On the Convergence and Calibration of Deep Learning with Differential Privacy

Anonymous authors
Paper under double-blind review

Abstract

Differentially private (DP) neural network achieves the privacy usually at the cost of slower convergence (and thus lower performance) than its non-private counterpart. To analyze the difficulty of DP training, this work gives the first convergence analysis through the lens of training dynamics and the neural tangent kernel (NTK). We successfully characterize the effects of two key components in the DP training: the per-sample gradient clipping (flat or layerwise) and the noise addition. Our analysis not only initiates a general principled framework to understand the DP deep learning with any network architecture and loss function, but also motivates a new clipping method – the *global clipping*, that significantly improves the convergence, as well as preserves the same DP guarantee and computational efficiency as the existing method, which we term as *local clipping*.

Theoretically speaking, we precisely characterize the effect of per-sample clipping on the NTK matrix and show that the noise scale of DP optimizers does not affect the convergence in the *gradient flow* regime. In particular, we shed light on several behaviors that are only guaranteed by our global clipping. For example, the global clipping can preserve the positive semi-definiteness of NTK, which is almost certainly broken by the local clipping; DP gradient descent (GD) with global clipping converges monotonically to zero loss, while the convergence of local clipping can be non-monotone; the global clipping is surprisingly effective at learning *calibrated classifiers*, whereas existing DP classifiers are oftentimes over-confident and unreliable. Notably, our analysis framework easily extends to other optimizers, e.g., DP-Adam. We demonstrate through numerous experiments that DP optimizers equipped with global clipping perform strongly. Implementation-wise, the global clipping can be realized by inserting only one line of code into the Pytorch `Opacus` library.

1 Introduction

Deep learning has achieved tremendous success in many applications that involve crowdsourced information, e.g., face image, emails, financial status, and medical records. However, using such sensitive data raises severe privacy concerns on a range of image recognition, natural language processing and other tasks Cadwalladr & Graham-Harrison (2018); Rocher et al. (2019); Ohm (2009); De Montjoye et al. (2013; 2015). For a concrete example, researches have recently demonstrated multiple successful privacy attacks on deep learning models, in which the attackers can re-identify a member in the dataset using the location or the purchase record, via the membership inference attack Shokri et al. (2017); Carlini et al. (2019). In another example, the attackers can extract a person's name, email address, phone number, and physical address from the billion-parameter GPT-2 Radford et al. (2019) via the extraction attack Carlini et al. (2020). Therefore, many studies have applied differential privacy (DP) Dwork et al. (2006); Dwork (2008); Dwork et al. (2014); Mironov (2017); Duchi et al. (2013); Dong et al. (2019), a mathematically rigorous approach, to protect against leakage of private information Abadi et al. (2016); McSherry & Talwar (2007); McMahan et al. (2017); Geyer et al. (2017). To achieve this gold standard of privacy guarantee, since the seminal work Abadi et al. (2016), DP optimizers are applied to train the neural networks while preserving the accuracy of prediction. To name a few, researchers have proposed DP-SGD Abadi et al. (2016); Bassily et al. (2014) and DP-Adam Bu et al. (2019) for private deep learning, DP-SGLD Wang et al. (2015); Li et al. (2019) for Bayesian neural network, and DP-FedSGD and DP-FedAvg McMahan et al. (2017) for federated learning.

Algorithmically speaking, DP optimizers generally have two extra steps in comparison to non-DP standard optimizers: the per-sample gradient clipping and the random noise addition, so that DP optimizers descend in the direction of the averaged, clipped, noisy gradient (see Figure 2). These extra steps protect the resulting models against privacy attacks via the Gaussian mechanism (Dwork et al., 2014, Theorem A.1), at the expense of performance degradation compared to the non-DP deep learning, in terms of much slower convergence and lower utility. For example, state-of-the-art CIFAR10 accuracy with DP is $\approx 70\%$ without pre-training Papernot et al. (2020) (while non-DP networks can easily achieve over 95% accuracy), and similar performance drops have been observed on facial images, tweets, and many other datasets Bagdasaryan et al. (2019).

Empirically, many works have evaluated the effects of noise scale, batch size, clipping norm, learning rate, and network architecture on the privacy-accuracy trade-off Abadi et al. (2016); Papernot et al. (2020). However, despite the prevalent usage of DP optimizers, there is only limited understanding about their convergence behavior from a theoretical viewpoint Chen et al. (2020); Bu et al. (2022), which is necessary to understand and improve the deep learning with differential privacy.

Our Contributions In this work, we establish a principled framework to analyze the dynamics of DP deep learning, which helps demystify the phenomenon of the privacy-accuracy trade-off.

- We characterize the *general training dynamics* of deep learning with DP gradient methods (e.g. DP-GD and DP-Adam; see (4.2)). We show a fundamental influence of the DP training on the NTK matrix, which causes the convergence to worsen.
- We successfully separate the per-sample gradient clipping and noise addition in the continuous time analysis, showing that the clipping only affects the convergence but not the privacy, and that the noise only affects the privacy but not the convergence.
- On top of our convergence analysis, we propose a novel *global clipping* method that shares the same privacy guarantee and efficiency as the existing clipping. This leads to a *mix-up training* strategy that applies both clippings interchangeably.
- We demonstrate via numerous experiments that the global clipping significantly improve the loss convergence. Interestingly, our clipping effectively mitigates the *calibration* issue of existing DP classifiers, which usually exacerbates the "over-confidence" in non-DP models.
- Our global clipping is *easy-to-code* (see Appendix D) and *generalizable* to arbitrary optimizers, network architectures, loss functions, and tasks.

