Rethinking DP-SGD in Discrete Domain: Exploring Logistic Distribution in the Realm of SIGNSGD

Jonggyu Jang [1] **Seongjin Hwang** [1] **Hyun Jong Yang** [1]

Abstract

Deep neural networks (DNNs) have a risk of remembering sensitive data from their training datasets, inadvertently leading to substantial information leakage through privacy attacks like membership inference attacks. DP-SGD is a simple yet effective defense method that incorporates Gaussian noise into gradient updates to safeguard sensitive information. With the prevalence of large neural networks, DP-SIGNSGD, a variant of DP-SGD, has emerged, aiming to curtail communication load while maintaining security. However, it is noteworthy that most DP-SIGNSGD algorithms default to Gaussian noise, suitable only for DP-SGD, without scant discussion of its appropriateness for SIGNSGD. Our study delves into an intriguing question: "*Can we find a more efficient substitute for Gaussian noise to secure privacy in* DP-SIGNSGD*?*" We propose an answer with a Logistic mechanism, which conforms to SIGNSGD principles and is interestingly evolved from an exponential mechanism. In this paper, we provide both theoretical and experimental evidence showing that our method surpasses DP-SIGNSGD.

1. Introduction

The realm of machine learning has faced substantial challenges related to data privacy. Notably, deep neural networks (DNNs) can unintentionally remember sensitive information from the datasets they were trained on (Feldman & Zhang, 2020). This issue becomes problematic not only when DNNs are released to the public (Song et al., 2017), but also when only inference results are made available (Papernot et al., 2017).

In response to these privacy challenges, several solutions have been proposed, including federated learning (FL) by McMahan et al. (2017), homomorphic encryption (HE) by Lee et al. (2022)), and differentially private stochastic gradient descent (DP-SGD) by Abadi et al. (2016). While HE is hampered by its high complexity, and FL can not prevent data memorization, the DP-SGD algorithm emerges as a simpler yet efficient method by adding Gaussian noise for securing trained DNN models. The effectiveness of DP-SGD is validated by the adoption of differential privacy (DP) (Dwork et al., 2006), the underlying principle of DP-SGD, by major tech companies like Apple, Google, and Microsoft in their data collection practices (Apple, 2017; Nguyên et al., 2016).

Aside from this, another new challenge has arisen: the ever-increasing size of deep neural networks in distributed learning frameworks. As a solution, one can adopt SIGNSGD to compress gradient vectors for efficient node-to-node gradient exchanges (Bernstein et al., 2019). The importance of gradient compression becomes increasingly pivotal as the size of neural networks explosively enlarges.

In this paper, we mainly focus on the combining advantages of SIGNSGD and DP-SGD for reserving communication efficiency while preserving privacy. Previously, few straightforward approaches have been proposed by Jin et al. (2020); Lyu (2021)—namely, DP-SIGNSGD. However, a notable limitation arises in the naive implementation of the DP-SIGNSGD algorithm: the use of Gaussian noise, while specialized for DP-SGD, is not ideally suited for SIGNSGD. In DP, the distribution of additive noise is an important factor, because the goal of DP research is to maximize utility while preserving statistical data privacy. At this point, we raise a fundamental question:

"*What is a better noise mechanism for* SIGNSGD*?*"

Our solution, DP-SIGNLOSGD, replaces the traditional Gaussian mechanism with an additive Logistic mechanism. To this end, we revisit the foundational principles of SIGNSGD, which finds a point in a set $\{-1,1\}^N$ nearest to the stochastic gradient—namely, *sign sampling problem*. Our initial strategy utilizes an exponential mechanism, a utility-maximizing mechanism, by setting the utility based

[1] Department of Electrical Engineering, Pohang University of Science and Technology, Pohang, South Korea. Correspondence to: Hyun Jong Yang <hyunyang@postech.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

on the sign sampling problem. However, we face a challenge: the exponential mechanism is not an additive noise form; thus, we cannot answer the above research question. To address this, we explore an alternative approach using the additive Logistic mechanism before `sign` function. Moreover, we show that our method can sample an element from the set $\{-1, 1\}^N$ in the same manner that mirrors the distribution of the exponential mechanism.

Novelty Extensive research has been conducted on DP-SGD; however, none of the following works directly addresses our question: tighter DP accounting (Andrew et al., 2023; Jagielski et al., 2020; Girgis et al., 2021), Renyi-DP (Girgis et al., 2021), Gaussian-DP (Dong et al., 2019), gradient sparsification (Zhu & Blaschko, 2021), enhanced projection (Sha et al., 2023), enhanced data sampling (Heo et al., 2023), sensitivity optimization (Galli et al., 2023), large language models (Hong et al., 2023), and gradient compression (Lin, 2022; Kerkouche et al., 2021).

Summary of our findings In this paper, our introduction of DP-SIGNLOSGD marks a significant advancement, offering enhanced accuracy and convergence over the additive Gaussian noise, and thereby establishing a new standard in the field. Our salient findings are four-fold:

- **Logistic mechanism for SIGNSGD:** We present DP-SIGNLOSGD, a novel approach that combines SIGNSGD with a Logistic mechanism for guaranteeing DP. To this end, we develop an exponential mechanism adhering to the principles of SIGNSGD. Furthermore, for sampling efficiency, we show that incorporating Logistic noise before the `sign` function seamlessly integrates with our exponential mechanism.
- **Enhanced utility and privacy loss:** Our theoretical and experimental findings confirm that the Logistic mechanism offers a tighter privacy loss and higher utility compared to Gaussian noise, particularly in accurately selecting the sign of the gradient within the same privacy budget.
- **Improved convergence:** We show that the convergence speed of SIGNSGD with additive noises is hampered by an additional noise standard deviation term. However, DP-SIGNLOSGD, requiring lower noise variance, ensures more efficient convergence than the DP-SIGNSGD.
- **Experimental result:** Our extensive experiments show that the classification accuracy of DP-SIGNLOSGD is higher than that of DP-SIGNSGD across all hyper-parameter combinations, under the same privacy budget. [1]

2. Backgrounds and Related Works

This section presents backgrounds and related works on DP, offering foundational insights for those new to the topic, while experts and practitioners with a pre-established understanding of DP may choose to proceed to subsequent sections.

2.1. Differential Privacy

DP is a mathematical definition that quantifies the privacy of a randomized algorithm. Let us consider a dataset $d \in \mathcal{X}^n$ and a randomized algorithm A, where n is the number of elements in the dataset. For all adjacent datasets[2] d and d', we can measure the statistical difference between $A(d)$ and $A(d')$. Intuitively, if $A(d)$ and $A(d')$ are statistically indistinguishable, the randomized algorithm A is private because any single instance seldom affects the algorithm's output. The formal definition of (ϵ, δ)-DP is available in the following definition.

Definition 2.1 (Approximate differential privacy)**.** For an arbitrary domain and range sets $\mathcal{X}^n$ and $\mathcal{R}$, consider a randomized mechanism $A : \mathcal{X}^n \to \mathcal{R}$. The randomized mechanism A satisfies (ϵ, δ)-DP, if for all two adjacent datasets $d, d' \in \mathcal{D}$ and for all $T \subset \mathcal{R}$,

$$\Pr\left[A(d) \in T\right] \leq \exp(\epsilon) \cdot \Pr\left[A(d') \in T\right] + \delta. \tag{1}$$

We note that the original definition of ϵ-DP is $(\epsilon, 0)$-DP, by setting additive term $\delta = 0$. DP is known for properties like composability (the combination of multiple DP mechanisms remains DP) and robustness to post-processing (any function applied after a DP mechanism does not weaken its DP guarantee).

In our work, we aim to render trained DNN models differentially private. This means making the models indistinguishable whether or not they include an arbitrary data element, thereby ensuring privacy preservation in DP contexts.

2.2. Representative DP Mechanisms

In the paper (Dwork & Roth, 2014), various DP mechanisms are introduced, including additive noise mechanisms and randomized selection mechanisms. To define a mechanism that satisfies (ϵ, δ)-DP, one crucial aspect to determine is the sensitivity of the function—essentially, how much its output varies with input changes. In simpler terms, if a function is highly sensitive to input changes, it requires the addition of proportionally larger noise to maintain privacy.

Definition 2.2 (ℓ_m sensitivity)**.** For adjacent datasets d and d', the ℓ_m sensitivity of a query f can be represented by

$$\Delta_m = \max_{d,d'} \|f(d) - f(d')\|_m. \tag{2}$$

[1] By following the standard in DP-SGD and its variations, we use CIFAR-10 and MNIST datasets for our experiments.

[2] We say that two datasets are adjacent if they differ in a single entry.

For numeric functions, three representative mechanisms are widely used to guarantee DP: 1) Laplace mechanism, 2) Gaussian mechanism, and 3) exponential mechanism.

Definition 2.3 (Laplace Mechanism)**.** Let Δ_1 denote the ℓ_1 sensitivity of f. The Laplace mechanism, a fundamental method for achieving $(\epsilon, 0)$-DP, adds Laplace noise to the output of f:

$$A(d) = f(d) + n_{\text{Lap}}, \tag{3}$$

where $n_{\text{Lap}} \sim$ Laplace$(0, \frac{\Delta_1}{\epsilon})$.

The Laplace noise is a symmetric and exponentially decaying distribution, where the tail of which exponentially decays, thereby achieving $(\epsilon, 0)$-DP.

Definition 2.4 (Gaussian Mechanism)**.** Let Δ_2 be the ℓ_2 sensitivity of f. The Gaussian mechanism, a representative method guaranteeing (ϵ, δ)-DP, adds Gaussian noise to the output of f by

$$A(d) = f(d) + n_{\text{Gau}}, \tag{4}$$

where $n_{\text{Gau}} \sim \mathcal{N}(0, \frac{2\Delta_2^2 \log(1.25/\delta)}{\epsilon^2})$.

The Gaussian mechanism adds a symmetric and bell-shaped distribution, Gaussian noise, to the function output. Unlike the Laplace mechanism, the tail of the Gaussian mechanism drops off faster than exponential; thus, an additional margin $\delta > 0$ is required.

Definition 2.5 (Exponential mechanism)**.** Let us consider a score function $s : \mathcal{X}^n \times \mathcal{R} \to \mathbb{R}$. Then, the exponential mechanism A samples an output from $\mathcal{R}$ by following the probability density function over $\mathcal{R}$:

$$\Pr(A(d) = r) = \frac{\exp(\frac{\epsilon s(d,r)}{2\Delta_1})}{\int_{\mathcal{R}} \exp(\frac{\epsilon s(d,a)}{2\Delta_1}) da}. \tag{5}$$

In the exponential mechanism, the scoring function is a method where the analyst assigns scores to each element based on specific criteria, determining the 'best' from the set. The key property is, that the exponential mechanism enables us to return a *precise answer* (without additive noise), i.e., the output is always a member of set $\mathcal{R}$. This property motivates us to apply the exponential mechanism to SIGNSGD problem, sampling exactly from the set $\{-1, 1\}$.

Remark 2.6 (Foundation mechanism). Another important property of the exponential mechanism is its *foundational* role in developing other DP mechanisms. By setting the score function as ℓ_1 distance between $f(d)$ and $M_E(d)$, we can derive the Laplace mechanism. Similarly, it provides an intuitive basis for the Gaussian mechanism by considering the square of ℓ_2 distance between $f(d)$ and $M_E(d)$, though $\delta > 0$ due to unbounded ℓ_1 sensitivity of this score function.

