelebA [Male] | ResNet9 | $(8, 5e-6)$ | $95.46 \pm 0.03$ | $95.48 \pm 0.04$ | $\mathbf{95.57 \pm 0.02}$ |
| CelebA Multi-label | ResNet9 | $(8, 5e-6)$ | $88.56 \pm 0.04$ | $88.49 \pm 0.10$ | $\mathbf{88.69 \pm 0.01}$ |

Table 2: Test accuracy of DP-SGD, Auto-S and DP-PSAC on image classification tasks.

*function, the following inequality holds:*

$$\mathbb{E}_t[f(x_{t+1})] - f(x_t) \leq -\eta \mathbb{E}_t[\frac{1}{B}\sum_{i=1}^{B}\langle w_{t,i}\nabla f(x_t), g_{t,i}\rangle]$$

$$+\mathbb{E}_t\frac{L_0 + L_1\|\nabla f(x_t)\|}{2}\eta^2(\frac{d\sigma^2}{B^2} + \frac{1}{B}\sum_{i=1}^{B}\|w_{t,i}g_{t,i}\|^2).$$

For the first term, it can be scaled to $\mathcal{O}(\eta\|\nabla f(x_t)\|)$ (when $\|\nabla f(x_t)\| \geq \tau_0/(1-\tau_1)$) or $\mathcal{O}(\eta\|\nabla f(x_t)\|^2) + \mathcal{O}(\eta)$) (when $\|\nabla f(x_t)\| < \tau_0/(1-\tau_1)$) by lemma 5 in Appendix A. For the second term, a suitable $\eta$ is chosen so that it can be upper bounded by $\mathcal{O}(\eta^2) + \mathcal{O}(\eta w_t\|\nabla f(x_t)\|^2)$. Since $\mathcal{O}(\eta w_t\|\nabla f(x_t)\|^2)$ is consistent with the form of the first item, it can be similarly scaled as the first term. At this point, we only need to take $\eta \propto 1/\sqrt{T}$, and sum up the above formula from $t = 1$ to $T$ to deduce convergence result.

The hardest part of dealing with the second term is bounding $(L_0 + L_1\|\nabla f(x_t)\|^2)w_t$ with a constant that does not depend on $\nabla f(x_t)$. Due to its monotonically decreasing weight function, NSGD can only find an upper bound that does not depend on $\nabla f(x_t)$ by assuming $r > \tau_0$. In our method, we can find a constant upper bound without making any additional assumptions by the following lemma.

**Lemma 2.** *Under Assumption 2, for any $r \in (0, 1]$, $w_{t,i} = 1/(\|g_{t,i}\| + r/(\|g_{t,i}\| + r))$, we have the following inequality:*

$$(L_0 + L_1\|\nabla f(x_t)\|)w_{t,i} \leq \max(\frac{L_0(1-\tau_1) + L_1(\sqrt{r}-r+\tau_0)}{(1-\tau_1)(2\sqrt{r}-r)},$$

$$\frac{L_0(1-\tau_1) + L_1\tau_0 + L_1\sqrt{r}}{\sqrt{r}(1-\tau_1)}).$$

Since Lemma 2 does not use any additional assumptions on $r$, any choice of $r \in (0, 1]$ is feasible to achieve the the theoretical results in Corollary 1.

**Remark 2.** *Theorems 2 and Corollary 1 give the non-vanishing bound in the order of $\mathcal{O}(r^{-1/2})$, which is superior compared with $\mathcal{O}(r^{-1})$ in NSGD (Yang et al. 2022).*

The normalization-based method innovatively solves the problem that the clipping threshold is difficult to tune. But it introduces an immortal deviation to the optimization process, which cannot be eliminated by increasing the number of iterations or the privacy budget. At the same time, the upper bound of this deviation is inversely proportional to the multiplication of $r$, which is a constant from 0 to 1 (e.g. 0.01). Our method reduces the upper bound on immortality deviation from $\mathcal{O}(r^{-1})$ to $\mathcal{O}(r^{-1/2})$ by controlling the maximum weight of the weight function.

## Experiments

We evaluate the effectiveness of the proposed algorithm on multiple datasets for both image and sentence classification.

**Hardware and software information** All experiments are performed on a server with an Intel Xeon Platinum 8369B CPU, an NVIDIA A100 GPU, and 125GB memory. The operating system is Ubuntu 20.04 and the CUDA Toolkit version is 11.3. All computer vision experimental training procedures are implemented based on the latest versions of Pytorch and Opacus (Yousefpour et al. 2021). The natural language processing experiments are based on private-transformers (Li et al. 2021) of version 0.1.0, transformers of version 4.11.3, and the latest version of Pytorch.

