Improving Differentially Private SGD via Randomly Sparsified Gradients

Anonymous authors
Paper under double-blind review

Abstract

Differentially private stochastic gradient descent (DP-SGD) has been widely adopted in deep learning to provide rigorously defined privacy, which requires gradient clipping to bound the maximum norm of individual gradients and additive isotropic Gaussian noise. With analysis of the convergence rate of DP-SGD in a non-convex setting, we identify that randomly sparsifying gradients before clipping and noisification adjusts a trade-off between internal components of the convergence bound and leads to a smaller upper bound when the noise is dominant. Additionally, our theoretical analysis and empirical evaluations show that the trade-off is not trivial but possibly a unique property of DP-SGD, as either canceling noisification or gradient clipping eliminates the trade-off in the bound. Based on the observation, we propose an efficient and lightweight extension using random sparsification (RS) to strengthen DP-SGD. Applying RS across various DP-SGD frameworks improves performance, while the produced sparse gradients of RS exhibit advantages in reducing communication cost and strengthening privacy against reconstruction attacks, which are also key problems in private machine learning.

1 Introduction

Internet-scale data promises to accelerate the development of data-driven statistical approaches, but the need for privacy constrains the amalgamation of such datasets. As a result, private data are in fact isolated, constraining our ability to build models that learn from a large number of instances. On the other hand, the information contained in locally stored data can also be exposed through releasing the model trained on a local dataset (Fredrikson et al., 2015; Shokri et al., 2017), or even reconstructed when gradients generated during training are shared (Zhu et al., 2019; Zhu & Blaschko, 2021).

To address these issues, many applications of machine learning are expected to be privacy-preserving, while differential privacy (DP) provides a rigorously defined and measurable privacy guarantee. As described in Definition 1, DP defines privacy with respect to the difficulty of distinguishing the outputs of different data:

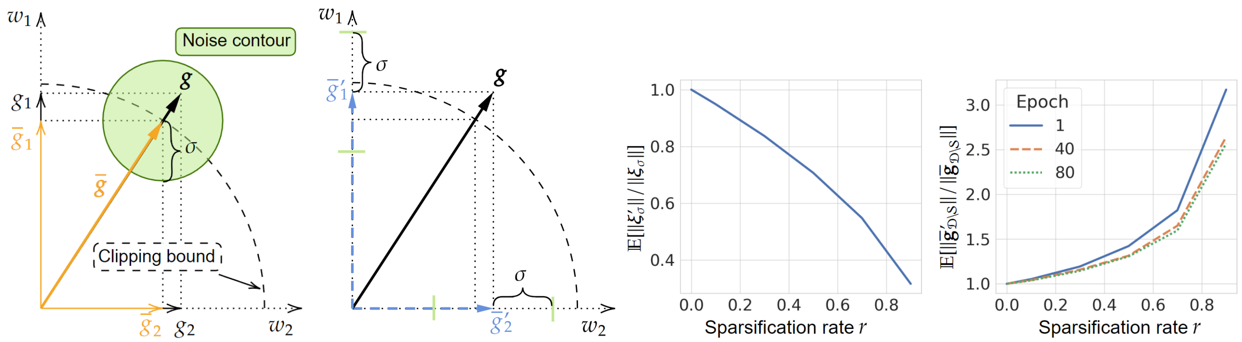
Definition 1 ((ε, δ)-DP (Dwork & Roth, 2014))**.** For a pair of neighboring datasets $X, X' \in \mathcal{X},\ X$ is obtained from X' by adding or removing an element. A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ is (ε, δ)-differentially private, if for any subset of outputs $S \subseteq \mathcal{R}$ it holds that:

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

A common paradigm for applying DP in deep learning is differentially private stochastic gradient descent (DP-SGD) proposed by Abadi et al. (2016), which lets the randomized mechanism $\mathcal{M}$ output perturbed gradients:

$$f(X) := \sum_{x \in X} g(x), \tag{2}$$

$$\mathcal{M}(X) := f(X) + \mathcal{N}(0, S_f^2 \sigma^2 \boldsymbol{I}_d), \tag{3}$$