Algorithm 1 DP-SIGNSGD

```
1: Input: dataset $\mathcal{D} = \{z_1, z_2, ..., z_n\}$, loss function $\mathcal{L}(\theta) = \sum_i l(\theta, z_i)$.
2: Parameters: noise scale $\sigma$, batch size $B$, and gradient norm bound $C$.
3: Initialize $\theta_0$ randomly
4: for $t \in \{0, ..., T-1\}$ do
5:   Sample a random batch $\mathcal{B}_t$.
6:   for $i \in \mathcal{B}_t$ do
7:     $\mathbf{g}_t(z_i) \leftarrow \nabla_\theta l(\theta_t, z_i)$ {Compute gradient}
8:     $\tilde{\mathbf{g}}_t(z_i) \leftarrow \mathbf{g}_t(z_i) / \max\left(1, \frac{\|\mathbf{g}_t(z_i)\|_2}{C}\right)$ {Clipping}
9:   end for
10:  $\bar{\mathbf{g}}_t \leftarrow \texttt{sign}\left(\sum_{i \in \mathcal{B}_t} \tilde{\mathbf{g}}_t(z_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
11:  $\theta_{t+1} \leftarrow \theta_t - \eta_t \bar{\mathbf{g}}_t$
12: end for
13: Output: $\theta_T$ and computed privacy cost $(\epsilon, \delta)$.
```

2.3. DP-SIGNSGD

Here, we introduce DP-SIGNSGD (Jin et al., 2020; Lyu, 2021), outlined in Algorithm 1. DP-SIGNSGD is a variant of DP-SGD (Abadi et al., 2016), where the `sign` function is added to reduce the communication overhead in distributed learning (Bernstein et al., 2018). In Line 8, the gradient vector of each data is clipped by ℓ_2 norm with a threshold of C for boundedness of sensitivity. In Line 10, the Gaussian noise is added to ensure DP. By considering the `sign` function in Line 10 as a post-processing step, the privacy moments accountant works the same as in the DP-SGD.

3. Main Results

In this section, we present DP-SIGNLOSGD by replacing the additive Gaussian noise with additive Logistic noise. As in the paper (Jin et al., 2020), one might attempt to design DP-SIGNSGD by adding `sign` function as a post-processing module. Unfortunately, the number of trainable epochs is limited due to DP, and the added `sign` function slows down the convergence. Moreover, the Gaussian noise, designed for perturbing continuous-valued gradient, leads to a loose privacy loss, thereby destroying the utility of the learned model. Therefore, we aim to find a surrogate noise distribution appropriate for SIGNSGD.

To this end, we first formulate a problem to find the steepest direction that minimizes the linearly approximated loss function. Then, we derive that the exponential mechanism for the formulated problem is equivalent to adding Logistic noise before `sign` function.

Let us consider a stochastic optimization problem with the dataset $\mathcal{D} = \{z_1, ..., z_n\}$ and the loss function $\mathcal{L}(\theta) = \sum_{i=1}^n l(\theta, z_i)$, where $\theta \in \mathbb{R}^N$ denotes neural network parameters. For brevity, we denote $\mathbf{g}(z_i) = \nabla_\theta l(\theta, z_i)$. For guaranteeing DP, our method requires bounding the sensitivity of the mini-batch gradient, which is also a popular ingredient in machine learning for non-privacy objectives. Similar to DP-SGD, we employ ℓ_2-clipping with a threshold C. The clipped gradient is thus represented as

$$\tilde{\mathbf{g}}(z_i) = \mathbf{g}(z_i) / \max\left(1, \frac{\|\mathbf{g}(z_i)\|_2}{C}\right). \tag{6}$$

With a step size of α and minibatch $\mathcal{B}$, signSGD updates the parameter θ by

$$\theta \leftarrow \theta - \alpha \cdot \texttt{sign}\left(\sum_{i\in\mathcal{B}} \tilde{\mathbf{g}}(z_i)\right). \tag{7}$$

Intuition of signSGD In Equation (7), the sign of the gradient is leveraged to update the parameter θ. The choice of the sign can be reconsidered by finding a point in $\{-\alpha, \alpha\}^N$ nearest to the stochastic gradient. Thus, the *sign sampling problem* is formulated as

$$\min_{\mathbf{v}\in\{-\alpha,\alpha\}^N} \|\mathbf{v} + \alpha\tilde{\mathbf{g}}_\mathcal{B}\|_2^2 \longleftrightarrow \min_{\mathbf{v}\in\{-\alpha,\alpha\}^N} \mathbf{v}^\mathrm{T}\tilde{\mathbf{g}}_\mathcal{B}, \tag{8}$$

where $\tilde{\mathbf{g}}_\mathcal{B} = \sum_{i\in\mathcal{B}} \tilde{\mathbf{g}}(z_i)$, and the problem transformation holds because $\|\mathbf{v}\|_2^2 = N\alpha^2$ is a constant. The solution of the problem (8) is equivalent to signSGD in Equation (7), i.e., $\mathbf{v} = -\alpha \cdot \texttt{sign}(\tilde{\mathbf{g}}_\mathcal{B})$. In the next section, we propose an exponential mechanism based on the problem (8).

3.1. Motivation: Exponential Mechanism for signSGD

Our motivation is that the exponential mechanism can be used as a foundation mechanism. By following this, we design the exponential mechanism $\mathcal{E}$ for solving the problem (8) by letting the score function as $s(\tilde{\mathbf{g}}_\mathcal{B}, \mathbf{v}) = -\mathbf{v}^\mathrm{T}\tilde{\mathbf{g}}_\mathcal{B}$, where $\mathbf{v} \in \mathcal{H} = \{-\alpha, \alpha\}^N$. Then, by letting $v_i = [\mathbf{v}]_i$ and $\tilde{g}_i = [\tilde{\mathbf{g}}_\mathcal{B}]_i$, the sampling probability in the set $\mathcal{H}$ is defined by

$$\begin{aligned}\Pr[\mathcal{E}(\tilde{\mathbf{g}}_\mathcal{B}) = \mathbf{v}] &\propto \exp\left(-\mathbf{v}^\mathrm{T}\tilde{\mathbf{g}}_\mathcal{B}/(2sC)\right) \\ &= \prod_{i=1}^{N} \exp\left(-v_i\tilde{g}_i/(2sC)\right), \forall \mathbf{v} \in \mathcal{H},\end{aligned} \tag{9}$$

where $2sC$ is the scale factor of the distribution to guarantee DP. The remaining challenge is "*how to determine an appropriate scale s for guaranteeing DP?*" For instance, if $s = \infty$, the distribution in (9) is a uniform distribution over the set $\mathcal{H}$ regardless of input gradient, achieving $(0, 0)$-DP. One can directly use the exponential mechanism for machine learning algorithms; however, its distribution in (9) is non-identical and complex, thereby causing implementational inefficiency. Thus, our next goal is to derive an *identical* and simple additive noise distribution equivalent to this exponential mechanism.

Remark 3.1. The *Gaussian noise* is an appropriate choice for the *standard SGD without* `sign` function. Consider the intuition behind SGD as a proximal gradient descent, formulated as $\min_\mathbf{v} \mathbf{v}^\mathrm{T}\mathbf{g}_\mathcal{B} + \frac{\|\mathbf{v}\|_2^2}{2\alpha}$. Following this motivation, we derive Gaussian noise, where the probability is proportional to $\exp((\mathbf{v}^\mathrm{T}\mathbf{g}_\mathcal{B} + \frac{\|\mathbf{v}\|_2^2}{2\alpha}) \cdot \frac{2\alpha}{\sigma^2})$.

Remark 3.2. The exponential mechanism in (9) satisfies the original DP ($(\epsilon, 0)$-DP) if we set $s = \alpha\sqrt{N}/\epsilon$, because the ℓ_1 sensitivity of the score function is $\alpha\sqrt{N}$.

3.2. Transform Exponential to Logistic Mechanism

In this section, we aim to derive the additive noise mechanism equivalent to the exponential mechanism in (9). For brevity of notation, we will continue to use $\tilde{g}_i = [\tilde{\mathbf{g}}_\mathcal{B}]_i$. Given that each element of mechanism $\mathcal{E}$ is independently distributed, considering the output range is the set $\{-\alpha, \alpha\}^N$, the output of mechanism $\mathcal{E}$ has the following probability mass function:

$$\Pr([\mathcal{E}(\tilde{\mathbf{g}}_\mathcal{B})]_i = v_i) = \frac{\exp(-\frac{v_i\tilde{g}_i}{2sC})}{\exp(\frac{\tilde{g}_i}{2sC}) + \exp(-\frac{\tilde{g}_i}{2sC})}. \tag{10}$$

Here, our focus is to derive a noise distribution of random variable n_i that satisfies $[\mathcal{E}(\tilde{\mathbf{g}}_\mathcal{B})]_i = -\texttt{sign}(\tilde{g}_i + n_iC)$. We start from the distribution of noise n_i making the output negative: $\Pr(n_i + \tilde{g}_i/C < 0) = \Pr([\mathcal{E}(\tilde{\mathbf{g}}_\mathcal{B})]_i = 1)$. From this, the cumulative distribution function of n_i is represented by

$$\begin{aligned}&\Pr(n_i < -\tilde{g}_i/C) = \Pr(n_i + \tilde{g}_i/C < 0) \\ &= \Pr([\mathcal{E}(\tilde{\mathbf{g}}_\mathcal{B})]_i = 1) = \frac{\exp(-\frac{\tilde{g}_i}{2sC})}{\exp(\frac{\tilde{g}_i}{2sC}) + \exp(-\frac{\tilde{g}_i}{2sC})}.\end{aligned} \tag{11}$$

From the cumulative distribution function we obtained, the probability density function of n_i can be derived as a Logistic distribution:

$$f_{n_i}(x) = \frac{\exp(-x/s)}{s(1 + \exp(-x/s))^2}, \tag{12}$$

which can be denoted as $n_i \sim \text{Logistic}(0, s)$. We note that the obtained probability distribution is *independent and identical* for all elements of the gradient $\tilde{\mathbf{g}}$.

3.3. Proposed Method: DP-signLoSGD

Here, we present DP-signLoSGD, a differentially private signSGD mechanism with additive Logistic noise, which can be easily implemented by replacing the Gaussian noise in Line 10 of Algorithm 1 with Logistic distribution with scale of s. Due to limited space, the step-by-step algorithm of DP-signLoSGD is available in Algorithm 2 (on page 12). The details of the DP-signLoSGD update are defined in the following definition.

Definition 3.3 (DP-SIGNLOSGD)**.** For a stochastic gradient vector $\mathbf{g} \in \mathbb{R}^N$ targeted a specific weight θ, let us consider the clipped mini-batch stochastic gradient $\tilde{\mathbf{g}}$. Then, DP-SIGNLOSGD updates the neural network weight θ by

$$\theta \leftarrow \theta - \alpha \, \mathtt{sign}\left(\tilde{\mathbf{g}} + C \cdot \mathbf{l}\right), \tag{13}$$

where α is learning rate and $\mathbf{l} \sim \text{Logistic}(0, s\mathbf{I})$.