2 Warmup: Convergence of Non-Private Gradient Descent

We start by reviewing the standard, non-DP Gradient Descent (GD) for ***arbitrary neural network*** and ***arbitrary loss*** that can be represented as a sum of per-sample losses. In particular, we analyze the training dynamics of a neural network using the neural tangent kernel (NTK) matrix[1].

Suppose a neural network f is governed by weights $\mathbf{w}$, with samples $\boldsymbol{x}_i$ and labels y_i $(i = 1, ..., n)$. Denote the prediction by $f_i = f(\boldsymbol{x}_i, \mathbf{w})$, and the per-sample loss by $\ell_i = \ell(f(\boldsymbol{x}_i, \mathbf{w}), y_i)$ for some loss function ℓ. We define the objective function L to be the average of per-sample losses $L(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i, \mathbf{w}), y_i)$. The discrete gradient descent, with a step size η, can be written as: $\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \frac{\partial L}{\partial \mathbf{w}}^\top$. The corresponding *gradient flow*, i.e., the ordinary differential equation (ODE) describing the weight updates with infinitely small step size $\eta \to 0$ in the continuous time, is then: $\dot{\mathbf{w}}(t) = -\frac{\partial L}{\partial \mathbf{w}}^\top = -\frac{1}{n}\sum_i \nabla_{\mathbf{w}} \ell_i(t)$. Applying the chain rules to the gradient flow, we obtain the following general dynamics of the loss L,

$$\dot{L} = \frac{\partial L}{\partial \mathbf{w}}\dot{\mathbf{w}} = -\frac{\partial L}{\partial \mathbf{w}}\frac{\partial L}{\partial \mathbf{w}}^\top = -\frac{\partial L}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}}{\partial \mathbf{w}}\frac{\partial \boldsymbol{f}}{\partial \mathbf{w}}^\top\frac{\partial L}{\partial \boldsymbol{f}}^\top = -\frac{\partial L}{\partial \boldsymbol{f}}\mathbf{H}(t)\frac{\partial L}{\partial \boldsymbol{f}}^\top,$$

where $\frac{\partial L}{\partial \boldsymbol{f}} = \frac{1}{n}(\frac{\partial \ell_1}{\partial f_1}, ..., \frac{\partial \ell_n}{\partial f_n}) \in \mathbb{R}^{1\times n}$, and the Gram matrix $\mathbf{H}(t) := \frac{\partial \boldsymbol{f}}{\partial \mathbf{w}}\frac{\partial \boldsymbol{f}}{\partial \mathbf{w}}^\top \in \mathbb{R}^{n\times n}$ is known as the NTK matrix, which is positive semi-definite and crucial to analyzing the convergence behavior.

[1] We emphasize that our analysis works on any neural networks, not limited to the infinitely wide or over-parameterized ones. Put differently, we don't assume the NTK matrix $\mathbf{H}$ to be deterministic nor nearly time-independent, as was the case in Arora et al. (2019a); Lee et al. (2019); Du et al. (2018); Allen-Zhu et al. (2019); Zou et al. (2020); Fort et al. (2020); Arora et al. (2019b).

To give a concrete example, let ℓ be the MSE loss $\ell_i(\mathbf{w}) = (f(\boldsymbol{x}_i, \mathbf{w}) - y_i)^2$ and $L_{\text{MSE}} = \frac{1}{n}\sum_i \ell_i(\mathbf{w}) = \frac{1}{n}\sum_i (f_i - y_i)^2$, then $\dot{L}_{\text{MSE}} = -4(\boldsymbol{f} - \boldsymbol{y})^\top \mathbf{H}(t)(\boldsymbol{f} - \boldsymbol{y})/n^2$. Furthermore, if $\mathbf{H}(t)$ is positive definite, the MSE loss $L_{\text{MSE}} \to 0$ exponentially fast Du et al. (2018); Allen-Zhu et al. (2019); Zou et al. (2020) , the cross-entropy loss $L_{\text{CE}} \to 0$ at rate $O(1/t)$ and any loss convex in the prediction $L = \sum_i \ell_i/n$ converges to 0 Allen-Zhu et al. (2019).

3 Differentially Private Gradient Methods and Global Clipping

We now introduce the DP optimizers: one popular optimizer is the DP-SGD Song et al. (2013); Chaudhuri et al. (2011); Abadi et al. (2016); Bu et al. (2019) in Algorithm 1 and more optimizers such as DP-Adam can be found in Appendix F. In contrast to the standard SGD, the DP-SGD has two unique steps: the per-sample clipping (to bound the sensitivity of per-sample gradients) and the random noise addition (to guarantee the privacy of models), both are discussed in details via the Gaussian mechanism in Lemma 5.2. Some choices of clipping methods include the local clipping Abadi et al. (2016) and the automatic clipping Bu et al. (2022), which are illustrated in Figure 1. Notice that empirical observations have found that optimizers with the per-sample clipping (even when no noise is present) have much worse convergence and accuracy than their non-private counterparts Abadi et al. (2016); Bagdasaryan et al. (2019); Kurakin et al. (2022).