(a) DP-SGD (b) DP-SGD with RS (c) noise norm ratio (d) gradient norm ratio

Figure 1: (a) Schematic of DP-SGD. We denote $\boldsymbol{g}$ as the gradient vector and omit the notation for input data, g_1 and g_2 are two entries in $\boldsymbol{g}$ corresponding to the coordinates w_1 and w_2, $\bar{\boldsymbol{g}}$ is the clipped gradient vector. (b) Schematic of DP-SGD with RS, $\bar{g}'_1$ and $\bar{g}'_2$ are two possible results under RS. (c) Empirical ratio of noise norm with RS and without RS $\|\boldsymbol{\xi}'_\sigma\|/\|\boldsymbol{\xi}_\sigma\|$, we take $d = 5.5 \times 10^5$ which is the dimension of DP-CNN. (d) Empirical ratio of the gradient norm of relaxed coordinates with RS and without RS $\|\bar{\boldsymbol{g}}'_{\mathcal{D}\setminus\mathcal{S}}\|/\|\bar{\boldsymbol{g}}_{\mathcal{D}\setminus\mathcal{S}}\|$, measured on DP-CNN with dataset CIFAR10. We remark that for (a) and (b) clipping bound C within range $(\max(\{|g_i|\}_{i=1}^{\mathcal{D}}),\ \|\boldsymbol{g}\|)$ is representative, as in high-dimensional space $\|\boldsymbol{g}\| \gg \max(\{|g_i|\}_{i=1}^{\mathcal{D}})$, while the two strengths still exist for C being smaller.

where X now stands for a batch of data and $g(x) \to \mathbb{R}^d$ computes gradient given an individual example in the batch. The isotropic Gaussian distributed noise $\mathcal{N}(0, S_f^2\sigma^2\boldsymbol{I})$ is calibrated to f's sensitivity $S_f^2 := \max_{X,X'} \|f(X) - f(X')\|$, while the noise multiplier σ controls the strength of the privacy guarantee. As S_f is usually unknown, DP-SGD commonly clips the gradient of each individual example in Euclidean norm to a preset bound C, such that $\bar{f}(X) := \sum_{x\in X} g(x) \cdot \min(1, C/\|g(x)\|)$. Then S_f can be safely replaced by C and the randomized mechanism $\mathcal{M}$ can thus be expressed as:

$$\mathcal{M}(X) := \bar{f}(X) + \mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_d). \tag{4}$$

However, gradient clipping gives rise to a squeezed gradient distribution, while noisification further blurs the gradient distribution, both cause adverse effects for optimization. Understanding the influence of these two operations becomes crucial for the progress of DP-SGD.

1.1 Our contribution

In this work we present a special property of DP-SGD that is established jointly by gradient clipping and noisification. In particular, we analyze DP-SGD in a non-convex and smooth setting and identify that randomly sparsifying gradients before clipping and noisification can trigger a trade-off between internal components of the convergence bound and possibly result in a smaller upper bound. This observation suggests that in the context of DP-SGD randomly sparsified gradients could lead to faster convergence than the full gradient. This also implies that DP-SGD has inherent potential for straightforward random gradient compression. We further note that the same property is not present in other popular SGD schemes.

First, we describe the process of random sparsification (RS) and illustrate its impact on DP-SGD. RS uniformly selects a random subset $\mathcal{S}$ from the full coordinates set $\mathcal{D}$ and zeros out their gradients, we define sparsification rate $r := |\mathcal{S}|/|\mathcal{D}|$ as the percentage of coordinates we would like to sparsify. Figure 1a, 1b schematically illustrate the vanilla DP-SGD and DP-SGD with RS using a 2D example. In this 2D example RS can either have g_1 or g_2 zeroed-out so r can only be set to 0.5. In higher-dimensional cases it will be possible to set r to other values within the range $(0, 1)$. Compared with vanilla DP-SGD, the drawback of RS is indeed apparent as RS removes some gradient information. However, on the other side RS has two strengths: (i) due to the isotropic Gaussian noise $\boldsymbol{\xi}_\sigma$, the amount of noise in term of Euclidean norm scales with the gradient dimension $d = |\mathcal{D}|$. With RS the effective d is reduced as the sparsified coordinates do not receive any gradient information, and noisification in those dimensions is no longer necessary. Figure 1c demonstrates the empirical ratio of the noise norm with RS over without RS $\|\boldsymbol{\xi}'_\sigma\|/\|\boldsymbol{\xi}_\sigma\|$. (ii) Due to gradient clipping, for the coordinates $\mathcal{D} \setminus \mathcal{S}$ that are relaxed, i.e. corresponding gradients are not zeroed out, their gradients have a higher magnitude than without RS (see $|\bar{g}'_1| > |\bar{g}_1|$ or $|\bar{g}'_2| > |\bar{g}_2|$), implying the relaxed coordinates are better optimized, Figure 1d demonstrates an empirical ratio of the gradient norm of relaxed coordinates with RS over without RS $\|\bar{\boldsymbol{g}}'_{\mathcal{D}\setminus\mathcal{S}}\|/\|\bar{\boldsymbol{g}}_{\mathcal{D}\setminus\mathcal{S}}\|$. In Section 3, we will discuss the integrated impact of the drawback and strengthens of DP-SGD with RS for non-convex and smooth problems and as we will see RS gives rise to a trade-off between internal components of the convergence bound of DP-SGD. As RS modifies DP-SGD, we clarify that RS does not breach privacy with the following theorem:

Theorem 1.1. *For a Gaussian mechanism: $\mathcal{M}(x) := \sum_{x\in X} g(x) \cdot \min(1, C/\|g(x)\|) + \mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_d)$, which satisfies (ε, δ)-DP, after applying RS with a mask $m \in \{0,1\}^d$, the modified Gaussian mechanism $\mathcal{M}'(x) := \sum_{x\in X} m \odot g(x) \cdot \min(1, C/\|m \odot g(x)\|) + m \odot \mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_d)$ also satisfies (ε, δ)-DP.*

The Hadamard product is denoted using the symbol $\odot$.

Proof. Note that $\mathcal{M}'$ is equivalent to $\mathcal{M}''(x) := \sum_{x\in X} g_{\mathcal{D}\setminus\mathcal{S}}(x) \cdot \min(1, C/\|g_{\mathcal{D}\setminus\mathcal{S}}(x)\|) + \mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_{|\mathcal{D}\setminus\mathcal{S}|})$ in terms of privacy, then it is easy to observe that $\mathcal{M}''(x)$ and $\mathcal{M}(x)$ provide the same level of DP. It should be noted that the scenario of $m = \mathbf{0}$ is not considered in our analysis as it does not align with the objectives of optimization. □

Another interesting observation we will present is that for other popular SGD schemes, i.e. SGD with gradient clipping (same as canceling noisification in DP-SGD), noisy SGD (same as canceling clipping in DP-SGD), and vanilla SGD, the respective convergence bounds are only enlarged by applying RS. This observation highlights the fact that the reason for RS improving DP-SGD is not trivially due to the reduced noise, and suggests that *the trade-off raised by RS in the context of DP-SGD is special, which manifests under two preconditions: gradient clipping and noisification.*

An alternative to random sparsification would be e.g. to sparsify the smallest entries of the gradient, w_2 in Figure 1. It is worth emphasizing that the selection of significant coordinates is also privacy related. It could be done by using public dataset (Zhou et al., 2021; Kairouz et al., 2021) which we do not assume exists, or by providing additional privacy budget for sparse vector techniques (Dwork & Roth, 2014) or DP selection (Zhang et al., 2021) which have certain technical difficulties of defining the threshold and privacy loss for the selection, and are thus only applied when the gradient is obviously sparse, e.g. the embedding layer of a language model. Perhaps most importantly, the analysis of these methods assumes the original sparsity of the gradients (e.g. $\boldsymbol{g}$ lies in w_1 in Figure 1) or low-rank property (e.g. $\boldsymbol{g}$ always lies in a subspace). In contrast our analysis, and the strengths of RS as illustrated in Figure 1, do not rely on these assumptions, which makes the contribution of this work complementary to previous works.

Paper organization: In Section 2, we discuss the related works. In Section 3, we elaborate on the impact of RS on popular SGD schemes and finally present the special trade-off of DP-SGD induced by RS. In Section 4, we provide an efficient and lightweight RS algorithm based on our analysis. In Section 5, we discuss the additional advantages of sparsified gradients. In Section 6 we empirically verify our analysis and show the utility of our proposed algorithm. All proofs are deferred to the Appendix.

2 Related works

Chen et al. (2020) analyze the convergence rate of DP-SGD in a non-convex setting with characterization of the adverse effect of the gradient clipping. Many works study adaptive clipping bounds (Andrew et al., 2021; Pichapati et al., 2019). Other works study the adverse effect of noisification and prove that the performance of DP-SGD is dependent on the gradient dimension d as, according to Equation 4, the amount of noise scales with d. Bassily et al. (2014) show that in a convex setting, DP-SGD achieves excess risk of $\tilde{O}(\sqrt{d}/n\varepsilon)$, where n is the dataset size. Wang & Xu (2019) show that the empirical gradient norm decreases to $\tilde{O}(d^{1/4}/\sqrt{n\varepsilon})$ for DP-SGD with a smooth non-convex loss function.