Then, what we need to discuss is the computation of accumulated privacy cost in Line 13 of Algorithm 1. We use the moments accountant, proposed by Abadi et al. (2016), for obtaining the accumulated privacy loss during the training.

Moments accountant The moments accountant has been widely used to track the privacy loss across multiple perturbed training steps with additive noises (generally Gaussian). From the obtained accumulated loss, the moments accountant enables us to find the value of ϵ or δ if the other is given.

For any arbitrary mechanism $\mathcal{M}$, let us define the divergence variable of two adjacent mini-batch stochastic gradients $\mathbf{g}$ and $\mathbf{g}'$, which only differ in a data instance:

$$c(\mathbf{v}; \mathcal{M}, \mathbf{g}, \mathbf{g}') \triangleq \log \frac{\Pr[\mathcal{M}(\mathbf{g}) = \mathbf{v}]}{\Pr[\mathcal{M}(\mathbf{g}') = \mathbf{v}]}. \tag{14}$$

The moments accountant is based on the moment composition, where the summation of moment generating function (MGF) of the divergence variable is used for computing accumulated privacy loss. The MGF with exponent λ is written as

$$\begin{aligned}\alpha_{\mathcal{M}}(\lambda; \mathbf{g}, \mathbf{g}') :=& \\ &\mathbb{E}_{\mathbf{v} \sim \mathcal{M}(\mathbf{g})}\left[\exp(\lambda c(\mathbf{v}; \mathcal{M}, \mathbf{g}, \mathbf{g}'))\right].\end{aligned} \tag{15}$$

For the privacy guarantees, we have to find the worst-case MGF for all adjacent gradients $\mathbf{g}$ and $\mathbf{g}'$ as

$$\alpha_{\mathcal{M}}(\lambda) = \max_{\mathbf{g}, \mathbf{g}'} \alpha_{\mathcal{M}}(\lambda, \mathbf{g}, \mathbf{g}'). \tag{16}$$

Then, the moments accountant defined in Definition 3.4 can obtain the upperbound of the accumulated privacy loss (composability) and the value of δ from the accumulated privacy loss (tail bound).

Definition 3.4 (Moments accountant (Abadi et al., 2016))**.** Let $\alpha_M(\lambda)$ be defined as above. Then

1. *[Composability]* Suppose that a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where $\mathcal{M}_i : \prod_{j=1}^{i-1} \mathbb{R}^N \times \mathcal{X}^n \to \mathbb{R}^N$. Then, for any λ

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{M}_i}(\lambda). \tag{17}$$

2. *[Tail bound]* For any $\varepsilon > 0$, the mechanism $\mathcal{M}$ is (ε, δ)-differentially private for

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon). \tag{18}$$

The proof can be found in DP-SGD.

λ-th order MGF of Logistic mechanism In Algorithm 2, the proposed method utilizes Logistic mechanism for perturbing the gradient. For obtaining the accumulated privacy loss, we should take manual integration in (16) to achieve tighter privacy loss bound. If the gradient of each data instance is clipped by C, the MGF $\alpha_{\mathcal{E}}(\lambda)$ is bounded by

$$\begin{aligned}\alpha_{\mathcal{E}}(\lambda) \leq & -N \cdot \log\left(1 + \exp(-G)\right) \\ & + N \log\left(1 + \exp(-(1 + 2\lambda)G)\right) \\ & + N\lambda G,\end{aligned} \tag{19}$$

where q is ratio of batch size $q = |\mathcal{B}|/n$ and $G = q/(2s\sqrt{N})$. We use this MGF formulation in our experiments.

3.4. Theoretic Analysis

Here, we show the theoretic superiority of the proposed method compared to DP-SIGNSGD.

Momentum generating function We first compare the MGF (16) of the proposed method and DP-SIGNSGD. In this analysis, we assume that the variance of Gaussian noise in DP-SIGNSGD is bounded by $\sigma < \frac{1}{16q}$. Under this assumption, Abadi et al. (2016) have proved that the MGF of DP-SIGNSGD is bounded by $\lambda(\lambda+1)q^2/(1-q)\sigma^2$. For fairness of the comparison, we equivalently assume that the scale of the additive Logistic mechanism is bounded by $s < \frac{\sqrt{3}}{16\pi q}$, because the variance of Logistic distribution with a scale s is $\frac{\pi^2 s^2}{3}$. In Theorem 3.5, we show that the MGF is bounded by $\frac{\lambda(\lambda+1)q^2}{50s^2}$ under this assumption, which is significantly tighter than that of DP-SIGNSGD.

Theorem 3.5 (Asymptotic bound of $\alpha_{\mathcal{M}}(\lambda)$)**.** *If $s < \frac{\sqrt{3}}{16\pi q}$, $\alpha_{\mathcal{M}}(\lambda)$ with the Logistic mechanism is bounded by*

$$\alpha_{\mathcal{E}}(\lambda) \leq \frac{\lambda(\lambda+1)q^2}{50s^2}. \tag{20}$$

Trainable epochs This result indicates that the proposed method has a significantly smaller noise scale compared to DP-SIGNSGD ($s = \frac{\sigma}{\sqrt{50}}$). In DP-SGD, because the noise scale is inversely proportional to $\sqrt{T}$, where T denotes the number of total trainable steps. Thus, our method can 50x more training steps with the obtained bound. We note that the MGF bound is not used for practical implementation; thus, with the numeric integration of MGF, the proposed method can have 1.5x more training epochs (see Figure 2).

Convergence of SIGNSGD with additive noise Next, we show the ℓ_1 convergence of SIGNSGD with arbitrary additive noises. In Theorem 3.6, by assuming β-smoothness, we show that the ℓ_1 convergence is bounded by SIGNSGD convergence term (first term) and the additive noise term (second term). Our focus is on the second term, in which the proposed method requires a smaller noise variance (≈20 times) compared to the DP-signSGD. Thus, the proposed method can have much faster convergence.

Theorem 3.6 (Convergence of SIGNSGD with additive noise). *For the loss function $\mathcal{L}$, let us assume that the parameter θ satisfies $\vec{\beta}$-smoothness. If* SIGNSGD *optimizer is implemented with zero-mean independent additive noise, the ℓ_1 convergence of the gradient is bounded by*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|_1] \\ \leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\vec{\beta}\|_1}\left(\mathcal{L}(\theta_0)-\mathcal{L}^*+\frac{1}{2}\right)+2\xi\right] \\ +2\frac{\sqrt{\mathrm{Tr}(\mathbf{N})}}{\sqrt{T}},\end{aligned} \tag{21}$$

where T denotes the total number of trainable steps, $\mathbf{N}$ *denotes the covariance matrix of additive noise, and ξ is a constant related to boundedness of gradients.*

Accuracy analysis Here, we aim to find the upperbound of the error rate of sign sampling. In the proposed method, the sign of the proposed method has an error if $\mathtt{sign}(x+l) \neq \mathtt{sign}(x)$, where $l \sim \mathrm{Logistic}(0,s)$. Then, the error rate of the proposed method is bounded by

$$\Pr[\mathtt{sign}(x+l)\neq\mathtt{sign}(x)] \leq \frac{1}{1+e^{\sqrt{50}|x|/\sigma}}. \tag{22}$$

On the other hand, in DP-SIGNSGD, the error rate is bounded by

$$\Pr[\mathtt{sign}(x+n_{\text{gau}})\neq\mathtt{sign}(x)] \leq \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{23}$$

where $n_{\text{gau}} \sim \mathcal{N}(0,\sigma^2)$. From Equations (22) and (23), we confirm that the proposed method has a lower error rate if $|x| < 14\sigma$. If $|x| > 14\sigma$, the error probability is almost zero as it is an extreme case in the probability distribution, approximately $1.56 \cdot 10^{-44} \approx 0$. Thus, we show that the proposed method has a higher chance to correctly choose the sign of the gradient in general ($|x| < 14\sigma$).

4. Numerical Results

In this section, we compare the proposed method and DP-SIGNSGD. We have implemented both methods in Pytorch, where the source code is available in https://github.com/jonggyujang0123/sign-dp-sgd. The experiments are done for the two datasets (MNIST (Cohen et al., 2017) and CIFAR-10 (Krizhevsky, 2009)) with a custom 3-layer fully connected layer (Dense), CNN models (ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2015)), and vision transformer (ViT) (Dosovitskiy et al., 2021). For comparison, we use the DP-SIGNSGD in (Jin et al., 2020) as a baseline scheme. Further implementation details are available in Appendix B. For CIFAR-10 dataset results, please refer to Appendix E.

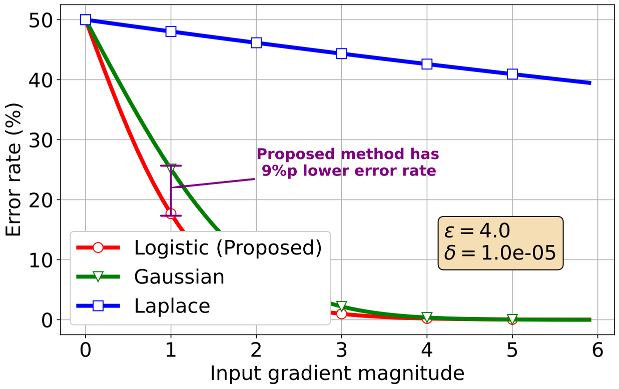


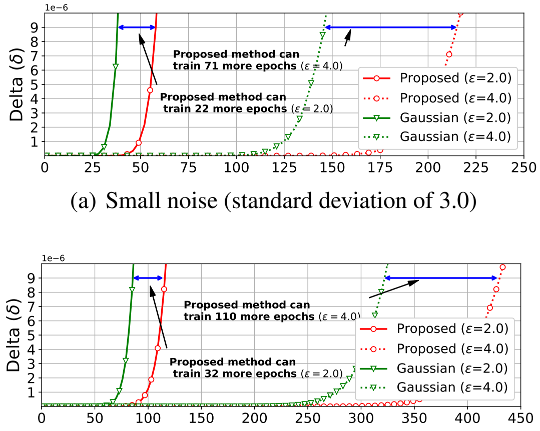
Figure 1. The sign error rate of the DP-signSGD methods. The noise scale of all the methods used in this experiment is configured to satisfy ($\epsilon = 4.0, \delta = 10^{-5}$)-DP.

4.1. Experiment 1: Privacy Accountant

In the previous section, we theoretically show that the proposed scheme has a tighter bound of privacy loss with the same noise variance and higher gradient accuracy. In previous studies (Abadi et al., 2016; Jin et al., 2020), the moments accountant computes the accumulated MGF of the privacy loss via empirical integration of the probability distribution; thus, we empirically compare the proposed method and the DP-SIGNSGD with the gradient accuracy (in Figure 1) and accumulated privacy loss (in Figure 2).

In this experiment, we assume that the number of total training steps is 10^4, and the batch size ratio $q = L/N$ is 0.01. We set the following privacy parameters ($\epsilon \in \{2.0, 4.0\}, \delta = 10^{-5}$). The number of trainable parameters is assumed to be one for brevity.