Algorithm 1 `DP-SGD` (with local or global flat per-sample clipping)

```
Parameters: initial weights $\mathbf{w}_0$, learning rate $\eta_t$, subsampling probability $p$, number of iterations $T$, noise scale $\sigma$, clipping norm $R$.
for $t = 0, \dots, T-1$ do
    Subsample a batch $I_t \subseteq \{1, \dots, n\}$ from training set with probability $p$
    for $i \in I_t$ do
        $v_t^{(i)} \leftarrow \nabla_{\mathbf{w}} \ell(f(\boldsymbol{x}_i, \mathbf{w}_t), y_i)$
        Option 1: $C_{local,i} = \min\left\{1, R/\|v_t^{(i)}\|_2\right\}$        ▷ Local clipping factor (existing)
        Option 2: $C_{global,i} \equiv \mathbb{I}\{\|v_t^{(i)}\|_2 \leq R\}$        ▷ Global clipping factor (ours)
        $\bar{v}_t^{(i)} \leftarrow C_i \cdot v_t^{(i)}$        ▷ Clip the gradient
    $\bar{V}_t \leftarrow \sum_{i \in I_t} \bar{v}_t^{(i)}$        ▷ Sum over batch
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \frac{\eta_t}{|I_t|}\left(\bar{V}_t + \sigma R \cdot \mathcal{N}(0, I)\right)$        ▷ Gaussian mechanism and Descent
```

We propose a new clipping method, namely the **global clipping** as in Option 2 of Algorithm 1, where the clipping operation takes place on all per-sample gradients that pass the screening procedure. The global clipping is similar to the *batch clipping* as all clipped per-sample gradients share the same clipping factor 1. At the high level, the idea of global clipping is to mitigate the bias introduced by the clipping, which in turn can preserve the positive semi-definiteness of the NTK matrix for large R (see Theorem 2).

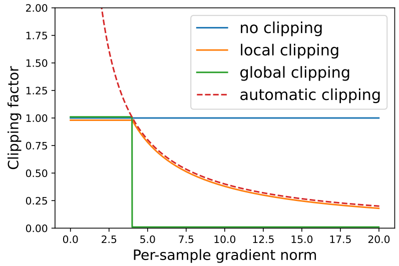

Figure 1: Clipping functions ($R = 4$).

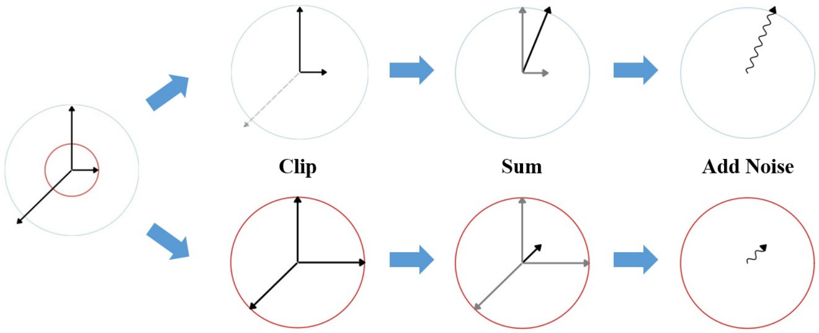

Figure 2: Illustration of global (upper) and local per-sample clipping (lower) in Algorithm 1. The black arrowed lines are three per-sample gradients. Red and grey circles mean small R and large R.

4 Convergence Analysis of DP Optimizers

In this section, we analyze the weight and loss dynamics of DP optimizers with the local or global per-sample clipping, denoted in the subscript, e.g., DP-SGD$_{local}$ and DP-SGD$_{global}$. Our narrative here focuses on the most widely used DP-GD for the sake of simplicity, and our analysis generalizes to other full-batch DP optimizers such as DP-HeavyBall, DP-RMSprop, and DP-Adam in Appendix F.

4.1 Effect of Noise Addition on Convergence

Our first result is easy yet surprising: the gradient flow of a noisy GD (4.1) is the same as that of a deterministic GD without the noise (4.2). Put differently, the noise addition has no effect on the convergence of DP optimizers in the continuous time analysis.

To elaborate this point, we consider the DP-GD with Gaussian noise, as in Algorithm 1,

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\eta}{n}\left(\sum_i \nabla_{\mathbf{w}} \ell_i C_i + \sigma R \cdot \mathcal{N}(0,1)\right). \tag{4.1}$$

Notice that this general formula covers both the non-DP GD ($\sigma = 0$ and $C_i \equiv 1$) and DP-GD with local or global clipping ($\sigma \neq 0$ and arbitrary R). Through Fact 4.1 and its proof in Appendix B, we show that the gradient flow of (4.1) is the same ODE regardless of the value of σ.

Fact 4.1. For all $\sigma \geq 0$, the gradient descent in (4.1) corresponds to the continuous gradient flow

$$d\mathbf{w}(t) = -\frac{1}{n}\sum\nolimits_i \nabla_{\mathbf{w}} \ell_i(t) C_i(t) dt. \tag{4.2}$$

This result indeed aligns the conventional wisdom[2] of tuning the clipping norm R first (e.g. setting $\sigma = 0.0$ or small) then the noise scale σ, since the convergence is not sensitive to σ (see Figure 3).

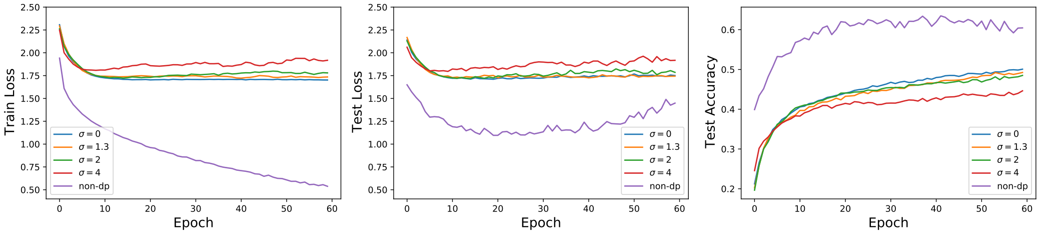

Figure 3: Performance of DP-SGD$_{local}$ with various noise σ on CIFAR10, using same setting as in Section 6.2. Notice that when $\sigma = 0$ and no clipping is applied, the test accuracy $\approx 62\%$.

Remark 4.2. Our proof of Fact 4.1 in Appendix B shows that the independence on σ holds true for general DP optimizer besides DP-GD, e.g. DP-Adam.