A line of work builds on gradient space compression to improve the utility of DP-SGD. Abadi et al. (2016) propose DP-Transfer Learning to pre-train a network on an auxiliary dataset, then transfer the feature extractor and only re-train the linear classifier on the private data. Similarly, Tramer & Boneh (2021) adopt ScatterNet (Oyallon et al., 2019) to extract handcrafted features. Both works decrease d by excluding the majority of parameters during DP learning. Inspired by the empirical observation that the optimization trajectory is contained in a lower-dimensional subspace (Vogels et al., 2019; Gooneratne et al., 2020; Li et al., 2020), several recent works (Zhou et al., 2021; Yu et al., 2021a; Kairouz et al., 2021; Yu et al., 2021b) project the gradient into a subspace which is identified by auxiliary data or released historical gradients. Zhang et al. (2021) target NLP tasks where gradients are extremely sparse, and propose a DP selection method.

In contrast to the strategy of reducing gradient space dimension, Li et al. (2021) observe that DP-SGD works well on full fine-tuning of large language models. Papernot et al. (2021) propose a dedicated DP-CNN with tempered activation function which is deemed as robust to the adverse effects of gradient clipping and noisification in DP-SGD.

The studies of Damaskinos et al. (2021) and Mangold et al. (2022) on DP coordinate descent also involve RS as the gradient of a random coordinate is computed in backpropagation. However, Damaskinos et al. (2021) limit their study to the generalized linear model and they investigate the dual model vector instead of directly analyzing the model parameters. Mangold et al. (2022) rely on precise coordinate-wise regularity measures of the objective function (component-smoothness constants), which we do not need. To the best of our knowledge, our work is the first to show the value of straightforwardly applying RS to DP-SGD in a general non-convex setting.

3 Applying random sparsification to gradient descent methods

Notation & Terminology Let $\mathcal{L}$ be the objective function $\mathcal{L}(w) := \mathbb{E}_{x\in\mathcal{X}}[\ell(w;x)]$ which is G-Lipschitz smooth, and x denotes a training example sampled from the dataset $\mathcal{X}$. We define $g_{t,i} := \nabla\ell(w_t;x_i)$ the gradient at step t with example x_i and define the true gradient $\nabla w_t := \mathbb{E}_{x\in\mathcal{X}}[g_{t,i}]$. We assume the gradient deviation $g_{t,i} - \nabla w_t$ is sampled from a zero-mean random variable ξ_t. The averaged gradient of B samples at the step t is denoted as $g_t := \frac{1}{B}\sum_i g_{t,i}$, and the averaged clipped gradient as $\bar{g}_t := \frac{1}{B}\sum_i g_{t,i} \cdot \min(1, C/\|g_{t,i}\|)$. To conduct RS, we use a random mask $m \in \{0,1\}^d$ and uniformly draw rd indices and set these positions in the mask to 0 while others to 1 so that $\|m\|_1 = (1-r)d$, r is the predefined sparsification rate. The average of sparsified and clipped gradients can thus be expressed as $\hat{g}_t := \frac{1}{B}\sum_i g'_{t,i} \cdot \min(1, C/\|g'_{t,i}\|)$, where we define $g'_{t,i} := m \odot g_{t.i}$. For noisification, isotropic Gaussian noise is denoted as ξ_σ, when RS is applied we do $m \odot \xi_\sigma$. As this work involves multiple ways of processing gradients, we summarize their notation in Table 1 for convenience. We do not bold vectors or matrices in the analysis since it is clear from the context.

3.1 Applying random sparsification to vanilla SGD and noisy SGD

We first note that for vanilla SGD (Bottou et al., 2018), applying RS is the same as randomly dropping gradient information, thus RS cannot provide any benefits. Next, we consider the impact of RS under noisification and gradient clipping separately. If we cancel the gradient clipping in DP-SGD, the resulting optimization method is noisy SGD which has been used in Bayesian learning, e.g. stochastic gradient Langevin dynamics (Welling & Teh, 2011). When RS is applied (pseudo code is given in Appendix E), the amount of noise will be reduced as illustrated in Figure 1, however such a benefit seems not sufficient to compensate the loss of gradient information, which we show in Theorem 3.1.