Error rate. In Figure 1, we show the gradient's sign error rate with respect to the input gradient magnitude. For comparison, we use Gaussian and Laplace mechanisms with moments accountant. The standard deviation (std) of the additive Logistic, Gaussian, and Laplace mechanisms are 1.17, 1.48, and 70.7, respectively. That is, the proposed method (Logistic) has a smaller std compared to baseline



(a) Small noise (standard deviation of 3.0)

(b) Large noise (standard deviation of 6.0)

Figure 2. Results on the value of δ for different noise levels on moments accountant. In all the experiments, two different target ϵ is used, and the target δ is set to be 10^{-5}. Also, the number of trainable parameters is assume to be one.

methods. As depicted in the figure, the proposed method has the smallest error rate of selecting the sign of the gradient. Since the Laplace mechanism is vulnerable to guarantee a tight privacy loss in the composition setting, we only compare DP-SIGNSGD using the Gaussian mechanism in the remainder part.

Accumulated privacy loss. By setting the std of all the noise distributions the same, we draw figures of the accumulated privacy loss in Figure 2. As in Definition 3.4, we can measure δ for given ϵ and $\alpha_M(\lambda)$. With the noise std of 3.0, we depict the value of δ obtained via moments accountant in Figure 2(a). In the figure, the proposed method can have 22 and 71 more training epochs than the DP-SIGNSGD for ϵ of 2.0 and 4.0, respectively. Similarly, if the noise std is fixed to 6.0, the proposed method has 32 and 110 more training epochs than the DP-SIGNSGD for ϵ of 2.0 and 4.0, respectively.

By doing these experiments, we confirm that the proposed method's tight privacy loss enables 1) a smaller error in sign gradient than the additive Gaussian mechanism under the same setting and 2) more training epochs with the same std of the additive noise.

4.2. Experiment 2: MNIST

In this experiment, we train various neural network models for the MNIST dataset. In Figure 3, we compare the proposed method and the DP-SIGNSGD by varying a) ϵ, b) δ, c) batch size, d) total number of epochs, and e) gradient clipping constant. The most important result is that the proposed method outperforms the DP-SIGNSGD for all hyper-parameter changes.

Table 1. MNIST classification accuracy for various neural network models.

MODELS	PROPOSED		DP-SIGNSGD	
	TRAINING	TEST	TRAINING	TEST
DENSE	96.14%	95.78%	92.31%	92.55%
RESNET-10	98.36%	98.49%	96.07%	96.48%
RESNET-18	98.73%	98.70%	96.77%	96.490%
RESNET-34	98.61%	98.55%	96.60%	96.83%
RESNET-50	98.12%	98.17%	94.60%	94.92%
VGG11	98.87%	98.95%	97.45%	97.66%
VGG13	98.97%	99.03%	97.79%	97.98%
VGG16	98.91%	98.80%	97.44%	97.63%
VGG19	98.78%	98.80%	97.41%	97.58%
VIT-B-16	99.06%	99.01%	84.72%	85.46%

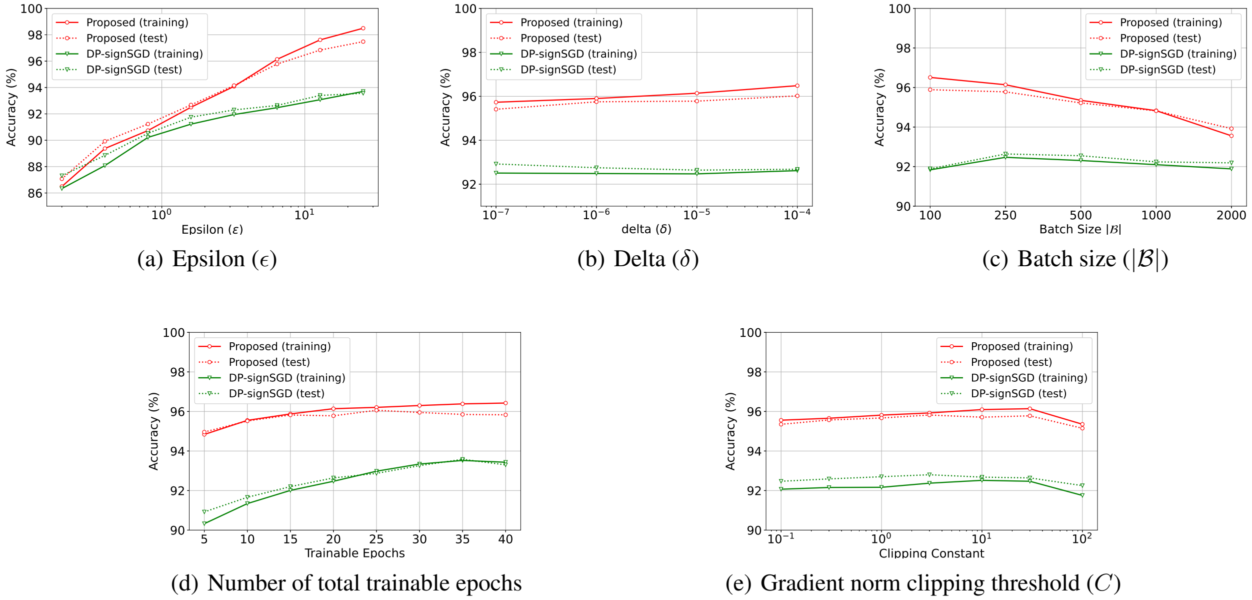
Privacy budget In Figures 3(a) and 3(b), we depict the training/test accuracy by varying the value of privacy budget parameters ϵ and δ. As the privacy budget is loosened (ϵ and $\delta \uparrow$), the training/test accuracy is gradually increased. Specifically, in Figure 3(a), the gap between the proposed method and DP-SIGNSGD increases as ϵ increases. However, as $\epsilon \to \infty$, the accuracy of both methods converge as the scale of the noise goes to zero[3].

Learning parameters In Figure 3(c), we depict the accuracy for varying batch size. In this figure, we can find that the batch size suggestion in (Abadi et al., 2016) still holds ($|\mathcal{B}| \approx \sqrt{N}$). In other figures (Figures 3(d) and 3(e)), we show that the gradient clipping constant and the number of epochs do not significantly affect the trained model if a sufficient number of epochs are given.

Various neural network models Table 1 shows the MNIST classification accuracies for various neural network models, including custom dense network, ResNet, VGG, and ViT. Here, we note that the privacy budget parameter is fixed to $(\epsilon, \delta) = (6.4, 10^{-5})$. In the results, the proposed method has training/test accuracies compared to DP-SIGNSGD for all models, even with the same hyper-parameter settings.

Through these experiments, we show that the proposed method can outperform DP-SIGNSGD by replacing the Gaussian noise with more appropriate noise, Logistic noise.

[3] We implement both methods with $\epsilon > 10^{10}$ and confirm that the training and test accuracy converges around 99.85% and 98.22%, respectively.



(a) Epsilon (ϵ)

(b) Delta (δ)

(c) Batch size ($|\mathcal{B}|$)

(d) Number of total trainable epochs

(e) Gradient norm clipping threshold (C)

Figure 3. MNIST classification accuracy when one hyper-parameter varies, and the others are fixed at the reference values in Appendix B.

5. Discussion

Logistic vs. Gaussian The Logistic and Gaussian distributions are symmetric, bell-shaped, and log-concave distributions. The tail distribution is important because the concept of DP is bounding the divergence variable in tails with probabilistic margin δ. In tail distribution, the Logistic distribution has a heavier tail compared to the Gaussian distribution. As shown in (Vinterbo, 2022), by virtue of its heavier tail, the Logistic mechanism has a lower ℓ_2 error than the Gaussian mechanism when δ is small. More intuitively, the Logistic mechanism does not always require positive δ because its tail distribution decays sub-exponentially. On the other hand, the Gaussian mechanism, whose tail decays faster than exponential, always requires a margin $\delta > 0$.

Positive effect This work paves the way for further research into identifying optimal noise distributions for various optimization methods. While we have focused on the integration of additive Logistic noise in DP-SIGNSGD, our findings suggest broader applications and potential improvements in differential privacy. The incorporation of this noise type not only reduces privacy loss but also contributes to the overall robustness and efficiency of the learning process.

6. Conclusion

In this paper, we have demonstrated the effectiveness of additive Logistic noise in reducing accumulated privacy loss in differentially private sign-based stochastic gradient descent (DP-SIGNSGD), leading to the development of DP-SIGNLOSGD. The primary advantage of DP-SIGNLOSGD over traditional DP-SIGNSGD, which utilizes additive Gaussian noise, is its ability to support training over more epochs without significantly compromising privacy. This feature makes DP-SIGNLOSGD a promising tool for ensuring differential privacy in federated and distributed learning methodologies (Jin et al., 2024).

Our proposed framework introduces an innovative approach to differentially private sign-based SGD, utilizing an exponential mechanism for sign sampling to identify the most effective direction for gradient descent. This method is theoretically equivalent to the addition of Logistic noise prior to the `sign` function. The comprehensive nature of our theoretical analysis highlights the superiority of DP-SIGNLOSGD in terms of convergence speed, accuracy, and the tightness of privacy loss bounds.

Future works Looking forward, our research will extend into exploring majority vote algorithms (Bernstein et al., 2019) and n-bit differentially private gradient compression (Lin, 2022; Kerkouche et al., 2021). These areas offer significant potential for advancing the field of differential privacy in machine learning. By focusing on these domains, we aim to further enhance the efficiency and effectiveness of privacy-preserving methodologies, contributing to the development of more secure and reliable machine learning systems.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgment

This research was supported in part by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-2021-0-02048) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation), in part by the research projects of Development of automatic screening and hybrid detection system for hazardous material detecting in port container financially (1525013872, No. 20200611) supported by the Ministry of Oceans and Fisheries, Korea, and in part by the IITP(Institute for Information & Communications Technology Planning & Evaluation), grant funded by MSIT(Ministry of Science and ICT) (RS-2023-00229541, Development of Big Data and Artificial Intelligence Based Radio Monitoring Platform).

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *CCS*, 2016.

Andrew, G., Kairouz, P., Oh, S., Oprea, A., McMahan, H. B., and Suriyakumar, V. One-shot empirical privacy estimation for federated learning, February 2023.

Apple, D. P. T. Learning with privacy at scale. In *Apple Machine Learning, Journal*, volume 1, pp. 1–25, 2017.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. signSGD with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*, 2019.

Chen, X., Chen, T., Sun, H., Wu, S. Z., and Hong, M. Distributed training with heterogeneous data: Bridging median-and mean-based algorithms. *Advances in Neural Information Processing Systems*, 33:21616–21626, 2020.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. EMNIST: anextension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

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

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*, 2021.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Proceedings of Advances in Neural Information Processing Systems*, 2020.

Galli, F., Palamidessi, C., and Cucinotta, T. Online sensitivity optimization in differentially private learning. *arXiv preprint arXiv:2310.00829*, 2023.

Girgis, A. M., Data, D., Diggavi, S., Suresh, A. T., and Kairouz, P. On the renyi differential privacy of the shuffle model. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2321–2341, 2021.

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

Heo, G., Seo, J., and Whang, S. E. Personalized DP-SGD using sampling mechanisms. *arXiv preprint arXiv:2305.15165*, 2023.

Hong, J., Wang, J. T., Zhang, C., Li, Z., Li, B., and Wang, Z. DP-OPT: Make large language model your privacy-preserving prompt engineer. *arXiv preprint arXiv:2312.03724*, 2023.

Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private SGD?, June 2020.

Jin, R., Huang, Y., He, X., Dai, H., and Wu, T. Stochastic-sign sgd for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940*, 2020.

