Differentially Private Sharpness-Aware Training

Jinseong Park [1] **Hoki Kim** [1 2] **Yujin Choi** [1] **Jaewook Lee** [1]

Abstract

Training deep learning models with differential privacy (DP) results in a degradation of performance. The training dynamics of models with DP show a significant difference from standard training, whereas understanding the geometric properties of private learning remains largely unexplored. In this paper, we investigate sharpness, a key factor in achieving better generalization, in private learning. We show that flat minima can help reduce the negative effects of per-example gradient clipping and the addition of Gaussian noise. We then verify the effectiveness of Sharpness-Aware Minimization (SAM) for seeking flat minima in private learning. However, we also discover that SAM is detrimental to the privacy budget and computational time due to its two-step optimization. Thus, we propose a new sharpness-aware training method that mitigates the privacy-optimization trade-off. Our experimental results demonstrate that the proposed method improves the performance of deep learning models with DP from both scratch and fine-tuning. Code is available at https://github.com/jinseongP/DPSAT.

1. Introduction

Deep learning models are known to have a risk of privacy leakage (Zhu et al., 2019). To protect the training data from potential data exposure, differential privacy (DP) (Dwork, 2006) provides a mathematical guarantee against adversaries. Nevertheless, training deep learning models with differential privacy (DP training) can result in a degradation of prediction performance compared to models without differential privacy (non-DP training) (Dwork et al., 2014; Abadi et al., 2016; Park et al., 2023).

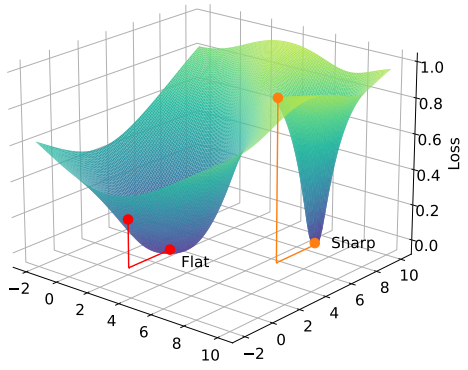


Figure 1. Illustration of flat and sharp minima. The flat minimum is robust to the sharp one given the same size of perturbation in DP training.

DP-SGD (Abadi et al., 2016) is the most popular algorithm for ensuring privacy in deep learning. The primary factors of accuracy drop in DP-SGD are known as **per-example gradient clipping** and **the addition of Gaussian noise**. To mitigate these effects and improve the performance of DP-SGD, many algorithmic solutions concentrate on finding the proper settings for private learning, i.e., different architectures (Tramer & Boneh, 2021; Cheng et al., 2022), loss functions (Shamsabadi & Papernot, 2023), activation functions (Papernot et al., 2021), and clipping functions (Andrew et al., 2021; Bu et al., 2021). However, achieving the ideal performance under private learning still remains an open question.

In this paper, we aim to answer *"how can we find a better optimum in DP training?"* Recently, in deep learning society, it is well known that finding flat minima is a key factor for improving generalization performance (Keskar et al., 2017; Foret et al., 2020). Figure 1 illustrates the importance of flat minimum for DP training. It shows that flat minimum exhibits robustness to the random perturbations, a fundamental idea of protecting data in DP methods. Furthermore, we show that the flatness of the loss landscape can reduce the negative influences of clipping and noise addition.

[1]Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea. [2]Institute of Engineering Research, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea. Correspondence to: Jaewook Lee <jaewook@snu.ac.kr>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Subsequently, we explore the effectiveness of optimization strategies for finding flat minima in private learning, particularly Sharpness-Aware Minimization (SAM) (Foret et al., 2020). SAM, the state-of-the-art optimization method in various domains (Bahri et al., 2021; Qu et al., 2022), efficiently finds flat minima by solving the min-max objective in two steps. Despite this, we point out that the two-step optimization of SAM may be detrimental to the privacy budget and computational time. Based on these observations, we propose a new sharpness-aware training without additional privacy and computational overheads, which successfully mitigates the privacy-optimization trade-off.

Our main contributions are summarized as follows:

- We demonstrate that finding flat minima can reduce the detrimental effects of clipping and noise addition in private learning.
- We show the effectiveness of SAM to seek flat minima and present its drawbacks in private learning. To the best of our knowledge, this is the first attempt to study sharpness-aware training in private learning.
- We propose *Differentially Private Sharpness-Aware Training (DP-SAT)* which makes use of sharpness-aware training without additional privacy costs, achieving both generalization and time efficiency.

This paper is structured as follows: Section 2 introduces related works on DP and flat minima. Section 3 investigates how flatness helps private learning. Section 4.1 adapts SAM to DP and evaluates the problems caused by the two-step updates of SAM. Section 4.2 introduces DP-SAT, a new while sharpness-aware training method for DP. Section 5 empirically demonstrates the effectiveness of DP-SAT across various datasets and tasks. Section 6 demonstrates limitations and future works, and Section 7 concludes the paper.

2. Background and Related Work

2.1. Differentially Private Deep Learning

Differential privacy (DP) (Dwork et al., 2014) provides a formal mathematical framework to guarantee the privacy of training data. It is defined as follows:

Definition 2.1. (Differential privacy) A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-DP, if for two adjacent inputs $d, d' \in \mathcal{D}$ and for any set of possible outputs $\mathcal{S} \subseteq \mathcal{R}$ it holds that

$$Pr[\mathcal{M}(d) \in \mathcal{S}] \leq e^{\varepsilon} Pr[\mathcal{M}(d') \in \mathcal{S}] + \delta. \tag{1}$$

The parameter $\varepsilon \geq 0$ represents the privacy budget with the broken probability $\delta \geq 0$. A smaller value of ε implies a strong privacy guarantee of mechanism $\mathcal{M}$. In the context of deep learning, DP-SGD (Abadi et al., 2016) calculates the per-sample gradient $\nabla \ell_i(\boldsymbol{w})$, where ℓ_i is the per-data loss function of the individual data sample $\boldsymbol{x}_i$ and $\boldsymbol{w}$ is the model parameter. After that, it clips each per-sample gradient to a fixed L_2-norm and then adds Gaussian noise to the average of clipped gradients. In summary, the model weights $\boldsymbol{w}_t$ are updated as $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$ at step t, where the modified gradients $\boldsymbol{g}_t$ for DP-SGD is calculated within mini-batch I_t as follows:

$$\bar{\boldsymbol{g}}_t = \frac{1}{|I_t|} \sum_{i \in I_t} \text{clip}(\nabla \ell_i(\boldsymbol{w}_t), C), \tag{2}$$

$$\boldsymbol{g}_t = \bar{\boldsymbol{g}}_t + \mathcal{N}(\mathbf{0}, C^2 \sigma^2 \mathbf{I}), \tag{3}$$

where $\text{clip}(\boldsymbol{u}, C)$ projects $\boldsymbol{u}$ to the L_2-ball with radius C and vector norm $\| \cdot \|$ indicates the L_2-norm $\| \cdot \|_2$. The noise level σ is determined by the privacy budget (ε, δ) as follows:

Proposition 2.2. *(Abadi et al. (2016)). There exist constant c_1 and c_2 so that given total steps T and sampling probability q, for any $\varepsilon < c_1 q^2 T$, DP-SGD (3) guarantee (ε, δ)-DP, for any $\delta > 0$ if we choose*

$$\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}. \tag{4}$$

As gradient clipping limits the sensitivity of average gradients, DP-SGD can impede the model from updating towards the dominant gradient direction (Papernot et al., 2021). Moreover, the addition of noise to guarantee the privacy of training data can interrupt the convergence of the model weights to the optimum (Yu et al., 2021a). Note that using larger clipping values increases alignment with the original gradients, but also increases the variance at the same time. The additional definitions and properties of DP are summarized in Appendix A.

2.2. Flat Minima and Sharpness-Aware Minimization

Sharp and flat minima Understanding the geometric properties of the loss landscape is a central topic for optimization in deep learning. Generally, the Hessian matrix of loss function $\mathbf{H} = \mathbf{H}_{\boldsymbol{w}} := \nabla^2 \ell(\boldsymbol{w})$ and its *sharpness*, which is defined as its spectral norm $\|\mathbf{H}\|_2$ (or its top eigenvalue λ_{max}), can explain the training dynamics of gradient descent (Cohen et al., 2021). In other words, the loss landscape in the vicinity of a flat minimum, which has small eigenvalues of the Hessian, exhibits slow variation within a neighborhood of $\boldsymbol{w}$. Conversely, near a sharp minimum with large eigenvalues, the loss function is vulnerable to small noises (Li et al., 2018; Keskar et al., 2017; Dinh et al., 2017), even adversarial perturbations (Wu et al., 2020; Lee

et al., 2021; Kim et al., 2023c). To find flatter minima, various optimization techniques have been proposed, such as stochastic weight averaging (Izmailov et al., 2018) and gradient regularizer (Barrett & Dherin, 2020).

Sharpness-aware training Recently, Foret et al. (2020) proposed Sharpness-Aware Minimization (SAM), which is the state-of-the-art optimization methodology in various domains. SAM minimizes the worst-case perturbations within a radius ρ in the vicinity of the parameter space as follows:

$$\min_{\boldsymbol{w}} \max_{\|\boldsymbol{\delta}^*\| \leq \rho} \ell(\boldsymbol{w} + \boldsymbol{\delta}^*). \tag{5}$$

As it is difficult to identify the optimal direction $\boldsymbol{\delta}^*$, SAM approximates the perturbation $\boldsymbol{\delta}$ with a first-order Taylor expansion. Subsequently, SAM updates the model weights in two steps, described as follows:

$$\boldsymbol{w}_t^p = \boldsymbol{w}_t + \rho \boldsymbol{\delta}_t = \boldsymbol{w}_t + \rho \frac{\nabla \ell(\boldsymbol{w}_t)}{\|\nabla \ell(\boldsymbol{w}_t)\|}, \tag{6}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla \ell(\boldsymbol{w}_t^p). \tag{7}$$

We initially calculate the *perturbed weight* $\boldsymbol{w}_t^p$ in the ascent step (6), and then update the model weights towards the gradient of perturbed loss $\nabla \ell(\boldsymbol{w}_t^p)$ in the descent step (7). Foret et al. (2020) defined the difference $\ell(\boldsymbol{w}_t^p) - \ell(\boldsymbol{w}_t)$ as *estimated sharpness* which should be minimized to find flat minima. Note that variants of SAM have been proposed recently to boost the generalization performance (Kwon et al., 2021; Zhuang et al., 2021; Kim et al., 2023a;b) and reduce the computation of two-step optimization (Du et al., 2022a;b; Park et al., 2022).

2.3. Loss Landscape of Private Learning

Recent studies have investigated the unique loss landscape and training dynamics of DP-SGD in comparison to SGD. Bu et al. (2021) analyzed the convergence of DP training in terms of different clipping methods and noise addition. Wang et al. (2021) first highlighted the problem of DP-SGD being stuck in local minima due to the training instability. The authors suggested that averaging the gradients of neighborhoods in the parameter space can achieve a smoother loss landscape and improved performance, yet this comes with a significant computational cost. Instead of averaging, Shamsabadi & Papernot (2023) proposed that loss functions with smaller norm can reduce the impact of clipping and thus create a smoother loss function.