### Image Classification Task

**Dataset** We conduct extensive experiments on multiple image classification datasets, including MNIST (LeCun et al. 1998), FashionMNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR10 (Krizhevsky, Hinton et al. 2009), imagenette (a subset of imagenet (Deng et al. 2009) with ten labels), and CelebA (Liu et al. 2015).

**Method** Our main comparison methods are DP-SGD and Auto-S/NSGD. For DP-SGD, we refer to the implementations of Papernot et al. (2021), Tramer and Boneh (2020), and Klause et al. (2022), which achieves the state-of-the-art performance of Abadi's clipping-based DP learning on different image datasets. For Auto-S/NSGD, we adopt the same settings as Bu et al. (2022), which exhibits the state-of-the-art differentially private optimization performance. Specifically, we train a four-layer CNN model on MNIST and FashionMNIST, which have the same settings as Tramer and Boneh (2020). Then for CIFAR10, we keep the same experimental setup as Tramer and Boneh (2020) and use pre-trained SimCLRv2 (Chen et al. 2020). Further, we train a ResNet9 (He et al. 2016) model on imagenette and CelebA to validate the performance of our method on more complex multi-classification and multi-label classification problems, and the experimental setup for this part is the same as previous works (Klause et al. 2022; Bu et al. 2022). We run all methods five times to get all of the results shown in Table 2.

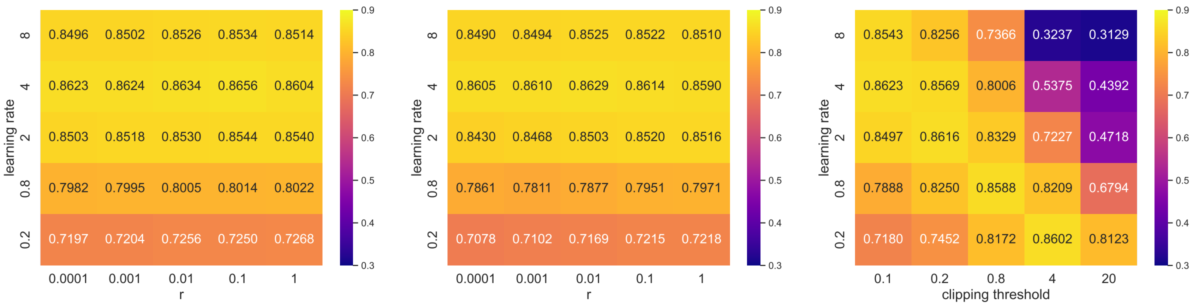

Figure 4: Test accuracy heatmap on the FashionMNIST task. Left: DP-PSAC. Middle: Auto-S/NSGD. Right: DP-SGD.

| Method | $\epsilon = 3$ | | | | $\epsilon = 8$ | | | |
|---|---|---|---|---|---|---|---|---|
| | MNLI(m/mm) | QQP | QNLI | SST-2 | MNLI(m/mm) | QQP | QNLI | SST-2 |
| DP-SGD (Li et al. 2021) | 82.45/82.99 | 85.56 | 87.42 | 91.86 | 83.20/83.46 | 86.08 | 87.94 | 92.09 |
| Auto-S (Bu et al. 2022) | **83.22**/83.21 | 85.76 | 86.91 | 92.32 | **83.82**/83.55 | 86.58 | 87.85 | 92.43 |
| DP-PSAC(Ours) | 82.74/**83.36** | **85.83** | **87.48** | **92.43** | 83.65/**83.87** | **86.60** | **88.03** | **92.55** |

Table 3: Test accuracy of sentence classification for DP-SGD, Auto-S, and DP-PSAC with $\epsilon = 3, 8$.

**Result** Firstly, as shown in Figure 4, we notice that the test accuracy changes very little with $r$ in DP-PSAC and Auto-S/NSGD for the same learning rate. Correspondingly, when using DP-SGD, the test accuracy is very sensitive to the clipping threshold $C$. This shows that the hyperparameter $r$ is more stable and easier to tune than the clipping threshold $C$. Usually, we only need to set r to a positive number not larger than 1, for instance, 0.1, to get a near-optimal result. It can be observed from Table 2 that, our method outperforms both DP-SGD and Auto-S in differentially private learning on the mainstream image classification datasets. In particular, DP-PSAC is more robust than Auto-S/NSGD since it exhibits a lower level of variance. This corroborates with our theoretical result that DP-PSAC has a lower non-vanishing bound than Auto-S/NSGD. These evaluations show that our algorithm performs well on logistic regression, basic CNN, and ResNet, and its high performance is independent of any particular network architecture.