Jin, R., Liu, Y., Huang, Y., He, X., Wu, T., and Dai, H. Sign-based gradient descent with heterogeneous data:

Convergence and byzantine resilience. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2024. ISSN 2162-2388. doi: 10.1109/tnnls.2023.3345367. URL http://dx.doi.org/10.1109/TNNLS.2023.3345367.

Kerkouche, R., Ács, G., Castelluccia, C., and Genevès, P. Compression boosts differentially private federated learning. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 304–318, 2021. doi: 10.1109/EuroSP51992.2021.00029.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Lee, E., Lee, J.-W., Lee, J., Kim, Y.-S., Kim, Y., No, J.-S., and Choi, W. Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12403–12422. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/lee22e.html.

Lin, J. On the interaction between differential privacy and gradient compression in deep learning. *arXiv preprint arXiv:2211.00734*, 2022.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Lyu, L. Dp-signsgd: When efficiency meets privacy and robustness. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3070–3074, 2021. doi: 10.1109/ICASSP39728.2021.9414538.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H., and Shin, J. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349444. doi: 10.1145/3052973.3053009. URL https://doi.org/10.1145/3052973.3053009.

Sha, H., Liu, R., Liu, Y., and Chen, H. PCDP-SGD: Improving the convergence of differentially private SGD via projection in advance, December 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2015.

Song, C., Ristenpart, T., and Shmatikov, V. Machine learning models that remember too much. In *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 587–601, October 2017.

Vinterbo, S. A. Differential privacy for symmetric log-concave mechanisms. In *International Conference on Artificial Intelligence and Statistics*, pp. 6270–6291. PMLR, 2022.

Zhu, J. and Blaschko, M. B. Improving differentially private sgd via randomly sparsified gradients. *arXiv preprint arXiv:2112.00845*, 2021.

A. Proof of Theorem 3.5

Theorem 3.5. (Asymptotic bound of $\alpha_{\mathcal{E}}(\lambda)$) *If $s < \frac{\sqrt{3}}{16\pi q}$, the MGF of the Logistic mechanism $\alpha_{\mathcal{E}}(\lambda)$ is bounded by*

$$\alpha_{\mathcal{E}}(\lambda) \leq \frac{\lambda(\lambda+1)q^2}{50s^2}. \tag{24}$$

Proof. Before proving Theorem 3.5, we first address the MGF of the privacy loss divergence variable. From the definition in (16), we have

$$\alpha_{\mathcal{E}}(\lambda) = \max_{\mathbf{g},\mathbf{g}'} \log \mathbb{E}_{\mathbf{v}\sim\mathcal{E}(\mathbf{g})}\left[\exp\left(\lambda \log \frac{\Pr(\mathcal{E}(\mathbf{g}) = \mathbf{v})}{\Pr(\mathcal{E}(\mathbf{g}') = \mathbf{v})}\right)\right]. \tag{25}$$

Because each element of the $\mathcal{E}(\mathbf{g})$ is independently distributed, the MGF in (25) can be rewritten as

$$\alpha_{\mathcal{E}}(\lambda) = \max_{\mathbf{g},\mathbf{g}'} \sum_{i=1}^{N} \log \mathbb{E}_{v_i\sim\mathcal{E}(g_i)} \exp\left(\lambda \log \frac{\Pr(\mathcal{E}(g_i) = v_i)}{\Pr(\mathcal{E}(g_i') = v_i)}\right), \tag{26}$$

where $\|\mathbf{g}-\mathbf{g}'\|_2 \leq C$. Because $\mathbf{g} = -\mathbf{g}' = \frac{C}{2\sqrt{N}} \cdot \mathbf{1}$ is the maximizer in (26), we can obtain $\alpha_{\mathcal{E}}(\lambda)$ for DP-SIGNLOSGD as follows:

$$\begin{aligned}
&\alpha_{\mathcal{E}}(\lambda) \\
&= N \cdot \log\left(\frac{\exp\left(\frac{q}{4s\sqrt{N}}\right)}{\exp\left(\frac{q}{4s\sqrt{N}}\right) + \exp\left(\frac{-q}{4s\sqrt{N}}\right)} \cdot \exp\left(\frac{\lambda q}{2s\sqrt{N}}\right) + \frac{\exp\left(\frac{-q}{4s\sqrt{N}}\right)}{\exp\left(\frac{q}{4s\sqrt{N}}\right) + \exp\left(\frac{-q}{4s\sqrt{N}}\right)} \cdot \exp\left(\frac{-\lambda q}{2s\sqrt{N}}\right)\right) \\
&= N \cdot \log\left(\frac{\exp\left(\frac{q}{4s\sqrt{N}}\right)}{\exp\left(\frac{q}{4s\sqrt{N}}\right) + \exp\left(\frac{-q}{4s\sqrt{N}}\right)}\right) + \frac{N\lambda q}{2s\sqrt{N}} + N\log\left(1 + \exp\left(-(1+2\lambda)\frac{q}{2s\sqrt{N}}\right)\right).
\end{aligned} \tag{27}$$

In (27), we obtain a closed-form representation of the numeric integration in (16). To find the theoretic upperbound of (27) in a similar form to the original DP-SGD ($K\lambda^2 q^2/\sigma^2$ for arbitrary constant K), we bring the following assumption from (Abadi et al., 2016): $\sigma < \frac{1}{16q}$. For brevity of the notation, we define an auxiliary variable $G = \frac{q}{2s\sqrt{N}}$. Since variance of Logistic distribution with scale s is $\pi^2 s^2/3$, we rewrite the assumption by $s < \frac{\sqrt{3}}{16\pi q}$. With this assumption, we can derive the upperbound of $\alpha_{\mathcal{E}}(\lambda)$ as

$$\begin{aligned}
\alpha_{\mathcal{M}}(\lambda) &= N \cdot \log\left(\frac{\exp(\lambda G)\exp(G/2) + \exp(-\lambda G)\exp(-G/2)}{\exp(G/2) + \exp(-G/2)}\right) \\
&< N \cdot \log(\exp(\lambda G) + \exp(-\lambda G)) \\
&\overset{(a)}{\leq} N \cdot \log(\exp(\frac{\lambda^2 G^2}{50})) \\
&= N \cdot \frac{\lambda^2 G^2}{50} = \frac{\lambda^2 q^2 C^2}{50s^2},
\end{aligned} \tag{28}$$

where the inequality (a) holds if $s < \frac{\sqrt{3}}{16\pi q}$. □

B. Implementation details

Algorithm 2 DP-signLoSGD

```
**Input**: Dataset $\mathcal{D} = \{z_1, z_2, ..., z_n\}$, loss function $\mathcal{L}(\theta) = \sum_i \mathcal{L}(\theta, z_i)$.
**Parameters**: noise scale $\sigma$, batch size $B$, and gradient norm bound $C$.
**Initialize** $\theta_0$ randomly
**for** $t \in [T]$ **do**
    Sample a random batch from $\mathcal{B}_t$.
    **for** $i \in \mathcal{B}_t$ **do**
        $\tilde{\mathbf{g}}_t(z_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$ {Compute gradient}
        $\tilde{\mathbf{g}}_t(z_i) \leftarrow \tilde{\mathbf{g}}_t(z_i)/\max(1, \frac{\|\tilde{\mathbf{g}}_t(z_i)\|_2}{C})$ {Clipping}
    **end for**
    $\mathbf{g}_t \leftarrow \mathtt{sign}(\sum_i \tilde{\mathbf{g}}_t(z_i) + \text{Logistic}(0, sC\mathbf{I}))$
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \mathbf{g}_t$
**end for**
**Output**: $\theta_T$ and computed privacy cost $(\epsilon, \delta)$.
```

Hardware Our experiments are conducted at a workstation with 12th Gen Intel(R) Core(TM) i9-12900K 16-Core Processor CPU @ 5.20GHz and one NVIDIA Geforce RTX 3090 GPU.

Hyper-parameters In our MNIST experiments, the neural network models except the ViT model are trained with the initial learning rate of $1.0 \cdot 10^{-3}$ and momentum of 0.0. The cosine learning rate scheduler is used. The gradient of each data instance is clipped[4] by 30.0, i.e., $C = 30$, and the reference batch size is 250. The MNIST images are resized to 32x32 images. For the ViT model, we fine-tune the ViT-B-16 model for 5 epochs with a batch size of 45, in which the MNIST images are resized to 224. For privacy loss parameters, $\epsilon = 6.4$ and $\delta = 10^{-5}$ are used as reference values. The custom dense neural network consists of a three-layer fully connected layer, each of which has 512 neurons and is activated by a rectified linear unit (ReLU). During the experiment, the custom dense neural network is used as a reference model.

Modification for running DP-signLoSGD We use Pytorch 2.0 library in our experiment, where `torch.func.vmap` function is used to get a data-wise gradient. In our implementation, because `torch.func.vmap` does not support the batch normalization module, we replace all the batch normalization modules with group normalization. We confirm that this modification does not degrade the accuracy of the trained model. Also, to train VGG and ResNet for 32x32 images, we use slightly modified versions of them[5]. For further details, please refer to our source code.

[4] In the camera-ready revision, numerical results corresponding to the per-layer norm are fixed to the global norm, where performance gap between the baseline and ours is almost the same.

[5] https://github.com/kuangliu/pytorch-cifar

C. Proof of Theorem 3.6

Theorem 3.6. (Convergence of signSGD with additive noise) *For the loss function $\mathcal{L}$, let us assume that the parameter θ satisfies $\vec{\beta}$-smoothness. If* signSGD *optimizer is implemented with zero-mean independent additive noise, the ℓ_1 convergence of the gradient is bounded by*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|_1] \leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\vec{\beta}\|_1}\left(\mathcal{L}(\theta_0)-\mathcal{L}^*+\frac{1}{2}\right)+2\xi\right]+2\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{T}} \tag{29}$$

where T denotes the total number of trainable steps, $\mathbf{A}$ denotes the covariance matrix of additive noise, and ξ is a constant related to boundedness of gradients.

C.1. Assumptions and Lemmas for convergence analysis

To prove the convergence of signSGD with zero-mean additive noise, we have assumed the following assumption. In Assumption C.1, let us denote the loss function as $\mathcal{L}(\theta)=\sum_k l(\theta,z_k)$ with trainable parameter $\theta\in\mathbb{R}^N$, where z_k denotes k-th data in the training dataset. Then, we first assume that the loss function $\mathcal{L}(\theta)$ is lower-bounded by a constant $\mathcal{L}^*$. For instance, the generally used loss functions (e.g., negative log-likelihood loss, cross-entropy loss, and mean square error loss) are lower-bounded by zero.

Assumption C.1 (Lower bounded loss function)**.** For all θ and some constant $\mathcal{L}^*$, the loss function $\mathcal{L}(\theta)$ is lower-bounded by $\mathcal{L}^*$, i.e., $\mathcal{L}(\theta)\geq\mathcal{L}^*,\forall\theta$.

In addition to this assumption, we also assume element-wise β-smoothness on $\mathcal{L}(\theta)$ for all θ.