4.2 Effect of Per-Sample Clipping on NTK Matrix

We move on to analyze the effect of the per-sample clipping on the DP training (4.2). It has been empirically observed that the per-sample clipping results in worse convergence and accuracy even without the noise Bagdasaryan et al. (2019). We highlight that the NTK matrix is the key to understanding the convergence behavior and that the clipping affects NTK through its linear algebra properties, especially the positive semi-definiteness.

Definition 4.3. For a (not necessarily symmetric) matrix A, it is

1. *positive in quadratic form* if and only if $\boldsymbol{x}^\top A \boldsymbol{x} \geq 0$ for every non-zero $\boldsymbol{x}$;

2. *positive in eigenvalues* if and only if all eigenvalues of A are non-negative.

These two positivity definitions are equivalent for a symmetric or Hermitian matrix, but not so for non-symmetric matrices. We illustrate this difference in Appendix A with some concrete examples. Next, we introduce two styles of per-sample clippings. Both can be implemented locally or globally.

[2] See `https://github.com/pytorch/opacus/blob/master/tutorials/building_image_classifier.ipynb` and (Kurakin et al., 2022, Section 3.3).

Flat Clipping The DP-GD described in Algorithm 1 and (4.1), with the gradient flow (4.2), is equipped with the *flat* clipping McMahan et al. (2018). In words, the flat clipping upper bounds the entire gradient vector by a norm R. Using the chain rules, we get

$$\dot{L} = \frac{\partial L}{\partial \mathbf{w}} \dot{\mathbf{w}} = -\frac{1}{n^2} \sum_j \nabla_{\mathbf{w}} \ell_j \sum_i \nabla_{\mathbf{w}} \ell_i C_i = -\frac{\partial L}{\partial \boldsymbol{f}} \mathbf{H} \mathbf{C} \frac{\partial L}{\partial \boldsymbol{f}}^\top, \tag{4.3}$$

where $\mathbf{C}(t) = \text{diag}(C_1, \cdots, C_n)$ is the clipping factor matrix at time t.

Layerwise Clipping. We additionally analyze another widely used clipping – the *layerwise* clipping Abadi et al. (2016); McMahan et al. (2017); Phan et al. (2017). Unlike the flat clipping, the layerwise clipping upper bounds the r-th layer's gradient vector by a layer-dependent norm R_r, as demonstrated in Algorithm 2. Therefore, the DP-GD and its gradient flow with this layerwise clipping are: $\mathbf{w}_r(k+1) = \mathbf{w}_r(k) - \frac{\eta}{n}\left(\sum_i \nabla_{\mathbf{w}_r} \ell_i C_{i,r} + \sigma R_r \cdot \mathcal{N}(0,1)\right)$ and $\dot{\mathbf{w}}_r(t) = -\frac{1}{n}\sum_i \nabla_{\mathbf{w}_r} \ell_i C_{i,r}$. Then the loss dynamics is obtained by the chain rules:

$$\dot{L} = \sum_r \frac{\partial L}{\partial \mathbf{w}_r} \dot{\mathbf{w}}_r = -\sum_r \frac{\partial L}{\partial \boldsymbol{f}} \mathbf{H}_r \mathbf{C}_r \frac{\partial L}{\partial \boldsymbol{f}}^\top, \tag{4.4}$$

where the layerwise NTK matrix $\mathbf{H}_r = \frac{\partial \boldsymbol{f}}{\partial \mathbf{w}_r} \frac{\partial \boldsymbol{f}}{\partial \mathbf{w}_r}^\top$, and $\mathbf{C}_r(t) = \text{diag}(C_{1,r}, \cdots, C_{n,r})$.

In short, from (4.3) and (4.4), the per-sample clipping precisely changes the NTK matrix from $\mathbf{H} \equiv \sum_r \mathbf{H}_r$, in standard non-DP deep learning, to $\mathbf{HC}$ in DP training with flat clipping, and to $\sum_r \mathbf{H}_r \mathbf{C}_r$ in DP training with layerwise clipping. Subsequently, we will show that this breaks the NTK's positivity and worsens the convergence of DP training.

4.3 Per-Sample Clipping with Small R Breaks NTK Positivity

We start with the analysis of per-sample clipping when R is small, which is the prevailing choice De et al. (2022); Li et al. (2021). We show that the DP-GD with small R breaks the traditional positive semi-definiteness of the NTK matrix – it is not symmetric and maybe non-positive in the quadratic form (see Appendix A).

Theorem 1. *For an arbitrary neural network and a loss convex in f, suppose we clip the per-sample gradients in the gradient flow of DP-GD, and assume $\mathbf{H}(t) \succ 0$, then:*

1. *The flat clipping has the loss dynamics in (4.3), with NTK matrix $\mathbf{H}(t)\mathbf{C}(t)$, which is not symmetric and may be non-positive in quadratic form, but is positive in eigenvalues.*
2. *The layerwise clipping has the loss dynamics in (4.4), with NTK matrix $\sum_r \mathbf{H}_r(t)\mathbf{C}_r(t)$, which is not symmetric and may be non-positive in quadratic form and in eigenvalues.*
3. *For both flat and layerwise clipping, the loss $L(t)$ may not decrease monotonically.*
4. *If the loss $L(t)$ converges, for the flat clipping, it converges to 0; for the layerwise clipping, it may converge to a non-zero value.*

We prove Theorem 1 in Appendix B. The symmetry of NTK is almost surely broken by the per-sample clipping unless R is so large that the clipping does not happen. If furthermore the positive definiteness of NTK is broken, the loss convergence may be compromised, as depicted in Figure 9 and Figure 10.