3. Flat Minima Help Private Learning

In this section, we first prove that achieving flatness can be beneficial to DP training. Recently, discovering the relationship between the loss function, gradient norm, and flat minima has become an important topic to analyze in deep learning optimization (Zhao et al., 2022; Zhang et al., 2023). The interaction between these factors significantly influences generalization performance in various domains (Barrett & Dherin, 2020; Wu et al., 2020). Thus, we start by investigating the learning dynamics of DP-SGD, as stated in (Bagdasaryan et al., 2019; Wang et al., 2021). Figure 2 illustrates the training loss and test accuracy of DP-SGD with various privacy budgets ε, SGD only with clipping (SGD+clip), and standard SGD. The training loss of DP-SGD cannot converge to zero due to the effects of clipping and noise addition. This leads to instability of training and corresponding lower performance. To mitigate this phenomenon, we investigate the vulnerability of DP training to clipping and noise addition in terms of sharpness.

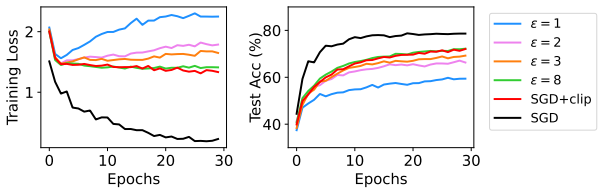


Figure 2. Training loss (left) and test accuracy (right) of DP-SGD with $\varepsilon \in \{1, 2, 3, 8\}$, SGD only with clipping ($C = 0.1$) without noise addition (SGD+clip), and SGD on CIFAR-10.

3.1. Flat Minima Mitigate the Effect of Clipping

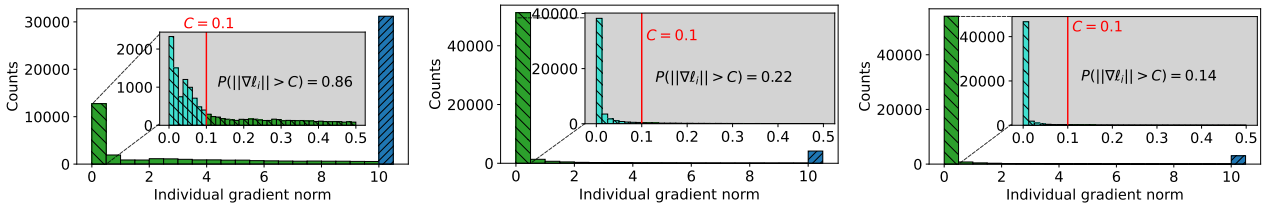
The difference between SGD and SGD+clip in Figure 2 indicates that clipping itself has negative effects in training. It means that reducing the gradient norm can improve performance by avoiding clipping (Papernot et al., 2021). To this end, we argue that a flat minimum exhibits additional advantages in DP training, i.e., reducing the negative impact of clipping by bounding the gradient norm near a local optimum.

Theorem 3.1. ***(Flat minimum mitigates the effect of clipping)*** *The difference between gradients before and after clipping can be bounded by the sharpness as*

$$\begin{aligned} &\|\nabla \ell_i(\boldsymbol{w}) - clip(\nabla \ell_i(\boldsymbol{w}), C)\| \\ =&\mathbb{1}(\|\nabla \ell_i(\boldsymbol{w})\| > C) \cdot (\|\nabla \ell_i(\boldsymbol{w})\| - C) \\ \leq&\mathbb{1}(\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \Delta_{\boldsymbol{w}} > C) \cdot (\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \Delta_{\boldsymbol{w}} - C), \end{aligned}$$

near a local minimum $\boldsymbol{w}^*$, where $\|\mathbf{H}_{\boldsymbol{w}^*}\|_2$ is the sharpness at $\boldsymbol{w}^*$. $\Delta_{\boldsymbol{w}} = \|\boldsymbol{w} - \boldsymbol{w}^*\|$ and $\mathbb{1}$ denotes an indicator function.

We defer the proof to Appendix B.1. As the gradient norm is upper bounded by the sharpness, a lower proportion of gradients is to be clipped within a flat minimum. Empirically, we measure the gradient norm of each data sample $\nabla \ell_i(\boldsymbol{w}_t)$ trained with DP-SGD on the MNIST dataset in Figure 3. The proportion of data samples being clipped



(a) Epoch 1 (Early stage of training) (b) Epoch 10 (c) Epoch 40 (End of training)

Figure 3. Distribution of individual gradient norm $\|\nabla \ell_i\|$ for all i at epoch 1 (early), 10, and 40 (end). We enlarge the values between 0 and 1 for clarifying the distribution of small gradients. The blue chart indicates the sum of all counts larger than 10. The red line indicates the gradient clipping value C. The proportion of being clipped is high at the early stage and decreases constantly.

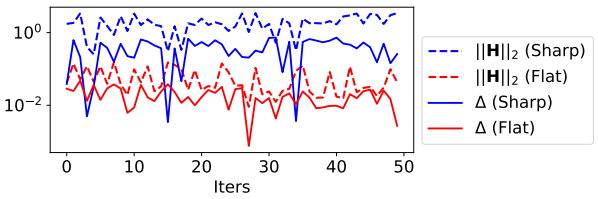


Figure 4. **(dashed line)** Sharpness $\|\mathbf{H}\|_2$ and **(solid line)** the difference of gradients after updating without noise Δ on toy example of Figure 1. The dashed and solid lines in the same color show a positive correlation during the training. Small $\|\mathbf{H}\|_2$ in flat loss surface (red) bounds the gradient difference Δ compared to the sharp one (blue).

$P(\|\nabla \ell_i(\boldsymbol{w})\| > C)$ is high in the early stage of training, indicating that weights are updated in a significantly different direction due to clipping. Even though the proportion $P(\|\nabla \ell_i(\boldsymbol{w})\| > C)$ diminishes as training proceeds, some individual gradients are still clipped. As these clipped gradients act as the primary direction in SGD, it is detrimental to finding an effective direction in DP-SGD. By uncovering a flatter loss landscape, we can reduce the amount of clipped individual gradients in DP-SGD during training.

3.2. Flat Minima Reduce the Bias of Noise

To push further, we investigate the relationship between sharpness and noise addition. Motivated by (Shamsabadi & Papernot, 2023), the following theorem illustrates that flat minima can reduce the error caused by adding Gaussian noise to the clipped gradient in Equation (3).

Theorem 3.2. ***(Flat minimum reduces the bias of noise addition)*** *The difference of gradients between updated without noise $\bar{\boldsymbol{g}}_t$ and with noise $\boldsymbol{g}_t$ is affected by the sharpness as*

$$\|\nabla \ell(\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t) - \nabla \ell(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)\| \leq \eta \max_{\mathbf{H} \in \mathbb{H}} (\|\mathbf{H}\|_2) \cdot \|\boldsymbol{\mu}\|,$$

where $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, C^2\sigma^2\mathbf{I})$ and $\mathbb{H}$ is a set of Hessian matrices $\mathbf{H}$ along the line of $\boldsymbol{\mu}$ from $\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t$ to $\boldsymbol{w}_t - \eta \boldsymbol{g}_t$.

We defer the proof to Appendix B.2. Theorem 3.2 suggests that the sharpness $\|\mathbf{H}\|_2$ can regulate the impact of Gaussian noise in the training, even with the same learning rate η, clipping value C, and noise level σ.

We now empirically validate Theorem 3.2. Revisit Figure 1 illustrating a toy example of a two-dimensional parameter space that comprises one sharp and one flat minimum, suggested in (Wang et al., 2021). Both the sharp and flat minimum have a loss value of 0. Please refer to Appendix C.3 for the details. In Figure 4, we measure the sharpness $\|\mathbf{H}\|_2$ and the gradient difference between updating with and without noise $\Delta_t := \|\nabla \ell(\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t) - \nabla \ell(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)\|$ as training proceeds by gradient descents. To clearly show the effect of noise addition, we select initial points that are equidistant from each minimum and add the same level of noise in each step.

The results show that a positive correlation exists between the sharpness $\|\mathbf{H}\|_2$ (dashed line) and the gradient difference Δ (solid line) during all training epochs, regardless of whether flat (red colored) or sharp (blue colored). More importantly, a flat minimum (red colored) has a lower value of the sharpness $\|\mathbf{H}\|_2$ (dashed line) and thus the gradient difference Δ (solid line), compared to the sharp one (blue colored).

4. Discovering Flat Minima in Private Learning

In the previous section, we show the importance of sharpness in private learning. To seek flat minima, we first demonstrate the effectiveness of SAM in DP training. However, at the same time, we also emphasize the drawbacks of SAM for private learning in terms of privacy budget and computational time. To address these limitations, we propose a new DP-friendly sharpness-aware training algorithm.

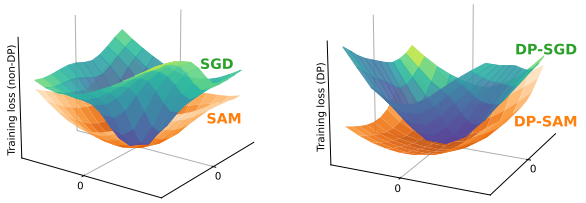
4.1. Challenges of SAM in Private Learning

We introduce the concept of SAM for DP-SGD to achieve flat minima in DP training. One of the main advantages of SAM is the ease of implementation, as it can be applied to various optimizers and architectures by modifying the gradient descent of SGD to a two-step optimization. Thus, we can easily formulate SAM for DP training, referred to as *DP-SAM*, as follows:

$$\boldsymbol{w}_t^p = \boldsymbol{w}_t + \rho \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|}, \tag{8}$$

$$\boldsymbol{g}_t^p = \frac{1}{|I_t|} \sum_{i \in I_t} \text{clip}(\nabla \ell_i(\boldsymbol{w}_t^p), C) + \mathcal{N}(\mathbf{0}, C^2\sigma^2\mathbf{I}), \tag{9}$$

where $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t^p$ and ρ is the radius in the parameter space. In Figure 5, we use the techniques of (Li et al., 2018) to visualize the effectiveness of SAM on the loss landscape for non-DP (left) and DP training (right). Specifically, we perturb a converged minimum to two randomly sampled Gaussian directions and calculate all the losses in grids. The results demonstrate that DP-SAM is more effective than DP-SGD in uncovering flat minima in private learning, which is consistent with the results of standard training with SGD and SAM.



(a) SGD and SAM (b) DP-SGD and DP-SAM

Figure 5. Visualization of loss landscapes.

Nevertheless, we argue that DP-SAM may not be suitable for private learning, despite its ability to discover flat minima. For further details of Equations 8 and 9, we should consider that SAM is a two-step optimization that employs the data samples within the same mini-batch twice. Therefore, we should inject noise into the gradients of the current weight $\boldsymbol{g}_t$ and the perturbed weight $\boldsymbol{g}_t^p$ to make both ascent and descent steps private. This might have two drawbacks, i.e., increased privacy cost and computational burden.

Increased privacy cost Because every query to training data increases the privacy budget, we need to consider the privacy budget of the two-step optimization in Equations (8) and (9). We now demonstrate the privacy guarantee of DP-SAM.