Assumption C.2 (Element-wise β-smoothness on $l(\theta,z_k)$)**.** For all θ,θ' and positive values $\beta_1,\beta_2,...,\beta_N$, the differentiable loss function $\mathcal{L}(\theta)$ is $\vec{\beta}$-smoothness, i.e.,

$$\mathcal{L}(\theta')\leq\mathcal{L}(\theta)+\nabla\mathcal{L}(\theta)^{\mathrm{T}}(\theta'-\theta)+\sum_{k=1}^{N}\frac{\beta_k}{2}|[\theta']_k-[\theta]_k|^2 \tag{30}$$

where k denotes the element index, and $\vec{\beta}=[\beta_1,\cdots,\beta_N]$.

We note that the element-wise $\vec{\beta}$-smoothness function also satisfies the original β-smoothness by setting $\beta=\max_{k=1,...,N}\beta_k$. Along with the smoothness assumption, we assume the ℓ_2 norm and variance of the stochastic gradient g are bounded.

Assumption C.3 (Bounded variance)**.** For all θ, a stochastic gradient $\mathbf{g}$ on the θ, and the expected gradient $\nabla\mathcal{L}(\theta)$, the variance of $\mathbf{g}$ is bounded by

$$\mathbb{E}\left[\|\mathbf{g}_t(z_i)-\nabla\mathcal{L}(\theta)\|_2^2\middle|\theta\right]\leq\eta. \tag{31}$$

where η is constant.

Assumption C.4 (Boundness of stochastic gradient)**.** For all θ, the ℓ_2 norm of the stochastic gradient $\mathbf{g}$ on the parameter θ is bounded as

$$\|\mathbf{g}_t(z_i)\|_2^2\leq G_2. \tag{32}$$

Lemma C.5. *(Bounded expectation of difference between clipped stochastic gradient and expected gradient) For a certain θ, the clipped stochastic gradient* `Clip`*($\mathbf{g}$), and the expected gradient $\nabla\mathcal{L}(\theta)$, the following inequality holds:*

$$\mathbb{E}\left[\left\|\frac{\sum_{i\in\mathcal{B}_t}\mathtt{Clip}(\mathbf{g}_t(z_i))}{|\mathcal{B}_t|}-\nabla\mathcal{L}(\theta)\right\|_2^2\middle|\theta\right]\leq\frac{\xi^2}{|\mathcal{B}_t|}. \tag{33}$$

where $\xi^2=G_2+\eta$.

Proof. To prove Lemma C.5, we have the following series of inequalities:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\frac{\sum_{i\in\mathcal{B}_t}\texttt{Clip}(\mathbf{g}_t(z_i))}{|\mathcal{B}_t|}-\nabla\mathcal{L}(\theta)\right\|_2^2\middle|\theta\right]\\
&=\frac{1}{|\mathcal{B}_t|^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{B}_t}\texttt{Clip}(\mathbf{g}_t(z_i))-\mathbf{g}_t(z_i)+\mathbf{g}_t(z_i)-\nabla\mathcal{L}(\theta)\right\|_2^2\middle|\theta\right]\\
&\leq\frac{1}{|\mathcal{B}_t|^2}\underbrace{\mathbb{E}\left[\left\|\sum_{i\in\mathcal{B}_t}\texttt{Clip}(\mathbf{g}_t(z_i))-\mathbf{g}_t(z_i)\right\|_2^2\middle|\theta\right]}_{\leq\sum_{i\in\mathcal{B}_t}\mathbb{E}[\|\mathbf{g}_t(z_i)\|]_2^2\leq|\mathcal{B}_t|G_2,\text{by Assumption C.4}}+\frac{1}{|\mathcal{B}_t|^2}\underbrace{\mathbb{E}\left[\left\|\sum_{i\in\mathcal{B}}\mathbf{g}_t(z_i)-\nabla\mathcal{L}(\theta)\right\|_2^2\middle|\theta\right]}_{\leq|\mathcal{B}_t|\eta,\text{ by Assumption C.3}}\\
&\leq\frac{G_2+\eta}{|\mathcal{B}_t|}=\frac{\xi^2}{|\mathcal{B}_t|}.
\end{aligned}
\tag{34}
$$

□

C.2. Proof of Theorem 3.6

Let us consider $\theta_1, ..., \theta_T$ updated by SIGNSGD with zero-mean additive noise $\mathbf{a}$, where its covariance matrix is $\mathbf{A}$, i.e.,

$$
\theta_{t+1}\leftarrow\theta_t-\alpha_t\cdot\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a}). \tag{35}
$$

where $\tilde{\mathbf{g}}$ denotes the clipped gradient $\texttt{Clip}(\mathbf{g})$. As we assumed element-wise $\vec{\beta}$-smoothness in Assumption C.2, for parameters θ_{t+1} and θ_t, we have

$$
\mathcal{L}(\theta_{t+1})\leq\mathcal{L}(\theta_t)+\langle\nabla\mathcal{L}(\theta_t),\theta_{t+1}-\theta_t\rangle+\sum_{k=1}^{N}\frac{\beta_k}{2}|[\theta_{t+1}]_k-[\theta_t]_k|^2 \tag{36}
$$

Because $\theta_{t+1}-\theta_t=-\alpha_t\cdot\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a})$, we can show that the following inequalities hold:

$$
\begin{aligned}
\mathcal{L}(\theta_{t+1})\leq\ &\mathcal{L}(\theta_t)-\alpha_t\sum_{k=1}^{N}[\nabla\mathcal{L}(\theta_t)]_k[\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a})]_k+\sum_{k=1}^{N}\frac{\alpha_t^2\beta_k}{2}\underbrace{|\texttt{sign}[(\tilde{\mathbf{g}}+\mathbf{a})]_k|^2}_{=1}\\
=\ &\mathcal{L}(\theta_t)-\alpha_t\sum_{k=1}^{N}[\nabla\mathcal{L}(\theta_t)]_k\ [\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a})]_k+\sum_{k=1}^{N}\frac{\alpha_t^2\beta_k}{2}\\
=\ &\mathcal{L}(\theta_t)-\alpha_t\|\nabla\mathcal{L}(\theta_t)\|_1+\alpha_t^2\frac{||\vec{\beta}||_1}{2}\\
&+2\alpha_t\sum_{k=1}^{N}|[\nabla\mathcal{L}(\theta_t)_k]|\cdot\mathbb{I}\left([\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a})]_k\neq[\texttt{sign}(\nabla\mathcal{L}(\theta_t))]_k\right),
\end{aligned}
\tag{37}
$$

where $\mathbb{I}(\cdot)$ is the indicator function. By applying $\mathbb{E}_{\mathbf{a},\mathcal{B}_t}[\cdot|\theta_t]$ to both side of Equation (37), we have

$$
\begin{aligned}
\mathbb{E}[\mathcal{L}(\theta_{t+1})|\theta_t]\leq\mathcal{L}(\theta_t)-\eta_t\|\nabla\mathcal{L}(\theta_t)\|_1+\alpha_t^2\frac{||\vec{\beta}||_1}{2}&\\
+2\eta_t\sum_{k=1}^{N}|[\nabla\mathcal{L}(\theta_t)]_k|\Pr\left[[\texttt{sign}(\tilde{\mathbf{g}}+\mathbf{a})]_k\neq[\texttt{sign}(\nabla\mathcal{L}(\theta_t))]_k|\theta_t\right].&
\end{aligned}
\tag{38}
$$

Then, our next focus is on obtaining the bound of the fourth term in (38). Before this, we have the following brief notations: each element of $\tilde{\mathbf{g}}$ as g_k, each element of $\nabla\mathcal{L}(\theta_t)$ as $\nabla\mathcal{L}(\theta_t)_k$, and each element of $\mathbf{a}$ as a_k, where k is element index. By ignoring the constant $2 \cdot \alpha_t$, and for a constant $B = |\mathcal{B}_t|$, we have

$$
\begin{aligned}
&\sum_{k=1}^{N} |\nabla\mathcal{L}(\theta_t)_k| \Pr\left[\texttt{sign}(g_k + a_k) \neq \texttt{sign}(\nabla\mathcal{L}(\theta_t)_k)|\theta_t\right] \\
&= \sum_{k=1}^{N} |\nabla\mathcal{L}(\theta_t)_k| \;\Pr\left[\texttt{sign}\left(\frac{g_k + a_k}{B}\right) \neq \texttt{sign}(\nabla\mathcal{L}(\theta_t)_k)\middle|\theta_t\right] \\
&\leq \sum_{k=1}^{N} |\nabla\mathcal{L}(\theta_t)_k| \Pr\left[\left|\frac{g_k + a_k}{B} - \nabla\mathcal{L}(\theta_t)_k\right| \geq |\nabla\mathcal{L}(\theta_t)_k|\right] \\
&\overset{(a)}{\leq} \sum_{k=1}^{N} |\nabla\mathcal{L}(\theta_t)_k| \frac{\mathbb{E}\left[\left|\frac{g_k + a_k}{B} - \nabla\mathcal{L}(\theta_t)_k\right| \middle|\theta_t\right]}{|\nabla\mathcal{L}(\theta_t)_k|} \\
&= \sum_{k=1}^{N} \mathbb{E}\left[\left|\frac{g_k + a_k}{B} - \nabla\mathcal{L}(\theta_t)_k\right| \middle|\theta_t\right] \\
&\leq \sum_{k=1}^{N} \mathbb{E}\left[\left|\frac{g_k}{B} - \nabla\mathcal{L}(\theta_t)_k\right| \middle|\theta_t\right] + \sum_{k=1}^{N} \mathbb{E}\left[\left|\frac{a_k}{B}\right|\right] \\
&\leq \sqrt{\sum_{k=1}^{N} \mathbb{E}\left[\left(\frac{g_k}{B} - \nabla\mathcal{L}(\theta_t)_k\right)^2 \middle|\theta_t\right]} + \underbrace{\sqrt{\sum_{k=1}^{N} \mathbb{E}\left[\left(\frac{a_k}{B}\right)^2\right]}}_{=\sqrt{\mathrm{Tr}(\mathbf{A}/B)}} \\
&= \sqrt{\mathbb{E}\left[\left\|\frac{\tilde{\mathbf{g}}}{B} - \nabla\mathcal{L}(\theta_t)\right\|_2^2 \middle|\theta_t\right]} + \sqrt{\frac{\mathrm{Tr}(\mathbf{A})}{B}} \\
&\overset{(b)}{\leq} \frac{\xi}{\sqrt{B}} + \frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{B}},
\end{aligned}
\tag{39}
$$

where the inequality (a) follows Markov inequality, and inequality (b) holds from Lemma C.5.