4.4 Per-Sample Clipping with Large R Improves Convergence

We now consider the large R regime. In the extreme case, if $\|v_t^{(i)}\| < R$ for all t and all i, then global clipping is equivalent to local clipping: $C_{global} = C_{local} = 1$ because the clipping does not happen. In this case, DP-GD is essentially GD with additional Gaussian noise (also known as the gradient langevin dynamics; GLD)[3] and the gradient flow of DP-GD from (4.2) is the same as the non-DP GD: for flat or layerwise clipping, $d\mathbf{w}(t) = -\frac{1}{n}\sum_i \nabla_{\mathbf{w}} \ell_i(t) dt$ and $\dot{L} = -\frac{\partial L}{\partial \boldsymbol{f}} \mathbf{H} \frac{\partial L}{\partial \boldsymbol{f}}^\top$. Hence we obtain the following result for DP training with large R.

Theorem 2. *For an arbitrary neural network and a loss convex in f, suppose we clip the per-sample gradients in the gradient flow of DP-GD such that $\|v_t^{(i)}\|_2 \leq R$, and assuming $\mathbf{H}(t) \succ 0$, then:*

[3] Note that GLD is widely applied to train Bayesian neural networks, whose capability of uncertainty quantification implies the amazing calibration of DP training in Section 6.

1. *The flat (resp. layerwise) clipping has loss dynamics in (4.3) (resp. (4.4)), with NTK matrix* $\mathbf{H}(t) \equiv \sum_r \mathbf{H}_r(t)$ *that is symmetric and positive definite.*
2. *For both flat and layerwise clipping, the loss* $L(t)$ *decreases monotonically to 0.*

We prove Theorem 2 in Appendix B. Our findings from Theorem 1 and Theorem 2 are visualized in the left plot of Figure 11 and summarized in Table 1.

Clipping type	NTK matrix	Symmetric NTK	Positive in quadratic form	Positive in eigenvalues	Loss convergence	Monotone loss decay	To zero loss
No clipping	$\mathbf{H} \equiv \sum_r \mathbf{H}_r$	✓	✓	✓	✓	✓	✓
Large R clipping	$\mathbf{H} \equiv \sum_r \mathbf{H}_r$	✓	✓	✓	✓	✓	✓
Small R clipping (Flat)	$\mathbf{HC}$	✗	✗	✓	✗	✗	✓
Small R clipping (Layerwise)	$\sum_r \mathbf{H}_r\mathbf{C}_r$	✗	✗	✗	✗	✗	✗

Table 1: Effects of per-sample clippings on DP gradient flow. Here "Yes/No" means guaranteed or not and the loss refers to the training set. "Loss convergence" is conditioned on $\mathbf{H}(t) \succ 0$.

However, too large R also means too much noise (proportional to σR) in the gradient descent (4.1), hence worsens the accuracy in practice (see Figure 7 (right)). Hence, we demonstrate the difference between global clipping and local clipping when R is moderately large. Using the setting in Section 6.2, we empirically observe that global clipping has better convergence than local clipping at the same clipping norm $R = 75$.

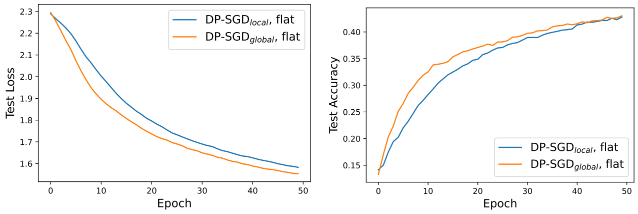

Figure 4: Test performance of global and local clippings on CIFAR10.

5 Privacy Analysis of DP Optimizers

In this section we define DP mathematically and prove that DP optimizers using the global clipping have the **same privacy** guarantee as those using the local clipping. Notice that for the privacy analysis, we work with the general DP optimizers, including those with mini-batches.

Definition 5.1. A randomized algorithm M is (ε, δ)-differentially private (DP) if for any neighboring datasets S, S' differ by an arbitrary sample, and for any event E,

$$\mathbb{P}[M(S) \in E] \leqslant \mathrm{e}^{\varepsilon} \mathbb{P}\left[M\left(S'\right) \in E\right] + \delta. \tag{5.1}$$

A common approach to guarantee DP when approximating a function g is via additive noise calibrated to g's sensitivity Dwork et al. (2006). This is known as the Gaussian mechanism and widely used in DP deep learning.

Lemma 5.2 (Theorem A.1 Dwork et al. (2014); Theorem 2.7 Dong et al. (2019))**.** *Define the* ℓ_2 ***sensitivity*** *of any function* g *to be* $\Delta g = \sup_{S,S'} \|g(S) - g(S')\|_2$ *where the supreme is over all neighboring* (S, S')*. Then the* ***Gaussian mechanism*** $\hat{g}(S) = g(S) + \sigma \Delta g \cdot \mathcal{N}(0, \mathbf{I})$ *is* (ϵ, δ)*-DP for some* ϵ *depending on* (σ, n, p, δ)*.*

For the same differentially private mechanism, different privacy accountants (e.g., Moments accountant Abadi et al. (2016); Canonne et al. (2020), Gaussian differential privacy (GDP) Dong et al. (2019); Bu et al. (2019), Fourier accountant Koskela et al. (2020)) accumulate the privacy risk $\epsilon(\sigma, n, p, \delta, T)$ differently over T iterations. The next result shows that using global clipping is as private as using local clipping, independent of the choice of the privacy accountant.