Theorem 4.1. *(Privacy guarantee) DP-SAM requires $(2\varepsilon, 2\delta)$-differential privacy, whereas DP-SGD is (ε, δ)-differential privacy.*

Proof. (Stated informally) For DP-SAM, let $\mathcal{M}_1(d)$ be the ascent step to calculate $\boldsymbol{w}_t^p$ in Equation (8) and $\mathcal{M}_2(d, \mathcal{M}_1(d))$ be the descent step to calculate $\boldsymbol{g}_t^p$ in Equation (9), $\forall d \in \mathcal{D}$. According to Proposition 2.2, $\mathcal{M}_1(d)$ and $\mathcal{M}_2(d)$ satisfy (ε, δ)-DP by clipping individual gradients and injecting noise with σ. Then, by the general composition, DP-SAM requires the addition of two privacy budgets, resulting in $(2\varepsilon, 2\delta)$-DP. The detailed mathematical proof can be found in Appendix B.3. □

Note that the utilization of the moments accountant of Proposition 2.2 (Abadi et al., 2016) or the advanced composition theorem (Dwork et al., 2014) is not feasible for DP-SAM, as these theorems require the random selection of data, known as a *k-fold composition experiment*.

To ensure the same level of privacy as DP-SGD, DP-SAM must satisfy either of the following conditions: $2\sqrt{\frac{\log \delta}{\log 2\delta}}$ (≈ 2.06 when $\delta = 10^{-5}$) times the noise level σ or $\frac{1}{4}\frac{\log 2\delta}{\log \delta}$ (≈ 0.24 when $\delta = 10^{-5}$) times the number of the training iterations compared to DP-SGD. In this paper, we choose to reduce the training time, which usually yields better performance than increasing the noise levels.

Computational overhead The primary weakness of SAM is its computation overhead for implementing two-step optimization (Du et al., 2022b). Furthermore, the computational burden of DP-SAM may impede the use of sharpness-aware training methods because DP training already has a large computational burden (Li et al., 2022).

Briefly, when given m data samples with d-dimensional features, a model size of w, DP-SGD requires $O(mdw)$ for one forward and backward step. However, DP-SAM requires the doubled computations of $O(mdw)$ because it requires two times of gradient computations. Note that clipping operation for DP training is generally $O(mw)$ and storing and recovering weights or individual gradients needs $O(w)$ computation.

4.2. DP-SAT: Differentially Private Sharpness-Aware Training

As aforementioned, DP-SAM yields a privacy-optimization trade-off, which results in a decrease in classification performance while attaining flat minima. The primary problem of DP-SAM is that it requires twice of privacy budget compared to DP-SGD. To maintain the privacy budget consumption of DP-SGD in each iteration, we should abstain from accessing the training data multiple times in a mini-batch.

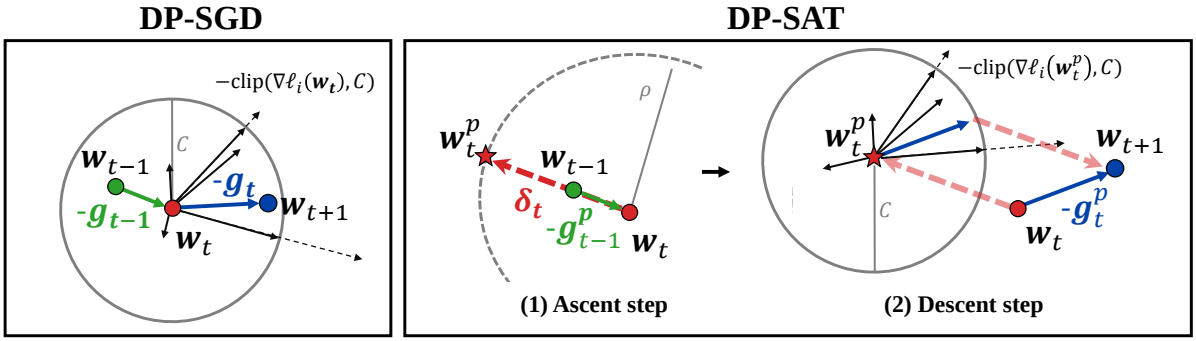


Figure 6. Illustration of DP-SGD and DP-SAT at step t. For ease of understanding, we set the learning rate $\eta = 1$.

Algorithm 1 DP-SAT

Input: Initial parameter $\boldsymbol{w}_0$, learning rate η, radius ρ, clipping threshold C, variance σ^2 from Proposition 2.2, and small τ to prevent zero division.
Output: Final parameter $\boldsymbol{w}_T$.
Initialize: $\boldsymbol{g}_0^p = \mathbf{0}$
for $t = 1, 2, \ldots, T$ **do**
  Construct a random mini-batch I_t
  if *DP-SGD* **then**
    $\boldsymbol{g}_t = \frac{1}{|I_t|}\sum_{i\in I_t} \text{clip}(\nabla\ell_i(\boldsymbol{w}_t), C) + \mathcal{N}(\mathbf{0}, C^2\sigma^2\mathbf{I})$
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\boldsymbol{g}_t$
  if *DP-SAT* **then**
    $\boldsymbol{\delta}_t = \rho\boldsymbol{g}_{t-1}^p/(\|\boldsymbol{g}_{t-1}^p\|+\tau)$ `// Post-processing`
    $\boldsymbol{g}_t^p =$ `// Sharpness-aware training`
    $\frac{1}{|I_t|}\sum_{i\in I_t} \text{clip}(\nabla\ell_i(\boldsymbol{w}_t+\boldsymbol{\delta}_t), C) + \mathcal{N}(\mathbf{0}, C^2\sigma^2\mathbf{I})$
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\boldsymbol{g}_t^p$
end

To achieve this goal, we make use of post-processing (Dwork et al., 2014). Because post-processing guarantees that prior differentially private outputs do not impact the privacy budget, the use of the perturbed gradients is permitted in the earlier steps $\boldsymbol{g}_1^p, \ldots, \boldsymbol{g}_{t-1}^p$ without any cost.

Based on this motivation, we present a new method that re-use the perturbed gradient of the previous step $t-1$ to alter the ascent direction of DP-SAM at the current step t. To be specific, its ascent step can be formulated as follows:

$$\boldsymbol{w}_t^p = \boldsymbol{w}_t + \rho\frac{\boldsymbol{g}_{t-1}^p}{\|\boldsymbol{g}_{t-1}^p\|}. \tag{10}$$

We call this approach *Differentially Private Sharpness-Aware Training (DP-SAT)*. The detailed training procedure of DP-SAT is explained in Algorithm 1 and Figure 6.

Now, we prove that DP-SAT satisfies (ε, δ)-DP, which consumes the same privacy budget as DP-SGD.

Theorem 4.2. *(Privacy guarantee) DP-SAT in Algorithm 1 can guarantee (ε, δ)-differential privacy.*

Table 1. Comparison of sharpness of minima, privacy budget, and computation cost of DP-SGD (Abadi et al., 2016), DP-SAM, and DP-SAT. The privacy budget is estimated by assuming DP-SGD is (ε, δ)-DP. Computational cost is calculated w.r.t. DP-SGD ($1\times$).

Methods	Minima	Privacy budget	Computational cost
DP-SGD	Sharp	(ε, δ)-DP	$1\times$
DP-SAM	Flat	$(2\varepsilon, 2\delta)$-DP	$2\times$ (doubled)
DP-SAT	Flat	(ε, δ)-DP	$1\times$

Proof. (Stated informally) It is sufficient to mention that $\boldsymbol{w}_t^p$ at step t is a result of the post-processing of $\boldsymbol{g}_{t-1}^p$. Then, DP-SAT in Algorithm 1 guarantees (ε, δ)-DP under σ satisfying Proposition 2.2, because it only accesses the training data within the current batch I_t once, which is the same as DP-SGD. The detailed mathematical proof can be found in Appendix B.4. □

Moreover, the ascent step of DP-SAT in Equation (10) does not require additional forward and backward propagation, which requires $O(mdw)$, because it uses the previous weight vector to calculate $\boldsymbol{w}_t^p$, requiring the marginal computation of $O(w)$. In Table 1, we summarize the comparison of sharpness, privacy budgets, and computational costs of DP-SGD, DP-SAM, and DP-SAT.

Higher gradient similarities in DP training The utilization of previous gradients in DP training is facilitated by employing a small clipping value C and a larger batch size, distinguishing it from standard training. This choice leads to increased gradient similarities between the current and previous steps. Nevertheless, the process of finding an appropriate ascent step still poses challenges (Andriushchenko & Flammarion, 2022). A comprehensive explanation is provided in Appendix F.

Momentum variants Recent studies have examined the potential benefits of using momentum variants to enhance

Table 2. Classification accuracy of DP-SGD, DP-SAM, and DP-SAT on the MNIST, FashionMNIST, CIFAR-10, and SVHN datasets. We also report the Error Reduction Rate (ERR) when trained with DP-SAT, in comparison to DP-SGD. We bold the highest average accuracy.

Datasets	Model	Privacy budget ε ($\delta = 10^{-5}$)	Optimizers			ERR
			DP-SGD	DP-SAM	DP-SAT	(%)
MNIST	GNResNet-10 (#Params: 4.90M)	$\varepsilon = 1$	95.15±0.17	92.50±0.44	**96.00±0.21**	17.53%
		$\varepsilon = 2$	96.68±0.27	94.52±0.47	**97.35±0.14**	20.18%
		$\varepsilon = 3$	97.30±0.14	95.62±0.29	**97.83±0.10**	19.63%
	DPNAS-MNIST (#Params: 0.21M)	$\varepsilon = 1$	97.77±0.13	97.21±0.31	**97.96±0.08**	8.52%
		$\varepsilon = 2$	98.60±0.06	97.94±0.20	**98.71±0.09**	7.86%
		$\varepsilon = 3$	98.70±0.12	98.11±0.33	**98.93±0.02**	17.69%
FashionMNIST	GNResNet-10 (#Params: 4.90M)	$\varepsilon = 1$	80.57±0.25	76.73±0.30	**81.33±0.45**	3.91%
		$\varepsilon = 2$	82.71±0.35	79.65±0.53	**84.53±0.41**	10.53%
		$\varepsilon = 3$	84.55±0.17	80.68±0.39	**85.91±0.22**	8.80%
	DPNAS-MNIST (#Params: 0.21M)	$\varepsilon = 1$	84.62±0.19	82.13±0.39	**85.92±0.35**	8.45%
		$\varepsilon = 2$	86.99±0.57	84.21±0.42	**87.75±0.24**	5.84%
		$\varepsilon = 3$	87.97±0.17	84.58±0.56	**88.60±0.04**	5.24%
CIFAR-10	CNN-Tanh with SELU (#Params: 0.55M)	$\varepsilon = 1$	45.24±0.42	44.30±1.16	**45.78±0.48**	0.99%
		$\varepsilon = 2$	56.90±0.33	51.32±0.34	**58.35±0.55**	3.36%
		$\varepsilon = 3$	61.84±0.48	51.81±0.62	**63.51±0.40**	4.38%
	DPNAS-CIFAR10 (#Params: 0.53M)	$\varepsilon = 1$	59.42±0.38	54.00±0.84	**60.13±0.34**	1.75%
		$\varepsilon = 2$	66.30±0.27	60.38±0.46	**67.23±0.12**	2.76%
		$\varepsilon = 3$	68.43±0.43	61.51±0.39	**69.86±0.49**	4.53%
SVHN	DPNAS-CIFAR10 (#Params: 0.53M)	$\varepsilon = 1$	82.25±0.15	80.64±0.34	**83.09±0.54**	4.73%
		$\varepsilon = 2$	86.85±0.33	85.06±0.26	**87.68±0.13**	6.31%
		$\varepsilon = 3$	88.18±0.23	86.24±0.22	**88.74±0.18**	4.74%

generalization and determine the optimal ascent step for SAM (Du et al., 2022a; Park et al., 2023). Similarly, DP-SAT enables the utilization of all the previous step's privatized gradients $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{t-1}$, similar to the aforementioned momentum-based approaches. Our empirical investigation confirms that using the previous gradient alone is sufficient to achieve meaningful performance enhancements, similar to the momentum approach. We believe that the introduction of noise in the DP training process may impede the effective utilization of momentum. We refer the readers to Appendix G.1 for detailed experiments.