Theorem 3.1. *Consider noisy SGD with RS on G-smooth function $\mathcal{L}$ with isotropic Gaussian noise $\xi_\sigma \sim \mathcal{N}(0, \sigma^2\boldsymbol{I}_d)$, learning rate $\gamma = 1/G$, batch size B and sparsification rate r, assume an upper bound of individual gradient deviation $\|\xi_t\|^2 \le \sigma_g^2$, we have:*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla w_t\|^2 \le \frac{2G}{(1-r)T}\Delta_{\mathcal{L}} + \frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}, \tag{5}$$

Gradient	
$g_{t,i}$	Gradient at step t with example x_i, i.e. $\nabla \ell(w_t, x_i)$.
∇w	True gradient at step t, i.e. $\mathbb{E}_{x\in\mathcal{X}}[g_{t,i}]$
g_t	Mini-batch gradient at step t, i.e. $\frac{1}{B}\sum_i g_{t,i}$
$\bar{g}_t$	Clipped mini-batch gradient at step t, i.e. $\frac{1}{B}\sum_i g_{t,i} \cdot \min(1, C/\lVert g_{t,i}\rVert)$
$g'_{t,i}$	Sparsified gradient due to random mask m, i.e. $m \odot g_{t,i}$.
$\nabla w'$	Sparsified true gradient due to random mask m, i.e. $m \odot \mathbb{E}_{x\in\mathcal{X}}[g_{t,i}]$
$\hat{g}_t$	Sparsified and clipped mini-batch gradient at step t, i.e. $\frac{1}{B}\sum_i g'_{t,i} \cdot \min(1, C/\lVert g'_{t,i}\rVert)$.
Gradient deivation	
ξ_t	Assumed random variable generating gradient deviations, i.e. $g_{t,i} - \nabla w_t$.
ξ'_t	Assumed random variable generating sparsified gradient deviations, i.e. $g'_{t,i} - \nabla w'_t$.
p_t	True probability distribution of ξ_t.
$\tilde{p}_t$	A symmetric proxy of p_t which approximates p_t while satisfies $\tilde{p}_t(\xi_t) = \tilde{p}_t(-\xi_t)$.
p'_t	True probability distribution of ξ'_t.
$\tilde{p}'_t$	A symmetric proxy of p'_t which is projected from p_t by random mask m.

Table 1: Notations for gradients.

where we define $\Delta_{\mathcal{L}} := \mathbb{E}[\mathcal{L}(w_1)] - \min_w \mathcal{L}(w)$. We note that the noise term $d\sigma^2/B^2$ on the r.h.s. was reduced by a factor of $1-r$ due to the RS, but this factor is canceled out by the same factor from the l.h.s. (see Appendix A). Additionally, Equation (5) can also be used to describe the case of vanilla SGD with RS by removing the last term $d\sigma^2/B^2$ at the r.h.s. Overall, we have the following observation:

Remark 1. Theorem 3.1 suggests that in applying RS to noisy SGD or vanilla SGD, the network may converge more slowly, as $\|\nabla w_t\|^2$ has a larger upper bound for $0 < r < 1$.

3.2 Applying random sparsification to SGD with gradient clipping

If we cancel the noisification in DP-SGD, then the optimization method becomes SGD with gradient clipping, which has been widely adopted for large network training to prevent exploding gradients or stabilize optimization (Zhang et al., 2020; Pascanu et al., 2013). If RS is applied (pseudo code is given in Appendix E), relaxed coordinators can be better optimized as illustrated in Figure 1, however such benefit also seems not sufficient to compensate the loss of gradient information.

First consider the convergence of gradient clipping without sparsification.

Lemma 3.1. *Consider SGD on G-smooth function $\mathcal{L}$ with gradient clipping of bound C, learning rate γ, we have:*

$$\mathbb{E}[\langle \nabla w_t, \bar{g}_t \rangle] \leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \gamma\frac{GC^2}{2}. \tag{6}$$

As the distribution of the gradient is unknown, the expectation involving gradient clipping, i.e. $\mathbb{E}[\langle \nabla w_t, \bar{g}_t \rangle]$, impedes the estimation of the convergence rate. Let p_t be the true distribution of the gradient deviation ξ_t. Chen et al. (2020) firstly observe that p_t appears to be symmetric and use this property to relate $\mathbb{E}[\langle \nabla w_t, \bar{g}_t \rangle]$ to $\|\nabla w_t\|$:

Lemma 3.2 (Chen et al. (2020))**.** *Assume $p_t(\xi_t) = p_t(-\xi_t)$, $\forall \xi_t \in \mathbb{R}^d$, gradient clipping with bound C has the following property:*

$$\mathbb{E}[\langle \nabla w_t, \bar{g}_t \rangle] \geq P_{\xi_t \sim p_t}(\|\xi_t\| < 3C/4) h(\nabla w) \|\nabla w_t\|, \tag{7}$$

where $h(\nabla w) := \min(\|\nabla w_t\|, C/4)$.