By substituting the result in (39) in Equation (38), we have

$$
\begin{aligned}
\mathbb{E}[\mathcal{L}(\theta_{t+1})|\theta_t] \leq \mathcal{L}(\theta_t) - \alpha_t||\nabla\mathcal{L}(\theta_t)||_1 + \alpha_t^2\frac{\|\vec{\beta}\|_1}{2} \\
+ 2\alpha_t\frac{\xi}{\sqrt{B}} + 2\alpha_t\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{B}}.
\end{aligned}
\tag{40}
$$

According to total expectation,

$$
\begin{aligned}
\mathbb{E}[||\nabla\mathcal{L}(\theta_t)||_1] \leq \frac{(\mathbb{E}[\mathcal{L}(\theta_t)] - \mathbb{E}[\mathcal{L}(\theta_{t+1})])}{\alpha_t} + \alpha_t\frac{\|\vec{\beta}\|_1}{2} \\
+ 2\frac{\xi}{\sqrt{B}} + 2\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{B}}.
\end{aligned}
\tag{41}
$$

For a fixed learning rate $\alpha_t = \alpha$, by summing up the inequality (41) from $t = 0$ to $T - 1$, we have

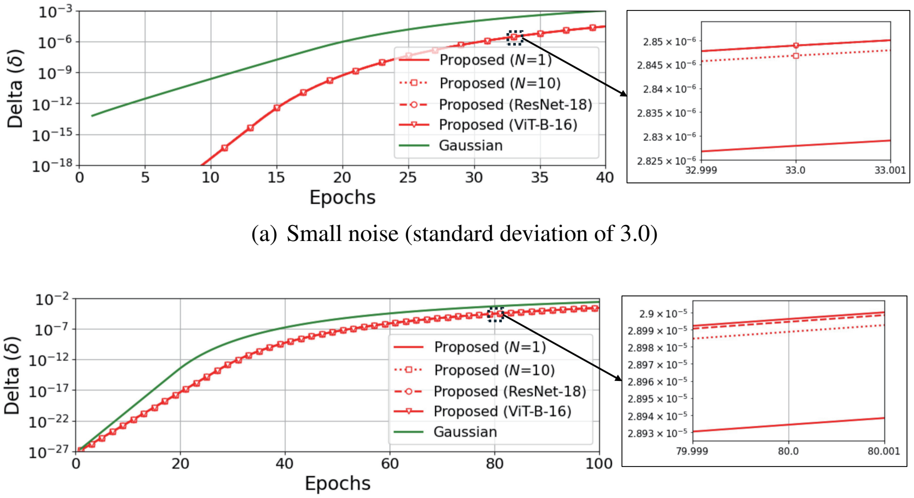
$$
\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla\mathcal{L}(\theta_t)\|_1\right] \leq \frac{\mathcal{L}(\theta_0) - \mathbb{E}[\mathcal{L}(\theta_T)]}{\alpha T} + \alpha\frac{\|\vec{\beta}\|_1}{2} + 2\frac{\xi}{\sqrt{B}} + 2\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{B}}.
\tag{42}
$$

Lastly, by substituting $\alpha = \frac{1}{\sqrt{T\|\vec{\beta}\|_1}}$ and $B = T$, and by using Assumption C.1, we conclude this proof as follows:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta_t)\|_1\right] \leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\vec{\beta}\|_1}\left(\mathcal{L}(\theta_0) - \mathbb{E}[\mathcal{L}(\theta_T)] + \frac{1}{2}\right) + 2\xi\right] + 2\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{T}} \tag{43}$$

$$\leq \frac{1}{\sqrt{T}}\left[\sqrt{\|\vec{\beta}\|_1}\left(\mathcal{L}(\theta_0) - \mathcal{L}^* + \frac{1}{2}\right) + 2\xi\right] + 2\frac{\sqrt{\mathrm{Tr}(\mathbf{A})}}{\sqrt{T}} \tag{44}$$

D. Privacy Loss Accumulation for Various Neural Network Models



(a) Small noise (standard deviation of 3.0)

(b) Large noise (standard deviation of 6.0)

Figure 4. Results on the value of δ for different noise levels on moments accountant. In all the experiments, the value of target ϵ is fixed to 2.0. Also, we bring practical experimental environment. (MNIST: number of data is 60,000, batch-size is 250)

Impact of the number of trainable parameters In Fig. 2, we operate under the assumption the number of trainable parameters is set to one, as it does not significantly impact the privacy accumulation of the proposed method. In addition to that result, we add the results for the privacy accumulation of various numbers of parameters. In Fig. 4, we illustrate the accumulated privacy loss in the same setting as our MNIST experiments, considering four scenarios with varying numbers of parameters: 1, 10, 5e9 (ResNet-18), and 5e10 (ViT) for curve plotting. The figure demonstrates that, irrespective of the parameter count, our proposed method consistently shows a significantly lower delta compared to DP-signSGD (Gaussian). For clarity, we include a magnified view of the figure, highlighting that the increase in delta with the number of trainable parameters in our method is negligible.

E. Numerical Results: CIFAR-10

In addition to the experiments with MNIST data, we have conducted supplementary experiments on the CIFAR-10 dataset. In our CIFAR-10 experiments, most hyper-parameters are the same as the MNIST experiments, except total epochs. We train all the neural networks except the ViT model for 50 epochs. The cosine learning rate scheduler is used. Unlike the MNIST experiment, we use the ResNet-18 model as the reference model. Also, the image augmentation with padding with 4 pixels and random crop 32x32 pixels is used.

Table 2. CIFAR-10 classification accuracies for various value of target ϵ.

ϵ	Proposed		DP-signSGD	
	training	test	training	test
0.4	32.73%	33.64%	30.74%	32.05%
0.8	37.45%	37.45%	35.20%	35.21%
1.6	44.90%	43.75%	40.95%	41.18%
3.2	54.20%	51.61%	47.64%	46.48%
6.4	62.57%	58.85%	51.71%	49.96%
12.8	69.41%	64.94%	54.42%	52.21%
25.6	74.70%	69.56%	57.62%	55.19%

Various target epsilon (ϵ) In Table 2, we show the CIFAR-10 classification accuracies for various values of ϵ. Similar to the result in Figure 3(a), the training/test accuracies are gradually increased as the privacy budget is loosened. More importantly, the gap between the proposed method and DP-signSGD is getting larger as ϵ increases.

Table 3. CIFAR-10 classification accuracies for various neural network models.

Models	Proposed		DP-signSGD	
	training	test	training	test
ResNet-10	59.47%	57.33%	49.25%	47.37%
ResNet-18	62.57%	58.85%	51.71%	49.96%
ResNet-34	60.73%	58.05%	49.30%	48.26%
VGG11	61.06%	62.03%	61.06%	62.03%
VGG13	61.44%	62.04%	61.44%	62.04%
VGG16	60.69%	61.56%	60.69%	61.56%
VGG19	58.94%	59.93%	58.94%	59.93%

Various neural network models In Table 3, the CIFAR-10 classification accuracies are indicated by varying the neural network models. The privacy budget is fixed to $\epsilon = 6.4$ and $\delta = 1 \cdot 10^{-5}$. For all models we used, the proposed method has a meaningful enhancement in both training/test accuracies than DP-signSGD. For the experiments without pre-trained neural network weights, the proposed method significantly outperforms DP-signSGD. Even if the neural network model is pre-trained with a large dataset (ImageNet), the proposed method still outperforms DP-signSGD.

Table 4. The proportional gap of the standard deviation of the noise added for guaranteeing various (ϵ, δ)-DP.

ϵ	Ours (A)	DP-signSGD (B)	$B - A$	$(B - A)/A$
0.4	5.48	6.11	0.63	0.11
0.8	2.76	3.15	0.39	0.14
1.6	1.40	1.71	0.31	0.22
3.2	0.72	1.05	0.33	0.46
6.4	0.38	0.76	0.38	1.00
12.8	0.21	0.60	0.39	1.85
25.6	0.11	0.49	0.38	3.45

Comparison of scale of additive noise As shown in the table, the proportional gap of the variance becomes larger as ϵ increases, indicating that the proposed method efficiently secures privacy loss. For instance, when $\epsilon = 0.4$, the standard deviations of the two methods are similar $\left(\frac{(B)-(A)}{(A)} = 0.11\right)$. However, when $\epsilon = 25.6$, the gap is much larger $\left(\frac{(B)-(A)}{(A)} = 3.45\right)$. Thus, our method have a more significant improvement for larger ϵ.

Table 5. (Fine-tuning) CIFAR-10 image classification results for various ϵ values. The model used in this experiment is ResNet-18.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
ϵ	Training	Test	Training	Test	Training	Test	Training	Test
0.8	65.78%	60.64%	86.25%	81.07%	49.51%	47.76%	41.92%	40.76%
1.6					59.81%	57.63%	48.40%	46.35%
3.2					70.41%	68.34%	52.28%	50.70%
6.4					77.32%	75.05%	56.99%	55.45%
12.8					82.77%	80.43%	61.45%	58.75%

Table 6. (Fine-tuning) CIFAR-10 image classification results for various neural network models. The target ϵ value is 6.4.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
Model	Training	Test	Training	Test	Training	Test	Training	Test
ResNet18	65.78%	60.64%	86.25%	81.07%	77.32%	75.05%	56.99%	55.45%
ResNet34	66.79%	62.32%	93.17%	94.82%	74.25%	71.74%	53.23%	49.26%
ResNet50	65.21%	62.34%	84.73%	78.75%	69.85%	64.95%	45.84%	44.11%
ViT-B-16	47.27%	49.21%	94.34%	96.48%	90.51%	94.27%	83.86%	87.31%
ViT-B-32	49.38%	49.31%	93.17%	94.82%	90.05%	93.44%	84.10%	87.92%

Fine-tuning experiments In addition to the training from scratch settings, we additionally have experiments in fine-tuning settings. The hyper-parameters except the total training epochs are almost the same with the training from scratch setting. (for detailed experimental setup, please follow our source code's setup file.) In this experiment, we consider two additional baseline methods without DP (signSGD and P-signSGD (Chen et al., 2020)). In Tables 5 and 6, we compare the training/test classification performance of the proposed method and baselines. As shown in the tables, the proposed method consistently outperforms the DP-signSGD for all ϵ values and neural network models.

More interestingly, the proposed sometimes outperforms the signSGD, even though it does not add any noise to the gradient. The reason is that the convergence speed of signSGD can be enhanced for large neural network models, where this is theoretically shown in the non-iid distributed learning settings (Chen et al., 2020). To verify this, we additionally compare the P-signSGD method with the standard signSGD method, which shows that the adding small amount of noise before sign function can enhance the convergence of the signSGD.[6]

[6] We note that the value of additive noise in P-signSGD is not optimized, because this is not target of our work. We just add this method to show that adding small noise can enhance the classification accuracy.

F. Numerical Results: CelebA Attribute Classification

For general results on the practical environments, we implemented more performance benchmarks on CelebA (Liu et al., 2015) attribute classification. Among 162,770 training samples, we use only first 60,000 images in our experiment, where the number of test images is 19,962 and the number of binary classes is 40. For the data augmentation, we use center cropping, resize to 128x128 pixels, random rotation, and random horizontal flip. Furthermore, we fine-tune the models from the pre-trained models available in Pytorch repository (ImageNet-v1). The number of epochs is fixed to four and the batch size is 120.

In Tables 7 and 8, the proposed method consistently outperforms the baselines, while even closely achieves or outperforms the accuracy of standard signSGD.

Table 7. CelebA attribute classification results for various ϵ values. The model used in this experiment is ResNet-18.

DP	signSGD		Proposed		DP-signSGD	
	X		O		O	
ϵ	Training	Test	Training	Test	Training	Test
0.4	88.41%	88.30%	86.01%	85.82%	84.33%	84.14%
0.8			87.37%	87.10%	84.85%	84.63%
1.6			88.58%	88.27%	86.38%	86.06%
3.2			89.49%	89.11%	86.73%	86.47%
6.4			89.93%	89.53%	87.14%	86.90%
12.8			90.19%	89.83%	87.41%	87.19%
25.6			90.24%	89.81%	87.73%	87.48%

Table 8. CelebA attribute classification results for various neural network models. The target ϵ value is 6.4.