5. Experiments

5.1. Experimental Setup

For the empirical results trained from scratch, we evaluate the performance of our method on three commonly used benchmarks for differentially private deep learning: MNIST, FashionMNIST, CIFAR-10, and SVHN. For architecture, we select various architectures: GNResNet-10 (Group Norm ResNet-10) and DPNASNet-MNIST (Cheng et al., 2022) for MNIST and FashionMNIST CNN-Tanh (Papernot et al., 2021) with SELU and DPNASNet-CIFAR (Cheng et al., 2022) for CIFAR-10, and also DPNASNet-CIFAR for SVHN. Particularly, DPNASNet architectures are state-of-the-art architectures with DP-SGD from scratch.

Due to the serious accuracy drop for private learning from scratch, recent studies explore the use of fine-tuning and transfer learning in natural language processing (Yu et al., 2021b; 2022; Li et al., 2022) and computer vision (Bu et al., 2022). For fine-tuning, we evaluate various pre-trained Vision Transformers (ViT), such as ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021), and CrossViT (Chen et al., 2021), with a wide range of model parameters using mixed ghost clipping proposed in (Bu et al., 2022) for CIFAR-10 and CIFAR-100.

The training data for each dataset was partitioned into training and test sets with a ratio of 0.8:0.2, and the test accuracy was averaged over 5 different random seeds for each dataset. All experiments are conducted using the PyTorch-based libraries (Kim, 2020; Yousefpour et al., 2021) with Python on four NVIDIA GeForce RTX 3090 GPUs. Please refer to Appendix C for more details of experimental settings.

5.2. Classification Performance

We conducted a performance comparison of DP-SGD, DP-SAM, and DP-SAT as presented in Table 2. The proposed DP-SAT exhibits superior classification performance compared to DP-SGD in all scenarios, including both small and large models. Specifically, DP-SAT enhances performance under DP-friendly architectures with fewer parameters, including state-of-the-art models such as DPNASNet architectures and CNN-Tanh, achieving an (ERR) of 5.81% on average in these scenarios. Moreover, the performance

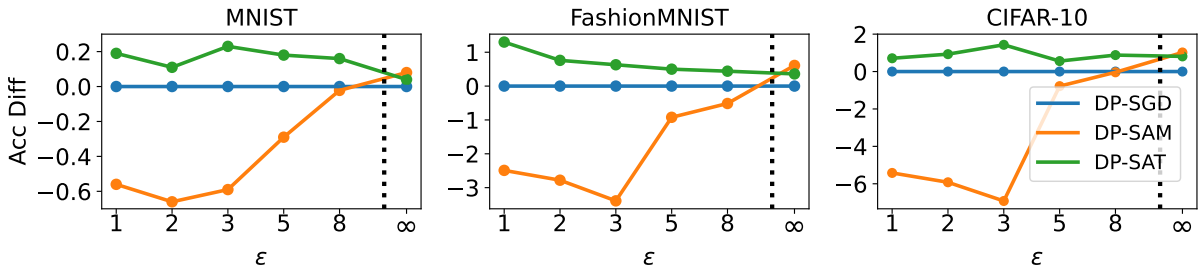


Figure 7. Difference of accuracy for DP-SAM and DP-SAT w.r.t DP-SGD. DPNAS architectures are used. $\varepsilon = \infty$ indicates non-DP settings.

Table 3. Fine-tuning accuracy of DP-SGD and DP-SAT on the CIFAR-10 and CIFAR-100 datasets with various pre-trained models. We also report the Error Reduction Rate (ERR) when trained with DP-SAT, in comparison to DP-SGD. The bold indicates results within the standard deviation of the highest mean score.

Privacy Budget ε ($\delta = 10^{-5}$)	Datasets		CIFAR-10			CIFAR-100		
	Model	#Params	Optimizers		ERR (%)	Optimizers		ERR (%)
			DP-SGD	DP-SAT		DP-SGD	DP-SAT	
$\varepsilon = 0.5$	CrossViT_18_240	42.6M	93.63±0.14	**93.90±0.23**	4.24%	66.15±0.42	**66.97±0.49**	2.42%
	ViT_small_patch16_224	85.8M	89.94±0.19	**90.21±0.20**	2.68%	37.37±1.13	**39.81±1.59**	3.9%
	DeiT_base_patch16_224	85.8M	**92.21±0.24**	**92.15±0.15**	-0.77%	49.25±1.03	**50.03±0.59**	1.54%
$\varepsilon = 2$	CrossViT_tiny_240	6.7M	88.34±0.25	**88.76±0.28**	3.60%	**59.58±0.31**	**59.75±0.40**	0.42%
	CrossViT_small_240	26.3M	93.89±0.18	**94.15±0.28**	4.26%	71.14±0.38	**71.38±0.23**	0.83%
	CrossViT_18_240	42.6M	**95.32±0.12**	**95.31±0.13**	-0.21%	74.29±0.17	**74.52±0.19**	0.89%
	ViT_small_patch16_224	85.8M	92.22±0.46	**92.50±0.27**	3.60%	65.89±0.72	**67.27±0.67**	4.05%
	DeiT_base_patch16_224	85.8M	**94.29±0.18**	**94.44±0.25**	2.63%	**69.20±0.49**	**69.69±0.76**	1.59%

improvements are particularly pronounced in large models, such as GNResNet-10, with an ERR of 13.43% on average. As large models with complicated architectures are known to face challenges in generalizing well in DP settings due to their susceptibility to perturbations (Tramer & Boneh, 2021), the identification of flat minima becomes notably advantageous. Meanwhile, due to the aforementioned privacy consumption, DP-SAM shows lower classification performance than DP-SGD.

To visualize the difference between optimization methods, we plot the accuracy difference with respect to DP-SGD in Figure 7. We tested on various privacy budgets $\varepsilon \in \{1, 2, 3, 5, 8\}$, including the non-DP ($\varepsilon = \infty$) setting. Consistent with prior results, DP-SAT shows the best performance among methods. Interestingly, as the privacy budget ε increases, gradually approaching the non-DP settings, the gap between DP-SAT and DP-SAM diminishes. Thus, it is clear that DP-SAT fully utilizes the positive effects of flatness in DP models by evaluating a broad range of privacy budgets ε.

For ablation studies, we conduct a range of experiments, including a sensitivity analysis on the parameter ρ, a comparison of various base optimizers, and investigations into other relevant factors. We further argue that the accuracy improvement achieved by DP-SAT is not solely reliant on the enlarged hyperparameter search space. The detailed results and analysis can be found in Appendix G.

5.3. Fine-tuning Performance

We now show that the idea of sharpness-aware training is effective in fine-tuning for private models. As the ViT models have well-generalizing latent space, they show higher fine-tuning accuracy than other CNN models. We tested DP-SGD and DP-SAM for various pre-trained ViT models on $\varepsilon = 0.5$ and 2, as shown in Table 3. The experimental results show that DP-SAT outperforms DP-SGD in the majority of fine-tuning cases. Note that the difference in fine-tuning experiments is marginal compared to from-scratch training because of the relatively small training epoch of only 5 epochs.

5.4. Sharpness Analysis

We measure the Eigenspectrum of the Hessian matrix $\mathbf{H}$ of the trained models with DP-SGD and DP-SAT on CIFAR-10 in Figure 8. In DP-SAT, the probability of eigenvalues $p(\lambda)$ is shifted towards the left, which indicates DP-SAT finds flatter minima compared to DP-SGD. In addition, both the sharpness $\lambda_{max} = 130.26$ and the ratio of eigenval-

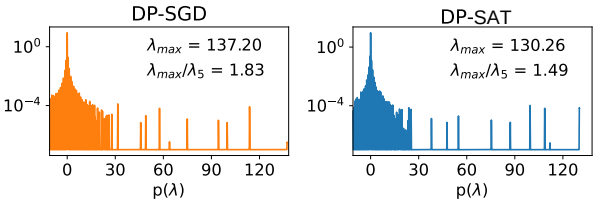
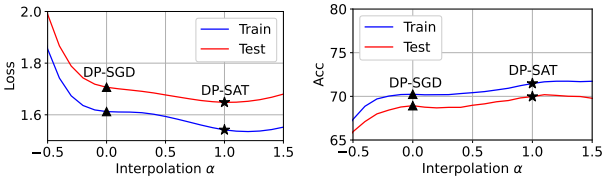


Figure 8. Eigenspectrum of Hessian on CIFAR-10.



(a) Loss (b) Accuracy

Figure 9. Results of linear interpolations $\boldsymbol{w}(\alpha) = (1-\alpha)\boldsymbol{w}+\alpha\boldsymbol{w}'$ (for $\alpha \in [-0.5, 1.5]$) between DP-SGD (▲) and DP-SAT (⋆).

ues $\lambda_{max}/\lambda_5 = 1.49$, which are the popular measures to estimate flat minima, are smaller than those of DP-SGD.

Furthermore, we adopt the idea of (Chen et al., 2022) to explain the success of DP-SAT, which can be explained by the similarities between training and test losses. Higher levels of similarity lead to a smaller generalization gap in performance under flatter local minima. This is a distinct factor contributing to the success of SAM that is separate from the impact of gradient norm. We interpolate the loss surface of models trained using both DP-SGD and DP-SAT in Figure 9. Our results are consistent with well-generalizing settings and demonstrate that: (1) DP training exhibits higher levels of similarity between training and test losses; (2) DP-SAT produces a flatter loss landscape and lower errors in both training and test settings; and (3) the shape of the functions is strongly correlated with generalization performance.

5.5. Computational Efficiency

We empirically show that DP-SAT does not need an additional computational burden for sharpness-aware training in Table 4. The training speed of DP-SGD and DP-SAT is almost the same in all datasets. On the other hand, DP-SAM requires further training times to calculate the ascent direction at every step t. Note that the training speed ratio compared to DP-SGD is twice in non-DP training, but it shows lesser results such as 164.9% or 181.9% because DP training itself needs additional computation, such as individual gradient accumulation and memory access.

Table 4. Training speed (images/sec) on the MNIST, FashionMNIST, and CIFAR-10 (higher is faster). The numbers in parentheses (·) indicate the training speed ratio w.r.t. DP-SGD (lower is faster).

	MNIST	FashionMNIST	CIFAR-10
DP-SGD	5,776	6,020	3,233
DP-SAM	3,503 (164.9%)	3,712 (162.2%)	1,777 (181.9%)
DP-SAT	5,777 (100.0%)	6,022 (100.0%)	3,188 (101.4%)

6. Limitations and Future Work

As the relationship between flat minima and generalization performance is still being actively researched, we hope that further work will be built on our work to explore the advantages of flatness under DP training schemes. First, there is a distinct line of work that attempts to achieve flatness, referred to as weight averaging, which has been actively compared and combined with SAM in recent studies (Kaddour et al., 2022). At present, it appears that weight averaging has difficulty in improving the performance of DP models (Panda et al., 2022); however, we believe that suitable variants of weight averaging may be beneficial for DP training methods from our benchmark results. Second, research into gradient norm regularization under DP schemes could be a beneficial direction. Lastly, we will further investigate the recently proposed variants of SAM in DP training.