## Sentence Classification Task

**Dataset** We used four sentence classification datasets from the GLUE benchmark dataset, including MNLI (multi-genre inference) (Williams, Nangia, and Bowman 2017), QQP (equivalence classification), QNLI (Question-answering inference) (Rajpurkar et al. 2016), and SST-2 (sentiment classification) (Socher et al. 2013).

**Method** The code of the sentence classification experiment refers to Li et al. (2021). In order to ensure the adequacy of the experiment, we use the roberta-base model to compare the full-parameter training performance of DP-PSAC, Auto-S/NSGD (Bu et al. 2022; Yang et al. 2022) and DP-SGD (Li et al. 2021) on four different datasets under large($\epsilon = 3$) and small($\epsilon = 8$) noise conditions, respectively. The test accuracy for DP-SGD and Auto-S are taken from (Li et al. 2021) and (Bu et al. 2022), respectively.

**Result** Table 3 shows that DP-PSAC performs better than or similar to the best baseline in both small and large noise conditions. Specifically, on the MNLI dataset, our method outperforms Auto-S/NSGD on the MNLI-mm test set, which is not independent and identically distributed with the training set, and outperforming DP-SGD on both MNLI-m and MNLI-mm. For the QQP dataset, a sentence classification dataset with uneven sample distribution, DP-PSAC achieves higher accuracy than the baselines. Although Auto-S/NSGD does not achieve better results than DP-SGD on the QNLI dataset, our method, as an improvement of Auto-S/NSGD, achieves the latest state-of-the-art. Meanwhile, on the SST-2 dataset, our method not only achieves better accuracy but also enables our model performance at $\epsilon = 3$ to reach the previous state-of-the-art at $\epsilon = 8$.

## Conclusion

In this study, we propose a differentially private optimization approach with per-sample adaptive clipping, which can reduce deviation by giving gradients different weights according to their magnitudes while preserving privacy constraints. Without making any extrinsic assumptions, we investigate the convergence of DP-PSAC in non-convex scenarios and demonstrate that it offers a reduced upper bound on indestructible deviation than Auto-S/NSGD. Experimental results demonstrate that DP-PSAC accomplishes the state-of-the-art in differentially private optimization on both language and computer vision problems.

Per-sample adaptive clipping is a new perspective, which is different from adaptive clipping with iterations (Du et al. 2021; Andrew et al. 2021) and per-axis adaptation (Asi et al. 2021). In future work, we will consider to develop a data-driven adaptive weight function and more realistic application scenarios, such as resource offloading and flow detection in network (Yao et al. 2022; Zhou et al. 2023).

## Acknowledgements

This work was in part supported by the National Key Research and Development Program of China under Grant 2022YFB3102301, the China National Funds for Distinguished Young Scientists with No. 61825204, the NSFC Project with No. 61932016, No. 62101301, No. 62132011, and No. 62132009, the Beijing Outstanding Young Scientist Program with No. BJJWZYJH01201910003011, National Natural Science Foundation of China (U22B2031), CCF-AFSG Research Fund (CCF-AFSG RF20210023) , China Computer Federation (CCF)-Huawei Populus euphratica forest fund (CCF-HuaweiBC2021005), Chinese Association for Artificial Intelligence (CAAI)-Huawei MindSpore Open Fund (CAAIXSJLJJ-2020-014A).

The author thanks Kai Xiao, Weiqiang Wang, as well as the reviewers/SPC/AC for the constructive comments to improve the paper.

## References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Andrew, G.; Thakkar, O.; McMahan, B.; and Ramaswamy, S. 2021. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34: 17455–17466.

Asi, H.; Duchi, J.; Fallah, A.; Javidbakht, O.; and Talwar, K. 2021. Private adaptive gradient methods for convex optimization. In *International Conference on Machine Learning*, 383–392. PMLR.

Bassily, R.; Guzmán, C.; and Menart, M. 2021. Differentially private stochastic optimization: New results in convex and non-convex settings. *Advances in Neural Information Processing Systems*, 34: 9317–9329.

Bu, Z.; Wang, Y.-X.; Zha, S.; and Karypis, G. 2022. Automatic Clipping: Differentially Private Deep Learning Made Easier and Stronger. arXiv:2206.07136.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Ieee.

Dong, J.; Roth, A.; and Su, W. J. 2019. Gaussian differential privacy. arXiv:1905.02383.

Du, J.; Li, S.; Feng, M.; and Chen, S. 2021. Dynamic differential-privacy preserving sgd. arXiv:2111.00173.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4): 211–407.

Esipova, M. S.; Ghomi, A. A.; Luo, Y.; and Cresswell, J. C. 2022. Disparate Impact in Differential Privacy from Gradient Misalignment. arXiv:2206.07737.