Theorem 3. *DP optimizers with the local or global clipping are equally* (ϵ, δ)*-DP.*

While a DP model by definition is resilient to all types of privacy attacks, we illustrate that DP-SGD$_{global}$ offers similar privacy protection to DP-SGD$_{local}$ against the membership inference attacks (MIA) in Figure 5. MIA is a common privacy attack by which the attacker aims to determine whether a given data point belongs to the sensitive training set Shokri et al. (2017). In our setting, the black-box attacker uses logistic regression with only access to the prediction logits and labels. The privacy vulnerability is characterized as the attack model's AUC (lower is preferred).

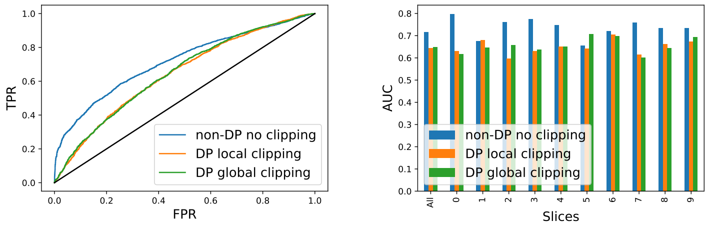

Figure 5: Attack model's ROC-AUC on CIFAR10 in Section 6.2. Left: non-DP overall AUC, 0.717; DP-SGD$_{local}$, 0.644; DP-SGD$_{global}$, 0.648. Right: AUC on subsets of samples by label classes.

6 Numerical Results

We highlight that the global clipping works with any DP optimizers (e.g., DP-Adam, DP-RMSprop, DP-FTRLKairouz et al. (2021), DP-SGD-JLBu et al. (2021a), etc.) that employ the local clipping, with *no additional computational complexity* (discussed in Appendix D). Empirically, DP optimizers with global clipping improve over existing DP optimizers on the convergence of training and generalization losses. We thus reveal a novel phenomenon that DP optimizers play important roles in producing well-calibrated and reliable models.

In M-class classification problems, we denote the probability prediction for the i-th sample as $\boldsymbol{\pi}_i \in \mathbb{R}^M$ so that $f(\boldsymbol{x}_i) = \text{argmax}(\boldsymbol{\pi}_i)$, then the accuracy is $\mathbf{1}\{f(\boldsymbol{x}_i) = y_i\}$. The confidence, i.e., the probability associated with the predicted class, is $\hat{P}_i := \max_{k=1}^M [\boldsymbol{\pi}_i]_k$ and a good calibration means the confidence is close to the accuracy[4]. Formally, we employ three popular calibration metrics from Naeini et al. (2015): the test loss, i.e. the negative log-likelihood (NLL), the Expected Calibration Error (ECE), and the Maximum Calibration Error (MCE):

$$\mathbb{E}_{\hat{P}_i}\left[\left|\mathbb{P}(f(\boldsymbol{x}_i) = y_i|\hat{P}_i = p) - p\right|\right], \quad \max_{p\in[0,1]}\left|\mathbb{P}(f(\boldsymbol{x}_i) = y_i|\hat{P}_i = p) - p\right|.$$

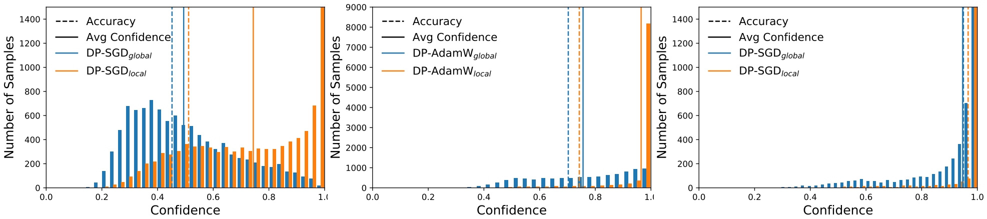

Figure 6: Confidence histograms on CIFAR 10 (left), SNLI (middle), and MNIST (right).

	ECE %			MCE %		
	non-DP	DP local	DP global	non-DP	DP local	DP global
CIFAR10	13.9	20.0	3.3	20.9	32.0	9.9
SNLI	13.0	22.0	17.6*	34.7	62.5	28.9*
MNIST	0.8	2.5	0.5	21.1	50.2	22.8

Table 2: Calibration metrics ECE and MCE by non-DP (no clipping) and DP optimizers. *Note that the SNLI's DP global indeed uses the mix-up training described in Section 6.3.

6.1 MNIST image data with CNN model

On the MNIST dataset, which contains 60000 training samples and 10000 test samples of 28×28 grayscale images in 10 classes, we use the standard CNN in the DP libraries[5]Google; Facebook (see Appendix E.1 for architecture) and train with DP-SGD. In Figure 7, both clippings result in $(2.32, 10^{-5})$-DP, similar test accuracy (96% for local and 95% for global), though the global clipping leads to smaller loss (or NLL). In Figure 7, we demonstrate how R affects the performance of global clipping, ceteris paribus.

In Figure 8, the *reliability diagram* DeGroot & Fienberg (1983); Niculescu-Mizil & Caruana (2005) displays the accuracy as a function of confidence. Graphically speaking, a calibrated classifier is expected to have blue bins close to the diagonal black dotted line. While the non-DP model is

[4] An over-confident classifier, when predicting wrong at one data point, only reduces its accuracy a little but increases its loss significantly due to large $-\log(\pi_{y_i})$, since too little probability is assigned to the true class.

[5] See `https://github.com/tensorflow/privacy/tree/master/tutorials` in Tensorflow and `https://github.com/pytorch/opacus/blob/master/examples` in Pytorch.