7. Conclusion

In this paper, we investigated the geometric properties of private learning, specifically sharpness. We showed that seeking flat minima can mitigate the negative effects of clipping and noise addition during training. However, we also identified that the two-step optimization of SAM may have negative impacts on privacy budget and computational time. To address this issue, we proposed a new sharpness-aware training method that can improve performance without additional privacy or computational burden. We believe that this work will contribute to the understanding of sharpness and optimization in deep learning with differential privacy.

Acknowledgements

This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation; No.2021-0-02068, Artificial Intelligence Innovation Hub) and the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) (No. 2019R1A2C2002358; No. 2022R1A5A600ff0840).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Andrew, G., Thakkar, O., McMahan, H. B., and Ramaswamy, S. Differentially private learning with adaptive clipping. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Andriushchenko, M. and Flammarion, N. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 639–668. PMLR, 2022.

Bagdasaryan, E., Poursaeed, O., and Shmatikov, V. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Bahri, D., Mobahi, H., and Tay, Y. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

Barrett, D. and Dherin, B. Implicit gradient regularization. In *International Conference on Learning Representations*, 2020.

Bu, Z., Wang, H., and Long, Q. On the convergence and calibration of deep learning with differential privacy. *arXiv preprint arXiv:2106.07830*, 2021.

Bu, Z., Mao, J., and Xu, S. Scalable and efficient training of large convolutional neural networks with differential privacy. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=SQbrWcMOcPR.

Chen, C.-F. R., Fan, Q., and Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 357–366, 2021.

Chen, X., Hsieh, C.-J., and Gong, B. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations*, 2022.

Cheng, A., Wang, J., Zhang, X. S., Chen, Q., Wang, P., and Cheng, J. Dpnas: Neural architecture search for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6358–6366, 2022.

Cohen, J., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Du, J., Daquan, Z., Feng, J., Tan, V., and Zhou, J. T. Sharpness-aware training for free. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=xK6wRfL2mv7.

Du, J., Yan, H., Feng, J., Zhou, J. T., Zhen, L., Goh, R. S. M., and Tan, V. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022b.

Dwork, C. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer, 2006.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Izmailov, P., Wilson, A., Podoprikhin, D., Vetrov, D., and Garipov, T. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 876–885, 2018.

Jang, C., Lee, S., Park, F. C., and Noh, Y.-K. A reparametrization-invariant sharpness measure based on information geometry. In *Advances in neural information processing systems*, 2022.

Kaddour, J., Liu, L., Silva, R., and Kusner, M. J. When do flat minima optimizers work? *Advances in Neural Information Processing Systems*, 35:16577–16595, 2022.

Keskar, N. S., Nocedal, J., Tang, P. T. P., Mudigere, D., and Smelyanskiy, M. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Kim, H. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

Kim, H., Park, J., Choi, Y., and Lee, J. Stability analysis of sharpness-aware minimization, 2023a.

Kim, H., Park, J., Choi, Y., Lee, W., and Lee, J. Exploring the effect of multi-step ascent in sharpness-aware minimization. *arXiv preprint arXiv:2302.10181*, 2023b.

Kim, H., Park, J., and Lee, J. Generating transferable adversarial examples for speech classification. *Pattern Recognition*, 137:109286, 2023c.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

Lee, S., Lee, W., Park, J., and Lee, J. Towards better understanding of training certifiably robust models against adversarial examples. *Advances in Neural Information Processing Systems*, 34:953–964, 2021.

Lee, S., Park, J., and Lee, J. Implicit jacobian regularization weighted with impurity of probability output, 2022. URL https://openreview.net/forum?id=RQ3xUXjZWMO.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022.

Panda, A., Tang, X., Sehwag, V., Mahloujifar, S., and Mittal, P. Dp-raft: A differentially private recipe for accelerated fine-tuning. *arXiv preprint arXiv:2212.04486*, 2022.

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

Papyan, V. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In *International conference on machine learning*, volume 97, pp. 5012–5021. PMLR, 2019.

Park, J., Kim, H., Choi, Y., Lee, W., and Lee, J. Fast sharpness-aware minimization for time series classification and forecasting. *Available at SSRN 4346500*, 2022.

Park, J., Choi, Y., Byun, J., Lee, J., and Park, S. Efficient differentially private kernel support vector classifier for multi-class classification. *Information Sciences*, 619:889–907, 2023.

Qu, Z., Li, X., Duan, R., Liu, Y., Tang, B., and Lu, Z. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*. PMLR, 2022.

Sagun, L., Evci, U., Güney, V. U., Dauphin, Y. N., and Bottou, L. Empirical analysis of the hessian of overparametrized neural networks. In *International Conference on Learning Representations (Workshop)*, 2018.

Shamsabadi, A. S. and Papernot, N. Losing less: A loss for differentially private deep learning. *Proceedings on Privacy Enhancing Technologies*, 3:307–320, 2023.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Tramer, F. and Boneh, D. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

Wang, W., Wang, T., Wang, L., Luo, N., Zhou, P., Song, D., and Jia, R. Dplis: Boosting utility of differentially private deep learning via randomized smoothing. *Proceedings on Privacy Enhancing Technologies*, 4:163–183, 2021.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Xie, Z., Wang, X., Zhang, H., Sato, I., and Sugiyama, M. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International Conference on Machine Learning*, pp. 24430–24459. PMLR, 2022.

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

Yu, D., Zhang, H., Chen, W., and Liu, T.-Y. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021a.

Yu, D., Zhang, H., Chen, W., Yin, J., and Liu, T.-Y. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pp. 12208–12218. PMLR, 2021b.

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

Zhang, X., Xu, R., Yu, H., Zou, H., and Cui, P. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20247–20257, 2023.

Zhao, Y., Zhang, H., and Hu, X. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, pp. 26982–26992. PMLR, 2022.

Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N. C., s Duncan, J., Liu, T., et al. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2021.

A. Composition Theorems

We present the basic composition theorems of (ε, δ)-DP algorithms.

General composition theorem (Dwork et al., 2014)

Definition A.1. (General composition theorem for (ε, δ)-DP algorithms) Let $\mathcal{M}_1 : d \mapsto \mathcal{M}_1(d) \in \mathcal{R}_1$ be an (ε, δ)-DP function, and for $k \geq 2$ and $s_j \in \mathcal{R}_j, \forall j \in \{1, \ldots, k-1\}, \mathcal{M}_k : (d, s_1, \ldots, s_{k-1}) \mapsto \mathcal{M}_k(d) \in \mathcal{R}_k$ be (ε, δ)-DP, given the previous outputs $s_1, \ldots, s_{k-1} \in \otimes_{j=1}^{k-1} \mathcal{R}_j$. Then, for all neighboring d, d' and all $\mathcal{S} \subset \otimes_{j=1}^{k} \mathcal{R}_j$,

$$Pr[(\mathcal{M}_1, \ldots, \mathcal{M}_k)(d) \in \mathcal{S}] \leq e^{k\varepsilon} Pr[(\mathcal{M}_1, \ldots, \mathcal{M}_k)(d') \in \mathcal{S}] + k\delta. \tag{11}$$

Note that Equation (11) does not require any assumption of d, which can be used in *DP-SAM*.

Advanced composition theorem (Dwork et al., 2014) From now on, we need strong assumptions on d, i.e., *k-fold composition experiment*, which is the repeated use of differentially private algorithms on different (random sampled) data that may nevertheless contain information of one individual. By assumption on d on each step, the concrete sequences of mechanisms can guarantee a tighter privacy budget than Equation (11).

Definition A.2. (Advanced composition theorem for (ε, δ)-DP algorithms) For all $\varepsilon, \delta, \delta' \geq 0$, the sequence of k-fold (ε, δ)-DP mechanisms satisfies $(\varepsilon', k\delta + \delta')$-DP, where $\varepsilon' = \sqrt{2k \ln(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1)$.

Moments accountant (Abadi et al., 2016) **(restated)** Equation (3) illustrates the weight update at step t as follows:

$$\boldsymbol{g}_t = \frac{1}{|I_t|} \sum_{i \in I_t} \text{clip}(\nabla \ell_i(\boldsymbol{w}_t), C) + \mathcal{N}(\mathbf{0}, C^2\sigma^2),$$
$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t,$$

where $\text{clip}(\boldsymbol{u}, C)$ projects $\boldsymbol{u}$ to the L_2-ball with radius C. Abadi et al. (2016) proved that there exist constant c_1 and c_2 so that given total steps T and sampling probability q, for any $\varepsilon < c_1 q^2 T$, Equation (3) guarantee (ε, δ)-DP, for any $\delta > 0$ if we choose

$$\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}. \tag{12}$$

The composition of moments can reduce the accumulated privacy budget to $(O(q\varepsilon\sqrt{T}), \delta)$-DP. Detailed proof can be found in Appendix B of (Abadi et al., 2016).

Post-processing Post-processing guarantees to use the previous differentially private outputs.

Definition A.3. (Post-processing (Dwork et al., 2014)) If a mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{R}_1$ is (ε, δ)-DP, for any randomized mapping $h : \mathcal{R}_1 \to \mathcal{R}_2$, $h \circ \mathcal{M} : \mathcal{X} \to \mathcal{R}_2$ is at least (ε, δ)-DP.

B. Proofs

B.1. Proof of Theorem 3.1

Proof. With the first-order Taylor expansion of the gradients $\nabla \ell_i(\boldsymbol{w})$ for $\boldsymbol{w}$ near a local minimum,[1]

$$\begin{aligned}
\|\nabla \ell_i(\boldsymbol{w})\| \approx & \|\nabla \ell_i(\boldsymbol{w}^*) + \mathbf{H}_{\boldsymbol{w}^*}^T(\boldsymbol{w} - \boldsymbol{w}^*)\| \\
\leq & \|\nabla \ell_i(\boldsymbol{w}^*)\| + \|\mathbf{H}_{\boldsymbol{w}^*}^T(\boldsymbol{w} - \boldsymbol{w}^*)\| \\
= & \|\mathbf{H}_{\boldsymbol{w}^*}^T(\boldsymbol{w} - \boldsymbol{w}^*)\| \\
& (\because \text{at local minimum } \boldsymbol{w}^*, \|\nabla \ell_i(\boldsymbol{w}^*)\| = 0) \\
\leq & \|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \cdot \|\boldsymbol{w} - \boldsymbol{w}^*\|.
\end{aligned}$$

[1]The second-order Taylor expansion of the loss function (and correspondingly, the first-order Taylor expansion of the gradient) is commonly employed to analyze properties in the vicinity of critical points in deep learning optimization (Zhao et al., 2022; Xie et al., 2022).