Ghadimi, S.; and Lan, G. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4): 2341–2368.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Klause, H.; Ziller, A.; Rueckert, D.; Hammernik, K.; and Kaissis, G. 2022. Differentially private training of residual networks with scale normalisation. arXiv:2203.00324.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

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

Kuru, N.; Birbil, S. I.; Gürbüzbalaban, M.; and Yildirim, S. 2022. Differentially private accelerated optimization algorithms. *SIAM Journal on Optimization*, 32(2): 795–821.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, T.; Zaheer, M.; Reddi, S.; and Smith, V. 2022. Private adaptive optimization with side information. In *International Conference on Machine Learning*, 13086–13105. PMLR.

Li, X.; Tramer, F.; Liang, P.; and Hashimoto, T. 2021. Large language models can be strong differentially private learners. arXiv:2110.05679.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Mangold, P.; Bellet, A.; Salmon, J.; and Tommasi, M. 2022. Differentially private coordinate descent for composite empirical risk minimization. In *International Conference on Machine Learning*, 14948–14978. PMLR.

Mironov, I. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, 263–275. IEEE.

Papernot, N.; and Steinke, T. 2021. Hyperparameter Tuning with Renyi Differential Privacy. In *International Conference on Learning Representations*.

Papernot, N.; Thakurta, A.; Song, S.; Chien, S.; and Erlingsson, Ú. 2021. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9312–9321.

Pichapati, V.; Suresh, A. T.; Yu, F. X.; Reddi, S. J.; and Kumar, S. 2019. AdaCliP: Adaptive clipping for private SGD. arXiv:1908.07643.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. arXiv:1606.05250.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642.

Tramer, F.; and Boneh, D. 2020. Differentially Private Learning Needs Better Features (or Much More Data). In *International Conference on Learning Representations*.

van der Veen, K. L.; Seggers, R.; Bloem, P.; and Patrini, G. 2018. Three tools for practical differential privacy. arXiv:1812.02890.

Wang, D.; Chen, C.; and Xu, J. 2019. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, 6526–6535. PMLR.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30.

Wang, P.; Lei, Y.; Ying, Y.; and Zhang, H. 2022. Differentially private SGD with non-smooth losses. *Applied and Computational Harmonic Analysis*, 56: 306–336.

Williams, A.; Nangia, N.; and Bowman, S. R. 2017. A broad-coverage challenge corpus for sentence understanding through inference. arXiv:1704.05426.

Wu, X.; Wang, L.; Cristali, I.; Gu, Q.; and Willett, R. 2021. Adaptive Differentially Private Empirical Risk Minimization. arXiv:2110.07435.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747.

Yang, X.; Zhang, H.; Chen, W.; and Liu, T.-Y. 2022. Normalized/Clipped SGD with Perturbation for Differentially Private Non-Convex Optimization. arXiv:2206.13033.

Yao, S.; Wang, M.; Qu, Q.; Zhang, Z.; Zhang, Y.-F.; Xu, K.; and Xu, M. 2022. Blockchain-Empowered Collaborative Task Offloading for Cloud-Edge-Device Computing. *IEEE Journal on Selected Areas in Communications*, 40: 3485–3500.

Yousefpour, A.; Shilov, I.; Sablayrolles, A.; Testuggine, D.; Prasad, K.; Malek, M.; Nguyen, J.; Ghosh, S.; Bharadwaj, A.; Zhao, J.; Cormode, G.; and Mironov, I. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. arXiv:2109.12298.

Yu, D.; Naik, S.; Backurs, A.; Gopi, S.; Inan, H. A.; Kamath, G.; Kulkarni, J.; Lee, Y. T.; Manoel, A.; Wutschitz, L.; Yekhanin, S.; and Zhang, H. 2022. Differentially Private Fine-tuning of Language Models. In *International Conference on Learning Representations*.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2021a. Gradient perturbation is underrated for differentially private convex optimization. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 3117–3123.

Yu, D.; Zhang, H.; Chen, W.; Yin, J.; and Liu, T.-Y. 2021b. Large Scale Private Learning via Low-rank Reparametrization. In Meila, M.; and Zhang, T., eds., *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 12208–12218.

Zhang, B.; Jin, J.; Fang, C.; and Wang, L. 2020a. Improved Analysis of Clipping Algorithms for Non-convex Optimization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15511–15521. Curran Associates, Inc.

Zhang, J.; He, T.; Sra, S.; and Jadbabaie, A. 2020b. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In *International Conference on Learning Representations*.

Zhang, X.; Ji, S.; and Wang, T. 2018. Differentially private releasing via deep generative model (technical report). arXiv:1801.01594.

Zhou, G.; Liu, Z.; Fu, C.; Li, Q.; and Xu, K. 2023. An Efficient Design of Intelligent Network Data Plane. In *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32.