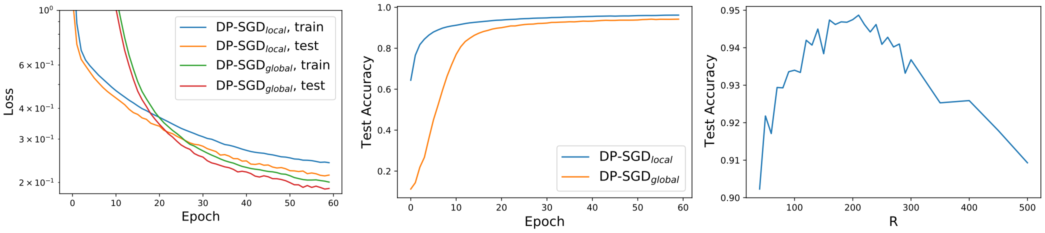

Figure 7: Loss (left) and accuracy (right) on MNIST with 4-layer CNN under batch size 256, $\sigma = 1.1, \eta_{local} = 0.15, R_{local} = 1, \eta_{global} = 0.0007, R_{global} = 210, (\epsilon, \delta) = (2.32, 10^{-5})$.

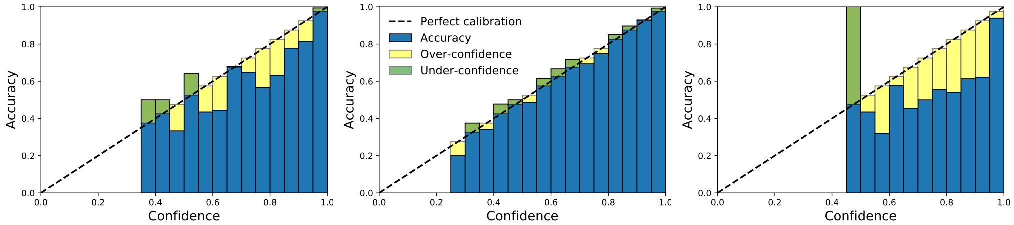

Figure 8: Reliability diagrams for non-DP (left), global (middle), local clipping(right) on MNIST.

generally over-confident and thus not calibrated, the global clipping effectively achieves nearly perfect calibration, thanks to its Bayesian learning nature. In contrast, the classifier with local clipping is not only mis-calibrated, but also falls into 'bipolar disorder': it is either over-confident and inaccurate, or under-confident but highly accurate. This disorder is observed to different extent in all classification experiments in this paper.

6.2 CIFAR10 IMAGE DATA WITH CNN MODEL

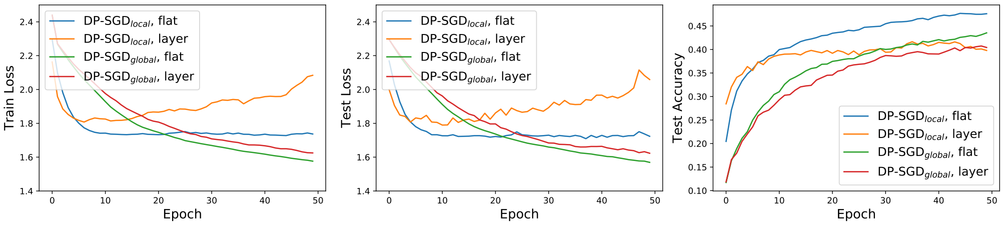

Figure 9: Loss (left and middle) and accuracy (right) on CIFAR10 with 5-layer CNN under batch size 250, $\sigma = 1.3$, $(\epsilon, \delta) = (1.96, 10^{-5})$. For flat clipping, $R_{global} = 75, \eta_{global} = 0.0007, R_{local} = 1.5, \eta_{local} = 0.05$. For layerwise clipping, $R_{global} = 1.5$ for weights, 0.3 for biases; $R_{local} = 1.5$ for weights and biases.

CIFAR10 is a more challenging image dataset, containing 50000/10000 training/test samples of 32×32 color images in 10 classes. We use the CNN on Pytorch CIFAR10 tutorial[6] (see Appendix E.2 for architecture) and train with DP-SGD without pre-training (unlike Abadi et al. (2016); Xu et al. (2020), which pretrain on CIFAR100). Both clippings result in $(1.96, 10^{-5})$-DP and the test accuracy (local: 47.6%; global: 43.5%; non-DP: 61.3%) is comparable with state-of-the-art in Papernot et al. (2020), which is around 47% at this privacy budget. Clearly from Figure 9, global clipping has better convergence and similar accuracy than local clipping. Especially, local layerwise clipping can be unstable, as indicated by Theorem 1. Notice that for classification tasks, the inconsistency between the optimization loss (cross-entropy) and the performance measure (accuracy) is not uncommon and even exaggerated in Section 6.3.

As indicated by the higher losses or NLL, the *confidence histogram* in Figure 14 shows the distribution of prediction confidence and validates that DP-SGD$_{local}$ results in poorly calibrated classifiers on CIFAR10 (i.e., its 75.3% confidence is significantly higher than the actual 47.6% accuracy) while DP-SGD$_{global}$ is much more well-calibrated.

[6] See `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`.

6.3 SNLI TEXT DATA WITH BERT AND MIX-UP TRAINING

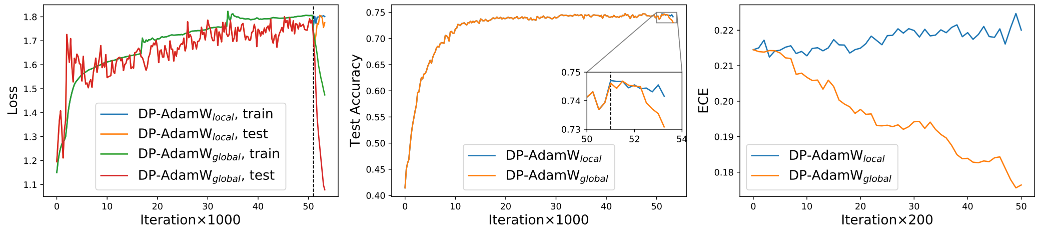

Figure 10: Loss (left), accuracy (middle) and calibration after switching clipping (right) on SNLI with pre-trained BERT, batch size 32, $\eta = 0.0005, \sigma = 0.4, R = 0.1, (\epsilon, \delta) = (1.25, 1/550152)$.