Let us define the clip operation as follows:

$$\text{clip}(\nabla \ell_i(\boldsymbol{w}), C) = \nabla \ell_i(\boldsymbol{w}) \cdot \frac{1}{\max(1, \frac{\|\nabla \ell_i(\boldsymbol{w})\|}{C})}.$$

Then,

$$\begin{aligned}
&\|\nabla \ell_i(\boldsymbol{w}) - \text{clip}(\nabla \ell_i(\boldsymbol{w}), C)\| \\
&\quad = \begin{cases} \|\nabla \ell_i(\boldsymbol{w})\| - C & \text{if } \|\nabla \ell_i(\boldsymbol{w})\| > C, \\ 0 & \text{otherwise.} \end{cases} \\
&\quad = \mathbb{1}(\|\nabla \ell_i(\boldsymbol{w})\| > C) \cdot (\|\nabla \ell_i(\boldsymbol{w})\| - C) \\
&\quad \leq \mathbb{1}(\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \cdot \|\boldsymbol{w} - \boldsymbol{w}^*\| > C) \cdot (\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \cdot \|\boldsymbol{w} - \boldsymbol{w}^*\| - C) \\
&\quad = \mathbb{1}(\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \Delta_{\boldsymbol{w}} > C) \cdot (\|\mathbf{H}_{\boldsymbol{w}^*}\|_2 \Delta_{\boldsymbol{w}} - C).
\end{aligned}$$

where $\Delta_{\boldsymbol{w}} = \|\boldsymbol{w} - \boldsymbol{w}^*\|$ and $\mathbb{1}$ denotes an indicator function. □

B.2. Proof of Theorem 3.2

We modify the proof of (Wang et al., 2021; Shamsabadi & Papernot, 2023), which indicates the effect of smoothness in terms of β-smoothness, to illustrate how the sharpness affects the Gaussian noise addition during training.

Proof. Let $\mathbf{H}_{\boldsymbol{w}} := \nabla^2 \ell(\boldsymbol{w})$.

$$\begin{aligned}
\|\nabla \ell(\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t) - \nabla \ell(\boldsymbol{w}_t - \eta \boldsymbol{g}_t)\| &= \eta \|(\int_0^1 \mathbf{H}_{(\boldsymbol{w}_t - \eta \boldsymbol{g}_t) + z\boldsymbol{\mu}} \boldsymbol{\mu} dz)\| \\
&\quad (\because (\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t) - (\boldsymbol{w}_t - \eta \boldsymbol{g}_t) = \eta \boldsymbol{\mu}) \\
&\leq \eta \int_0^1 \|\mathbf{H}_{(\boldsymbol{w}_t - \eta \boldsymbol{g}_t) + z\boldsymbol{\mu}} \boldsymbol{\mu}\| dz \\
&\leq \eta \max_{\mathbf{H} \in \mathbb{H}} (\|\mathbf{H}\|_2) \cdot \|\boldsymbol{\mu}\|
\end{aligned}$$

where $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, C^2 \sigma^2 \mathbf{I})$ and $\mathbb{H}$ is a set of Hessian matrices $\mathbf{H}$ along the line of $\boldsymbol{\mu}$ from $\boldsymbol{w}_t - \eta \bar{\boldsymbol{g}}_t$ to $\boldsymbol{w}_t - \eta \boldsymbol{g}_t$. □

B.3. Proof of Theorem 4.1

Given (ε, δ) privacy budget of DP-SGD, let each privacy budget for the t-th step is $(\varepsilon_t, \delta_t)$, which represents the additional privacy budget for calculating $\boldsymbol{g}_t$. Then, it is enough to show that the t-th update of DP-SAM is $(2\varepsilon_t, 2\delta_t)$-DP.

The t-th update of DP-SAM can be decomposed by two mechanisms:

$$\text{(ascent step) } w_t^p := \mathcal{M}_\infty(w_t, d) = w_t + \rho \frac{\boldsymbol{g}_t}{\|\boldsymbol{g}_t\|} \in \mathcal{R}_{1,t+1}$$

$$\text{(descent step) } w_{t+1} := \mathcal{M}_\in(w_t^p, w_t, d) = w_t - \eta \boldsymbol{g}_t^p \in \mathcal{R}_{2,t+1}$$

where $d \in \mathcal{D}$ and $\boldsymbol{g}_t^p$ defined by Equation (9). Note that the input of each mechanism is differentially private except for input data d. Calculating each $\boldsymbol{g}_t$ and $\boldsymbol{g}_t^p$ consumes the privacy budget of $(\varepsilon_t, \delta_t)$ same as DP-SGD. Therefore, the additional cost of each mechanism is $(\varepsilon_t, \delta_t)$. Then, by the general composition theorem,

$$\mathcal{M}_2 : (w_t^p, w_t, d) \mapsto \mathcal{M}_2(w_t^p, w_t, d) \in \mathcal{R}_{2,t+1}$$

be $(2\varepsilon_t, 2\delta_t)$ -DP for any $d \in \mathcal{D}$, $w_t \in \mathcal{R}_{2,t}$, and $w_t^p \in \mathcal{R}_{1,t+1}$. Therefore, the privacy budget of DP-SAM is $(2\varepsilon, 2\delta)$.

B.4. Proof of Theorem 4.2

Let us consider the same setting as the proof of Theorem 4.1 (presented in Appendix B.3). Then, it is enough to show that the t-th update of DP-SAT is $(\varepsilon_t, \delta_t)$-DP.

The t-th update of DP-SAT can be decomposed by two mechanisms:

$$\text{(ascent step) } w_t^p := \mathcal{M}_\infty([w_t], [w_{t-1}^p]) = w_t + \rho \frac{\boldsymbol{g}_{t-1}^p}{\|\boldsymbol{g}_{t-1}^p\|} \in \mathcal{R}_{1,t+1}$$

$$\text{(descent step) } w_{t+1} := \mathcal{M}_\in([w_t^p], [w_t], d) = w_t - \eta \boldsymbol{g}_t^p \in \mathcal{R}_{2,t+1}$$

where $d \in \mathcal{D}$, $[w_t] = \{w_1, \cdots, w_t\} \in \otimes_{j=1}^t \mathcal{R}_{2,j}$ and $[w_t^p] = \{w_1^p, \cdots, w_t^p\} \in \otimes_{j=1}^t \mathcal{R}_{1,j+1}$ and $\boldsymbol{g}_t^p$ defined by Equation (9). We used $[w_t]$ or $[w_t^p]$ instead of w_t or w_t^p to clearly indicate the accumulation of noise. For example, $\mathcal{M}_2$ requires the same input as Theorem 4.1; (w_t^p, w_t, d).

Then, since $\mathcal{M}_\in([w_{t-1}^p], [w_{t-1}], \cdot) : \mathcal{D} \to \mathcal{R}_{2,t}$ is $(\varepsilon_{t-1}, \delta_{t-1})$-DP mechanism and $\mathcal{M}_\infty(\cdot, [w_{t-1}], [w_{t-1}^p]) : \mathcal{R}_{2,t} \to \mathcal{R}_{1,t+1}$ is a randomized mapping, the ascent step requires no additional privacy budget by post-processing. Moreover, the descent step requires the same privacy budget as DP-SGD, the total additional privacy budget of t-th update is $(\varepsilon_t, \delta_t)$-DP, the same as DP-SGD. Therefore, the privacy budget of DP-SAT is (ε, δ).

C. Experimental settings

C.1. Classification

We use SGD as a base optimizer with a momentum of 0.9 and a learning rate of 2.0, without any learning rate decay, as mentioned in (Cheng et al., 2022). We conducted a hyperparameter search on $\rho = \{0.005, 0.01, 0.02, 0.03, 0.05, 0.1\}$, and the privacy broken probability $\delta = 10^{-5}$ in DP training.

Table 5. Hyperparameters for training on MNIST, FashionMNIST, CIFAR-10, and SVHN.

Dataset		MNIST		FashionMNIST		CIFAR-10		SVHN
Architecture		GNResNet-10	DPNAS-MNIST	GNResNet-10	DPNAS-MNIST	CNN-Tanh with SELU	DPNAS-CIFAR10	DPNAS-CIFAR10
Optimizer		SGD	SGD	SGD	SGD	SGD	SGD	SGD
Epoch		40	40	40	40	30	30	30
Batch size		2048	2048	2048	2048	2048	2048	2048
Learning rate η		2	2	2	2	2	2	2
Momentum β		0.9	0.9	0.9	0.9	0.9	0.9	0.9
Max grad norm C		0.1	0.1	0.1	0.1	0.1	0.1	0.1
Radius ρ	$\varepsilon = 1$	0.03	0.03	0.03	0.05	0.02	0.01	0.05
	$\varepsilon = 2$	0.03	0.03	0.03	0.03	0.1	0.01	0.05
	$\varepsilon = 3$	0.03	0.03	0.03	0.02	0.05	0.01	0.03

Note that the best radius ρ can be varied according to the randomized noise addition of each random seed due to the instability of DP training. For a detailed explanation of DPNAS architectures in (Cheng et al., 2022), please refer to their official GitHub code from `https://github.com/TheSunWillRise/DPNAS`.

C.2. Fine-tuning

We use Adam as a base optimizer with a learning rate of 0.002. We trained the model for 5 epochs with a batch size of 1000 and a mini-batch size of 100. Here, we use a hyperparameter search on a wide range of radius than the classification $\rho = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}$, as we use a smaller learning rate. We use the mixed ghost clipping (Bu et al., 2022) and their official GitHub code from `https://github.com/woodyx218/private_vision`.

Table 6. Radius ρ for fine-tuning on $\varepsilon = 0.5$.

	CrossViT_18_240	ViT_small_patch16_224	DeiT_base_patch16_224
CIFAR-10	0.05	0.1	0.1
CIFAR-100	0.05	0.2	0.2

Table 7. Radius ρ for fine-tuning on $\varepsilon = 2$.

	CrossViT_tiny_240	CrossViT_small_240	CrossViT_18_240	ViT_small_patch16_224	DeiT_base_patch16_224
CIFAR-10	0.05	1	0.5	0.001	0.05
CIFAR-100	0.001	0.001	0.01	0.001	0.05

C.3. Toy Example of Figure 1

Figure 4 illustrates the simple mixture of flat and sharp minima to investigate the effect of sharpness. Following (Wang et al., 2021), we generate the example as follows:

$$\min_{\boldsymbol{w}} \left[\mathcal{F}_{\boldsymbol{c}_1}(\boldsymbol{w}) \cdot \frac{\phi(\boldsymbol{w}, \boldsymbol{c}_1)}{\phi(\boldsymbol{w}, \boldsymbol{c}_1) + \phi(\boldsymbol{w}, \boldsymbol{c}_2)} + \mathcal{F}_{\boldsymbol{c}_2}(\boldsymbol{w}) \cdot \frac{\phi(\boldsymbol{w}, \boldsymbol{c}_2)}{\phi(\boldsymbol{w}, \boldsymbol{c}_1) + \phi(\boldsymbol{w}, \boldsymbol{c}_2)} \right]$$

where $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathbb{R}^2$ are two fixed centers for flat and sharp minima, respectively. We used $\boldsymbol{c}_1 = [2.5, 2.5]$ and $\boldsymbol{c}_2 = [7.5, 7.5]$. $\phi(\boldsymbol{w}, \boldsymbol{c}) = e^{-\|\boldsymbol{w}-\boldsymbol{c}\|}$, $\mathcal{F}_{\boldsymbol{c}_1}(\boldsymbol{w}) = \texttt{Sigmoid}\left(\frac{\|\boldsymbol{w}-\boldsymbol{c}_1\|}{5} - \frac{5}{\|\boldsymbol{w}-\boldsymbol{c}_1\|}\right)$, and $\mathcal{F}_{\boldsymbol{c}_2}(\boldsymbol{w}) = \texttt{Sigmoid}\left(\frac{5\|\boldsymbol{w}-\boldsymbol{c}_1\|}{5} - \frac{5}{5\|\boldsymbol{w}-\boldsymbol{c}_1\|}\right)$.