DP	signSGD		Proposed		DP-signSGD	
	X		O		O	
Model	Training	Test	Training	Test	Training	Test
ResNet18	88.73%	88.42%	89.93%	89.53%	87.14%	86.90%
ResNet34	88.22%	88.03%	89.71%	89.35%	87.00%	86.75%
ResNet50	88.38%	88.28%	89.09%	88.84%	86.16%	86.00%
ViT-B-16	87.84%	87.60%	91.52%	91.00%	88.50%	88.30%
ViT-B-32	88.41%	88.30%	91.17%	90.71%	88.39%	88.11%

G. Numerical Results: Brain Tumor MRI Dataset

In this section, we provide numerical results with a realistic vision dataset, brain tumor MRI dataset[7]. The dataset contains 5,712 training images and 1,331 test images, where the number of classes is four[8]. In this experiment, we have the following data augmentations:

- Random resized crop (scale=0.95∼1.0).
- Random rotation (-5 to 5 degrees).
- Random horizontal flip.

Also, we have trained 20 epochs for ResNets and 8 epochs for ViT models, where the batch size is fixed to 50. In Tables 9 and 10, the proposed method consistently outperforms the baselines, while even closely achieves or outperforms the accuracy of standard signSGD, where the accuracy of P-signSGDshows that adding negligibly small noise to the gradient excessively enhance the convergence speed.

Table 9. Brain tumor MRI classification results for various ϵ values. The model used in this experiment is ViT-B-32.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
ϵ	Training	Test	Training	Test	Training	Test	Training	Test
0.4	78.75%	73.53%	99.81	99.01	77.07%	72.08%	72.01%	67.35%
0.8					83.33%	77.42%	81.18%	74.75%
1.6					86.38%	78.64%	83.79%	78.26%
3.2					90.20%	83.91%	86.03%	79.41%
6.4					92.52%	88.56%	87.01%	79.94%
12.8					94.56%	91.46%	88.74%	81.39%
25.6					97.11%	95.19%	88.97%	82.00%

Table 10. Brain tumor MRI classification results for various neural network models. The target ϵ value is 6.4.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
Model	Training	Test	Training	Test	Training	Test	Training	Test
ResNet18	86.92%	85.28%	99.81%	99.47%	90.11%	83.83%	84.35%	77.88%
ResNet34	89.76%	88.63%	99.75%	99.16%	89.67%	84.06%	85.26%	77.19%
ResNet50	88.55%	85.89%	98.20%	97.18%	88.94%	82.53%	78.78%	71.40%
ViT-B-16	83.67%	81.31%	99.86%	99.54%	94.94%	92.75%	89.23%	83.60%
ViT-B-32	78.75%	73.53%	99.81%	99.01%	92.52%	88.56%	87.01%	79.94%

[7] https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

[8] ‘glioma’, ‘meningioma’, ‘notumor’, ‘pituitary’.

H. Numerical Results: Blood Cell Images

In this section, we provide numerical results with a realistic medical dataset, blood cell images dataset [9]. The numbers of training and test images are 9,957 and 2,487, respectively. We use the following data augmentations in our training:

- Resize to 224x224 pixels.
- Random rotation (-10 to 10 degrees).
- Random horizontal flip.

Also, the training scenario is fine-tuning from the pre-trained models (ImageNet-v1), where the number of epochs is 20 for ResNets and 8 for ViT models.

In Tables 11 and 12, the proposed method consistently outperforms the baselines, while even closely achieves or outperforms the accuracy of standard signSGD, where the accuracy of P-signSGDshows that adding negligibly small noise to the gradient excessively enhance the convergence speed.

Table 11. Blood cell images classification results for various ϵ values. The model used in this experiment is ViT-B-32.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
ϵ	training	test	training	test	training	test	training	test
0.4	83.16	77.12	95.45	87.09	62.13%	56.25%	55.53%	53.52%
0.8					66.80%	66.47%	63.36%	64.46%
1.6					79.04%	81.95%	67.42%	69.36%
3.2					82.73%	81.50%	70.95%	70.21%
6.4					88.00%	82.71%	74.21%	75.95%
12.8					91.38%	86.49%	79.06%	81.42%
25.6					92.91%	87.62%	80.13%	77.20%

Table 12. Blood cell images classification results for various neural network models. The target ϵ value is 6.4.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
Model	training	test	training	test	training	test	training	test
ResNet18	99.04%	90.71%	100.00%	87.98%	91.65%	85.69%	72.63%	79.01%
ResNet34	97.59%	87.98%	99.94%	90.31%	95.81%	87.70%	77.77%	73.70%
ResNet50	95.03%	76.72%	99.99%	89.47%	93.08%	85.52%	40.73%	39.53%
ViT-B-16	82.43%	81.58%	93.62%	89.51%	89.93%	86.29%	81.14%	85.61%
ViT-B-32	83.16%	77.12%	95.45%	87.09%	88.00%	82.71%	74.21%	75.95%

[9] https://www.kaggle.com/datasets/paultimothymooney/blood-cells/data

I. Numerical Results: Chest X-Ray Images (Pneumonia)

In this section, we present additional experiments with a medical image dataset, chest x-ray image dataset[10]. The number of training images is 5,126. The number of test images is 624. We have used the following data augmentation blocks in our training procedure:

- Random resized crop (scale is 0.7 to 1.0).

We fine-tune the models from the pre-trained models available in Pytorch repository (ImageNet-v1), where the number of epochs is 10 for both ResNets and ViTs.

In Tables 13 and 14, the proposed method consistently outperforms the baselines, while even closely achieves or outperforms the accuracy of standard signSGD, where the accuracy of P-signSGDshows that adding negligibly small noise to the gradient excessively enhance the convergence speed.

Table 13. Chest x-ray images classification results for various ϵ values. The model used in this experiment is ResNet-34.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
ϵ	training	test	training	test	training	test	training	test
0.4	96.26%	90.87%	98.73	94.09	91.32%	84.94%	78.62%	77.08%
0.8					93.83%	88.14%	91.10%	84.62%
1.6					93.81%	87.98%	93.65%	86.22%
3.2					94.59%	87.34%	92.85%	86.38%
6.4					95.49%	88.78%	94.36%	88.62%
12.8					96.09%	90.22%	94.21%	88.78%
25.6					96.70%	89.74%	94.44%	89.10%

Table 14. Chest x-ray images classification results for various neural network models. The target ϵ value is 6.4.

DP	signSGD X		P-signSGD X		Proposed O		DP-signSGD O	
Model	training	test	training	test	training	test	training	test
ResNet34	96.26%	90.87%	98.73%	94.07%	95.49%	88.78%	94.36%	88.62%
ResNet50	93.81%	89.42%	98.73%	94.07%	95.48%	90.38%	93.85%	88.30%
ViT-B-32	90.51%	86.54%	98.41%	94.87%	95.38%	92.15%	92.81%	88.78%

[10] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

J. Extension to Majority Vote Scenario

In Algorithms 1 and 2, the distributed optimization settings are not considered, i.e., the gradients need to be aggregated in full precision before adding noise and taking the sign. Hence, in distributed optimization settings, this would mean that the clients need to send the gradients in full precision to the server. In the majority vote paper (Bernstein et al., 2019), their method only need to send signs to the servers; hence, achieving further communication code reduction.

In this section, we present the method for extending our method to the majority vote scenario. We begin with the following gradient exchange process:

- **Step 1 (at k-th agent):** Local agents compute gradient and transmit the signs of their gradients (with noise perturbation).
  1. Compute Gradient: $\tilde{\mathbf{g}}_t{}^{(k)}(z_i) \leftarrow \nabla\ell(\theta_t, x_i^{(k)})$
  2. Clipping Gradient: $\tilde{\mathbf{g}}_t^{(k)}(z_i) \leftarrow \tilde{\mathbf{g}}_t^{(k)}(z_i)/\max(1, \frac{\|\tilde{\mathbf{g}}_t^{(k)}\|_2}{C})$.
  3. Mini-batch gradient for all $i \in \mathcal{B}$: $\tilde{\mathbf{g}}_t^{(k)} \leftarrow \sum_i \tilde{\mathbf{g}}_t^{(k)}$.
  4. Compute Sign of the Gradient: $\tilde{\mathbf{g}}_t^{(k)} \leftarrow \texttt{sign}(\tilde{\mathbf{g}}_t^{(k)} + \text{Logistic}(0, sC\mathbf{I}))$.

- **Step 2:** The server send back the aggregated sign gradient.
  1. Aggregate ($\tilde{\mathbf{g}}_t^{(k)}$: gradient received from agent k.) $\tilde{\mathbf{g}}_t \leftarrow \texttt{sign}(\sum_k \tilde{\mathbf{g}}_t^{(k)})$.

- **Step 3:** Local agent updates their model via aggregated gradient $\tilde{\mathbf{g}}_t$.

In the above update process, we have the following equation:

$$\alpha_{\mathcal{M}}(\lambda) = \max_{\mathbf{g},\mathbf{g}',\text{aux}} \alpha_{\mathcal{M}}(\lambda, \text{aux}, \mathbf{g}, \mathbf{g}'),$$

where we omit variable aux in our paper for brevity. (See Eq. (16) in our paper). In our analysis, we explore the differential privacy implications for data within client k in a majority voting framework. Here, the sign gradients from other clients are regarded as auxiliary information (aux). In this context, the value of $\alpha_{\mathcal{M}}(\lambda, \text{aux}, \mathbf{g}, \mathbf{g}')$ becomes non-zero iff client k acts as a decision maker—able to influence the sign of the aggregated gradient. To apply the maximum operator from the aforementioned equation, we configure aux such that client k assumes control over all parameters, i.e., client k is always a decision maker, aligning the equation with Eq. (19) in our paper.

In the following subsections, we present the numerical results for the **majority vote scenario**, where the number of nodes is three. For brevity, we skip the detailed analysis for Tables 15 and 16, because the results jointly show that the proposed method outperforms the DP-SIGNSGD.

Table 15. (From scratch) MNIST image classification results for varying ϵ, under majority voting scenario. The model used in this experiment is custom dense network.

	PROPOSED		DP-SIGNSGD	
ϵ	TRAINING	TEST	TRAINING	TEST
0.2	87.56	88.29	86.90	87.44
0.4	89.30	89.98	88.75	89.49
0.8	90.70	90.96	89.67	90.47
1.6	92.23	92.48	90.40	91.07
3.2	94.25	94.23	90.87	91.21
6.4	96.17	96.00	91.36	91.82
12.8	97.43	96.94	91.96	92.67
25.6	98.27	97.44	92.42	92.88

Table 16. (From scratch) MNIST image classification results for various neural network models, under majority voting setting. The value of ϵ is fixed to 6.4.

	Proposed		DP-signSGD	
Model	training	test	training	test
Dense	96.17	96.00	91.36	91.82
ResNet10	98.74	98.84	95.95	96.22
ResNet18	99.10	99.06	96.69	96.89
ResNet34	99.04	98.93	96.61	96.83
ResNet50	98.58	98.16	94.38	94.56
VGG11	99.17	98.94	97.44	97.66
VGG13	99.27	99.10	97.78	97.83
VGG16	99.19	99.04	97.75	97.84
VGG19	99.11	98.93	97.30	97.84
ViT-B-16	99.11	98.93	96.34	96.32