Stanford Natural Language Inference (SNLI) [7] is a collection of human-written English sentence paired with one of three classes: entailment, contradiction, or neutral. The dataset has 550152 training samples and 10000 test samples. We use the pre-trained BERT (Bidirectional Encoder Representations from Transformers) on `Opacus` tutorial[8], which gives a state-of-the-art privacy-accuracy result. Our BERT contains 108M parameters and we only train the last Transformer encoder, which has 7M parameters, using DP-AdamW. In particular, we use a **mix-up training**: for global clipping, we in fact train BERT with DP-SGD$_{local}$ for 3 epochs (51.5×10^3 iterations) and then use DP-SGD$_{global}$ for an additional 2500 iterations. In other words, 95% of the training is done with local clipping but the last 5% is done with global clipping. For local clipping, DP-SGD$_{local}$ is used for the entire training process of 54076 iterations.

Surprisingly, the existing DP optimizer does not minimize the loss at all, yet the accuracy still improves along the training. We again observe that global clipping has significantly better convergence than the local clipping (observe that when turned to global clipping in the last 2500 iterations, the test loss or NLL decreases significantly from 1.79 to 1.08; while keeping the local clipping does not reduce the losses). The resulting global model also has similar accuracy (local: 74.1%; global: 73.1%; as a benchmark, non-DP: 85.4%), same privacy guarantee, and much better calibration in comparison to the local clipping (see Figure 16 and Table 2). We remark that all hyperparameters are exactly the same as in the `Opacus` tutorial.

6.4 REGRESSION TASKS

On regression tasks, the performance measure and the loss function are unified as MSE. Figure 11 shows that global clipping is comparable if not better than local clipping. We experiment on the Wine Quality (1599 samples, 11 features) and California Housing data (20640 samples, 8 features). Additional experimental details are available in Appendix E.4.

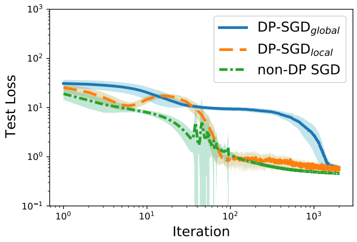

Figure 11: Performance of DP-GD on the Wine Quality dataset.

7 DISCUSSION

In this paper, we establish a continuous dynamics for DP deep learning, based on the NTK matrix, that applies to general neural network architectures, loss functions, and optimization algorithms. We show that in the continuous time, the noise only affects the privacy but not the convergence, whereas the per-sample clipping only affects the convergence but not the privacy. We then propose the global clipping method, as an alternative to the existing local clipping. Hence, one may apply two clippings interchangeably during the training – a strategy we refer to as mix-up training. With large clipping norm, our global clipping significantly outperforms the local clipping by obtaining lower loss and better calibration, while preserving comparable prediction accuracy. Our study sheds light on how the clipping method can fundamentally change the behavior of DP learning, thus encouraging future designs in this direction.

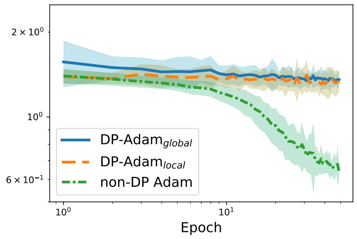

Figure 12: Performance of DP-Adam on California Housing dataset.

[7] We use SNLI 1.0 from `https://nlp.stanford.edu/projects/snli/`.

[8] See `https://github.com/pytorch/opacus/blob/master/tutorials/building_text_classifier.ipynb`.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via overparameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019b.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, pp. 15453–15462, 2019.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *arXiv preprint arXiv:1911.11607*, 2019.

Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Judy Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via jl projections. *arXiv preprint arXiv:2102.03013*, 2021a.

Zhiqi Bu, Shiyun Xu, and Kan Chen. A dynamical view on optimization algorithms of overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 3187–3195. PMLR, 2021b.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. *arXiv preprint arXiv:2206.07136*, 2022.

Carole Cadwalladr and Emma Graham-Harrison. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The guardian*, 17:22, 2018.

Clément Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.

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. *arXiv preprint arXiv:2012.07805*, 2020.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

André Belotto da Silva and Maxime Gazeau. A general system of differential equations to model first-order adaptive algorithms. *Journal of Machine Learning Research*, 21(129):1–42, 2020.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1–5, 2013.

Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. 2014.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438. IEEE, 2013.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pp. 1–19. Springer, 2008.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

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

Facebook. Pytorch Privacy library — Opacus.
`https://github.com/pytorch/opacus`.

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *arXiv preprint arXiv:2010.15110*, 2020.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Google. Tensorflow Privacy library.
`https://github.com/tensorflow/privacy`.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. *arXiv preprint arXiv:2103.00039*, 2021.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pp. 2560–2569. PMLR, 2020.

Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient mcmc and differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 557–566. PMLR, 2019.

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.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632, 2005.

Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA l. Rev.*, 57:1701, 2009.

Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *arXiv preprint arXiv:2007.14191*, 2020.

NhatHai Phan, Xintao Wu, Han Hu, and Dejing Dou. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 385–394. IEEE, 2017.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1): 1–9, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pp. 2493–2502. PMLR, 2015.

Zhiying Xu, Shuyu Shi, Alex X Liu, Jun Zhao, and Lin Chen. An adaptive and fast convergent approach to differentially private deep learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1867–1876. IEEE, 2020.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.