D. Illustration of DP-SGD, DP-SAM, and DP-SAT

We illustrate the training methods of DP-SGD, DP-SAM, and DP-SAT in Figure 10.

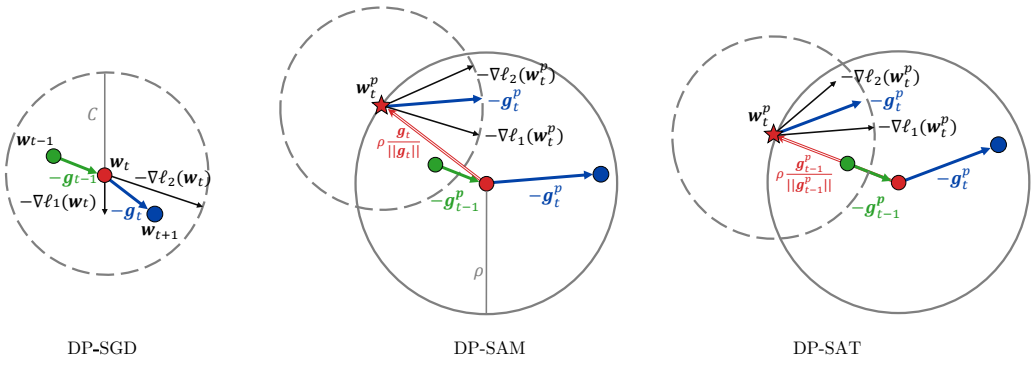
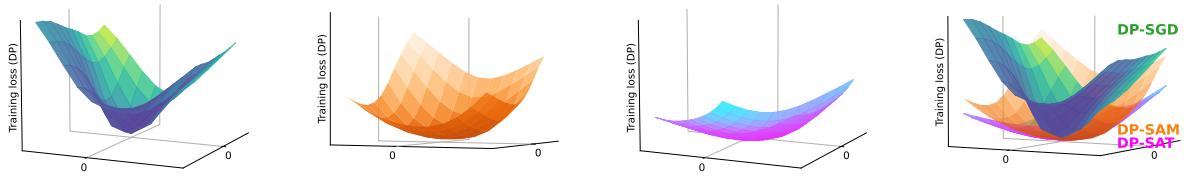


Figure 10. Illustration of DP-SGD, DP-SAM, and DP-SAT.

We further compare the loss landscapes of DP-SGD, DP-SAM, and DP-SAT in the same way of Figure 5.



(a) DP-SGD (b) DP-SAM (c) DP-SAT (d) All

Figure 11. Visualization of the loss landscapes of DP-SGD, DP-SAM, and DP-SAT.

E. Why Ascent Step should be privatized in DP-SAM?

For clear understanding, here we denote the non-private parameter as $\bar{\boldsymbol{\theta}}$ with bar and private parameter $\hat{\boldsymbol{\theta}}$ with a hat. To satisfy the condition of Moments accountant for DP-SGD, each of the previous weights should be private as $\hat{\boldsymbol{\theta}}_{t-1}$. In each step, the differential privacy of $\boldsymbol{\theta}_t$ is guranteed by a differentially private mechanism $\mathcal{M}$, i.e., $\hat{\boldsymbol{\theta}}_t = \mathcal{M}(\hat{\boldsymbol{\theta}}_{t-1}; \boldsymbol{x})$, where $\boldsymbol{x}$ is training data samples.

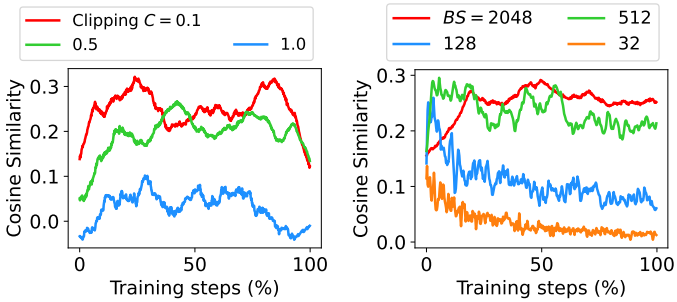
Consider the simple two-step mechanism $\mathcal{M}(\hat{\boldsymbol{\theta}}; \boldsymbol{x}) = \mathcal{M}_2(\mathcal{M}_1(\hat{\boldsymbol{\theta}}; \boldsymbol{x}), \hat{\boldsymbol{\theta}}; \boldsymbol{x})$. We argue that $\mathcal{M}(\hat{\boldsymbol{\theta}}; \boldsymbol{x})$ is differentially private when $\mathcal{M}_2, \mathcal{M}_1$ are private by the composition theorem w.r.t. training data $\boldsymbol{x}$.

By contradiction, let only $\mathcal{M}_2$ is private w.r.t $\boldsymbol{x}$, $\mathcal{M}(\hat{\boldsymbol{\theta}}; \boldsymbol{x}) = \mathcal{M}_2(\bar{\boldsymbol{\theta}}', \hat{\boldsymbol{\theta}}; \boldsymbol{x})$, where the output of $\mathcal{M}_1$ is not private $\bar{\boldsymbol{\theta}}'$. Thus, we cannot use the post-processing or calculate the sensitivity of $\mathcal{M}_2$ because $\bar{\boldsymbol{\theta}}'$ possesses information of $\boldsymbol{x}$. This means that adding noise only to $\mathcal{M}_2$ cannot guarantee the complete privacy level. Thus, we should consider the privacy budget consumed by $\mathcal{M}_1$ w.r.t. $\boldsymbol{x}$.

F. Why Previous Gradient can be used in DP-SAT?

This section demonstrates that DP-SAT can take advantage of sharpness-aware training even when the ascent direction is derived from different batch samples. The primary motivation is that gradient-based optimization occurs in a low-dimensional subspace of top eigenvalues (Sagun et al., 2018; Papyan, 2019). According to Figure 8, we can see that the Eigenspectrum of DP is separable into two divisions, analogous to non-DP training. The principal subspace consists of a number of classes outliers (about 10 in CIFAR-10) with large eigenvalues, separated from a continuous bulk centered on zero. Exploiting the principal subspace, it is more advantageous to utilize the previous perturbed gradients $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{t-1}$ as an ascent direction than random directions. Note that Jang et al. (2022) and Lee et al. (2022) have proposed methods to implicitly regularize the sharpness through the use of Hessian approximation, yielding experimental results similar to SAM.

The utilization of clipping and large batch size, two properties of DP training, cause the gradients from different batch samples to be alike. The clipping operation affects the training dynamics as mentioned in Section 3 and DP training usually uses larger batch sizes to decrease the level of injected noise σ (Tramer & Boneh, 2021). To check the cosine similarities, we now investigate how the clipping value C and batch sizes affect cosine similarities of ascent steps $\bar{\boldsymbol{\delta}}_t = \sum_{i \in I_t} \text{clip}(\nabla \ell_i(\boldsymbol{w}_t), C)$, before adding noise. The cosine similarities of current ascent direction $\bar{\boldsymbol{\delta}}_t$ and previous ascent direction $\bar{\boldsymbol{\delta}}_{t-1}$ for the total training procedure are illustrated in Figure 12. We fixed the batch size of 2048 for experiments on C and fixed $C = 0.1$ for experiments on batch size. In general, the larger batch size and the smaller clipping value result in higher cosine similarities of $\bar{\boldsymbol{\delta}}_t$ and $\bar{\boldsymbol{\delta}}_{t-1}$. Furthermore, all the cosine similarities drastically rise at the early epochs, which are coherent periods for the sudden drop of train loss in Figure 2.



(a) Impact of clipping value C (b) Impact of batch size

Figure 12. Cosine similarity of the current ascent direction $\bar{\boldsymbol{\delta}}_t$ and previous ascent direction $\bar{\boldsymbol{\delta}}_{t-1}$ before adding noise. Small clipping values C and large batch sizes lead to similar gradient directions. The curves are smoothed for better visualization and the x-axis is denoted in percentage because training with smaller batch size has more training steps.

We then take the average cosine similarities of Figure 12 in Table 8. In real experimental settings, batch size of 2048 and $C = 0.1$, cosine similarity of 0.26 indicates a high level of alignment, given the high dimensionality of deep learning models. These results are in contrast to zero alignments in the case of $C = \infty$ or a batch size of 32. Note that Andriushchenko & Flammarion (2022) revealed that accurately solving the internal maximization problem has analogous effects to enlarging

the radius ρ due to the nonlinearity of the weights space. This might be a reason that the ascent direction can be selected without computing the ascent direction within the same mini-batch.

Table 8. Average cosine similarities of ascent steps at the current and previous steps during all training steps in Figure 12.

C	∞	1	0.5	0.1
$\mathbb{E}_t[\cos(\boldsymbol{\delta}_t, \boldsymbol{\delta}_{t-1})]$	-	0.03	0.20	**0.26**
Batch size	32	128	512	2048
$\mathbb{E}_t[\cos(\boldsymbol{\delta}_t, \boldsymbol{\delta}_{t-1})]$	0.03	0.10	0.23	**0.26**

G. Ablation study

G.1. Momentum variants

Applying the concept of momentum in SAM is widely used to enhance the performance of SAM (Du et al., 2022a; Qu et al., 2022). We also suggest DP-SAT-Momentum, which uses the idea of momentum to calculate the ascent direction in DP-SAT. First of all, the momentum updates at t step as follows:

$$\boldsymbol{v}_t \leftarrow \gamma \boldsymbol{v}_{t-1} + \boldsymbol{g}_t. \tag{13}$$

Then, the accumulated momentum can be written as:

$$\boldsymbol{v}_t = \sum_{\tau=1}^{t-1} \gamma^{\tau} \boldsymbol{g}_{t-\tau}. \tag{14}$$

Then, we can set the ascent direction using all the perturbed gradient information from step $1, \dots, t-1$ as follows:

$$\boldsymbol{\delta}_t^{Momentum} = \rho \frac{\boldsymbol{v}_t}{\|\boldsymbol{v}_t\|} = \rho \frac{\gamma \boldsymbol{g}_{t-1} + \gamma^2 \boldsymbol{g}_{t-2} + \cdots + \gamma^{t-1} \boldsymbol{g}_1}{\|\gamma \boldsymbol{g}_{t-1} + \gamma^2 \boldsymbol{g}_{t-2} + \cdots + \gamma^{t-1} \boldsymbol{g}_1\|}. \tag{15}$$

As we use base optimizers with momentum (which will be discussed in the next subsection), we set the momentum γ the same as the momentum of the base optimizer for convenience. This method is also free from additional privacy costs because all the previous perturbed gradients are guaranteed by post-processing.

The experimental results of DP-SAT-Momentum are in Table 9. Empirically, we should use the radius ρ for DP-SAT-Momentum as 10 times larger than that of DP-SAT. Both optimization methods show similar experimental effects. We believe this phenomenon is that the gradient similarities induced in DP training aforementioned in Appendix F are sufficient to approximate the ascent step, different from non-DP training (Du et al., 2022a; Qu et al., 2022).

Table 9. Performance of DP-SAT and DP-SAT-Momentum on MNIST, FashionMNIST, and CIFAR-10.

Datasets	Model	Privacy budget ε ($\delta = 10^{-5}$)	Optimizers	
			DP-SAT	DP-SAT-Momentum
MNIST	DPNAS-MNIST (#Params: 0.21M)	$\varepsilon = 1$	97.96±0.08	98.00±0.13
		$\varepsilon = 2$	98.71±0.09	98.83±0.11
		$\varepsilon = 3$	98.93±0.02	98.91±0.11
FashionMNIST	DPNAS-MNIST (#Params: 0.21M)	$\varepsilon = 1$	85.92±0.35	85.47±0.42
		$\varepsilon = 2$	87.75±0.24	87.55±0.18
		$\varepsilon = 3$	88.60±0.04	88.56±0.30
CIFAR-10	DPNAS-CIFAR10 (#Params: 0.53M)	$\varepsilon = 1$	60.13±0.34	60.10±0.46
		$\varepsilon = 2$	67.23±0.12	67.24±0.21
		$\varepsilon = 3$	69.86±0.49	69.63±0.10

G.2. Without tuning the radius ρ

We provide a baseline for experiments that do not involve tuning the radius ρ in Table 10, particularly with regard to Table 2. To address this, we set $\rho = 0.03$ for the MNIST and FashionMNIST datasets and $\rho = 0.01$ for the DPNAS-CIFAR10 in all experiments presented in Table 2. The other settings except for the radius ρ are the same.

Table 10. Classification accuracy of DP-SGD, and DP-SAT on the MNIST, FashionMNIST, and CIFAR-10 datasets with fixing $\rho = 0.03$ (except for DPNAS-CIFAR10 architecture with $\rho = 0.01$). We bold the highest average accuracy.

Datasets	Model	ε	ρ	DP-SGD	DP-SAT
MNIST	GNResnet-10	1	0.03	95.15±0.17	**96.00±0.21**
		2	0.03	96.68±0.27	**97.35±0.14**
		3	0.03	97.30±0.14	**97.83±0.10**
	DPNAS-MNIST	1	0.03	97.77±0.13	**97.96±0.08**
		2	0.03	98.60±0.06	**98.71±0.09**
		3	0.03	98.70±0.12	**98.93±0.02**
FashionMNIST	GNResnet-10	1	0.03	80.57±0.25	**81.33±0.45**
		2	0.03	82.71±0.35	**84.53±0.41**
		3	0.03	84.55±0.17	**85.91±0.22**
	DPNAS-MNIST	1	0.03	84.62±0.19	**85.52±0.28**
		2	0.03	86.99±0.57	**87.72±0.28**
		3	0.03	87.97±0.17	**88.38±0.23**
CIFAR-10	CNN-Tanh with SELU	1	0.03	**45.24±0.42**	**45.45±0.36**
		2	0.03	**56.90±0.33**	**57.34±0.59**
		3	0.03	61.84±0.48	**62.64±0.45**
	DPNAS-CIFAR10	1	0.01	59.42±0.38	**60.13±0.34**
		2	0.01	66.30±0.27	**67.23±0.12**
		3	0.01	68.43±0.43	**69.86±0.49**
SVHN	DPNAS-CIFAR10	1	0.03	82.25±0.15	**82.95±0.28**
		2	0.03	86.85±0.33	**87.49±0.15**
		3	0.03	88.18±0.23	**88.74±0.18**

G.3. Different base optimizers

The experimental results of using other optimizers, such as SGD without momentum and Adam are in Table 11. For Adam, we select the learning rate of 0.0002. The difference is marginal between SGD with a momentum of 0.9 and Adam, showing better performance than SGD without momentum. In all cases, DP-SAT can achieve better performance regardless of base optimizers.

Table 11. Classification accuracies for DP-SGD, DP-Adam, and DP-SAT on the MNIST, FashionMNIST, and CIFAR-10 datasets. β indicates momentum. The bold indicates results within the standard deviation of the highest mean score.

Datasets	Privacy budget ε ($\delta = 10^{-5}$)	Optimizer				
		DP-SGD ($\beta = 0.0$)	DP-SGD ($\beta = 0.9$)	DP-Adam	DP-SAT (SGD, $\beta = 0.9$)	DP-SAT (Adam)
MNIST	$\varepsilon = 1$	97.62±0.13	97.77±0.13	97.96±0.15	97.96±0.08	**98.30±0.12**
	$\varepsilon = 2$	97.65±0.22	98.60±0.06	98.54±0.10	**98.71±0.09**	**98.65±0.10**
	$\varepsilon = 3$	97.70±0.23	98.70±0.12	98.68±0.07	**98.93±0.02**	98.85±0.09
FashionMNIST	$\varepsilon = 1$	80.80±0.24	84.62±0.19	84.85±0.34	**85.92±0.35**	85.27±0.34
	$\varepsilon = 2$	81.55±0.52	86.99±0.57	87.17±0.21	**87.75±0.24**	87.26±0.28
	$\varepsilon = 3$	82.03±0.19	87.97±0.17	87.84±0.38	**88.60±0.04**	88.26±0.40
CIFAR-10	$\varepsilon = 1$	56.15±0.74	59.42±0.38	61.75±0.60	60.13±0.34	**62.52±0.61**
	$\varepsilon = 2$	56.59±0.79	66.30±0.27	66.68±0.39	**67.23±0.12**	**67.26±0.36**
	$\varepsilon = 3$	56.87±0.73	68.43±0.43	68.86±0.39	**69.86±0.49**	69.48±0.38

G.4. Effect of radius ρ

We compare the performance of DP-SAT under different ρ. We illustrate the results of various ρ on CIFAR-10 in Figure 13. It shows the importance of finding an appropriate radius in parameter space, where too small or big radius cannot improve the performance. Furthermore, because of the instability of DP training, the accuracy has a high variance and shows some fluctuation in the tendency.

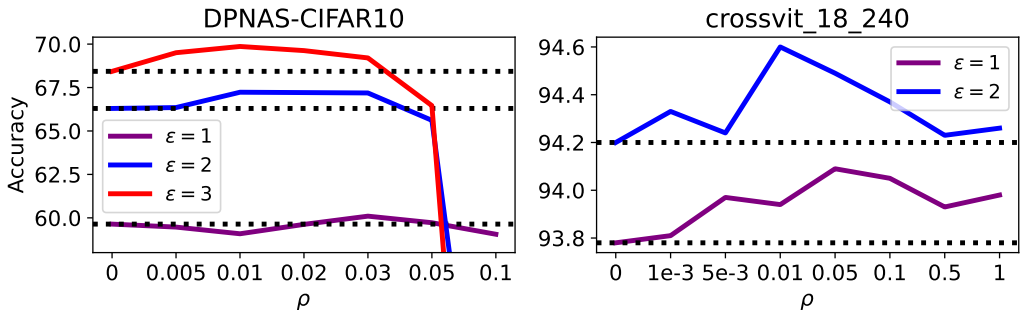


Figure 13. Sensitivity analysis of ρ in DP-SAT on CIFAR-10 dataset.

G.5. Performance of $(2\varepsilon, 2\delta)$ DP-SAM

Table 12 shows the performance of DP-SGD and DP-SAM with $\varepsilon = \{1, 2, 3\}$ of DPNAS-CIFAR models on CIFAR-10. The experimental settings are the same in Section 5. DP-SAM shows worse accuracy than DP-SGD due to its extensive privacy consumption on each iteration. If we consider $(2\varepsilon, 2\delta)$ DP-SAM †, which can guarantee the same training steps T as DP-SGD, then sharpness-aware training achieves better performance in DP training. This indicates that sharpness-aware training can improve performance if we can choose the direction of the ascent step without privacy consumption.

Table 12. Performance of DP-SGD and DP-SAM on CIFAR-10. DP-SAM shows an accuracy drop because of the doubled privacy budget. The last column $(2\varepsilon, 2\delta)$ DP-SAM† indicates the possible improvements of sharpness-aware training.

Privacy budget ε ($\delta = 10^{-5}$)	Methods		
	DP-SGD	DP-SAM	$(2\varepsilon, 2\delta)$ DP-SAM †
$\varepsilon = 1$	59.42±0.38	45.15±0.63	60.18±0.52
$\varepsilon = 2$	66.30±0.27	58.85±0.70	66.74±0.37
$\varepsilon = 3$	68.43±0.43	63.59±0.25	69.58±0.27
Training epochs	30	7.07	30

G.6. Convergence analysis

We illustrate the convergence of training loss and corresponding test accuracy of DP-SGD and DP-SAT on CIFAR-10 and MNIST in Figures 14 and 15. The convergence speed is a little bit slower than DP-SGD but it can reach lower training loss, which is a similar phenomenon of SGD and SAM as depicted in (Kaddour et al., 2022).

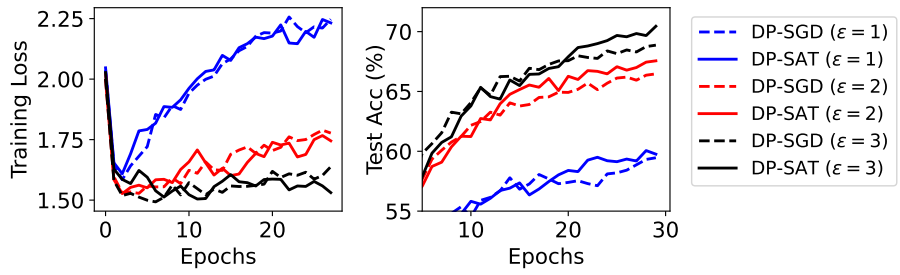


Figure 14. Training loss and Test accuracy of DP-SGD and DP-SAT by varying ε on CIFAR-10.

G.7. Accuracy plot of Figure 7

We illustrate the accuracy plot of Figure 7 in Figure 16.

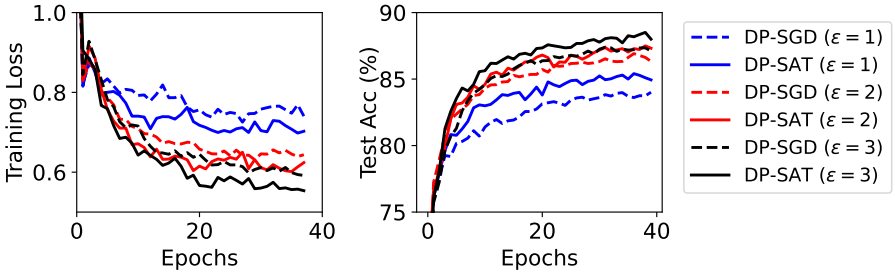


Figure 15. Training loss and Test accuracy of DP-SGD and DP-SAT by varying ε on MNIST.

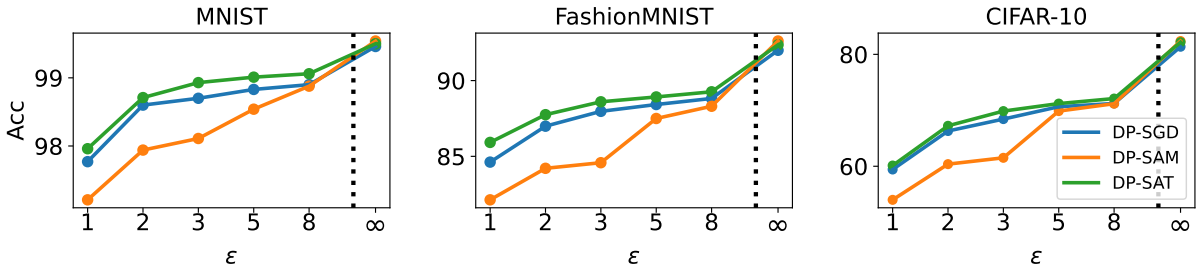


Figure 16. Accuracy for DP-SAM and DP-SAT w.r.t DP-SGD. $\varepsilon = \infty$ indicates non-DP settings.