We see the r.h.s. of Equation 7 is proportional to $\|\nabla w\|$ or $\|\nabla w\|^2$ as long as $P_{\xi_t \sim p_t}(\|\xi_t\| < 3C/4)$ is not close to zero. Combining Lemma 3.1 with Lemma 3.2, it is possible to form an upper bound for finding the critical point. However, the real gradient distribution cannot be exactly symmetric. Instead of using p_t in Equation (7), we can choose a proxy $\tilde{p}$ which is symmetric and use an error term b_t to represent the difference as $b_t := \mathbb{E}_{\xi_t \sim p_t}[\langle \nabla w, \bar{g}_t \rangle] - \mathbb{E}_{\xi_t \sim \tilde{p}_t}[\langle \nabla w, \bar{g}_t \rangle]$.

Theorem 3.2 (Chen et al. (2020)). *Following Lemma 3.1 and Lemma 3.2, choose a symmetric distribution $\tilde{p}(\cdot)$ satisfying $\tilde{p}_t(\xi_t) = \tilde{p}_t(-\xi_t)$, $\forall \xi_t \in \mathbb{R}^d$, consider T iterations:*

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t)\|\nabla w_t\| \leq \frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma \frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T} b_t. \tag{8}$$

Chen et al. (2020) argues that there exists $\tilde{p}_t$ which is a close approximation of p_t such that b_t is small while $P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)$ is bounded away from zero, as in practice p_t tends to be approximately symmetric. They also empirically demonstrate that p_t becomes approximately symmetric during training. So Theorem 3.2 indicates the convergence rate of SGD with gradient clipping. To make sure this property of p_t is generally valid in our case, we have also verified that p_t becomes approximately symmetric in our experimental environments in Section 6 (see Appendix F).

Now consider the convergence rate after applying RS, note that the sparsification happens before clipping, so the conclusion cannot be straightforwardly derived from Theorem 3.2. Since m, ξ_t are independent, we have:

$$\begin{aligned}\mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] &= \mathbb{E}[\langle \nabla w'_t, \hat{g}_t \rangle] \\ &= \mathbb{E}_m\left[\mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w'_t, \hat{g}_t \rangle]\right] + \mathbb{E}_m[b'_t],\end{aligned} \tag{9}$$

where we define $\nabla w'_t := m \odot \nabla w_t$, $\xi'_t := m \odot \xi_t$ with corresponding true distribution p'_t and proxy $\tilde{p}'_t$ which are projected from p_t and $\tilde{p}_t$, $b'_t := \mathbb{E}_{\xi'_t \sim p'_t}[\langle \nabla w', \hat{g}_t \rangle] - \mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w', \hat{g}_t \rangle]$. Since the projection of a symmetric distribution to a subspace is symmetric, we have $\tilde{p}'_t(\xi'_t) = \tilde{p}'_t(-\xi'_t)$, so following Lemma 3.2, we have for the first term on the r.h.s. of Equation 9:

$$\mathbb{E}_m\left[\mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w'_t, \hat{g}_t \rangle]\right] \geq \mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) h(\nabla w'_t)\|\nabla w'_t\|]. \tag{10}$$

Then to solve the expectation over the random mask m in the Equation 10, we provide Lemma 3.3:

Lemma 3.3. *Apply RS with sparsification rate r, choose $\tilde{p}_t$ such that $P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \geq \sqrt{1-r}$, then $\exists\, \kappa_t \in (1-r, 1)$ such that:*

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) h(\nabla w'_t)\|\nabla w'_t\|] = \kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t)\|\nabla w_t\|, \tag{11}$$

κ_t takes the value 1 when $r = 0$. Based on Lemma 3.3, we provide Theorem 3.3 to characterize the convergence of SGD with gradient clipping and RS.

Theorem 3.3. *Combine Lemma 3.1, Lemma 3.3 and Equations 9, 10, consider T iterations, then $\exists\, \kappa \in (1-r, 1)$ such that:*

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa}\left(\frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma \frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_m[b'_t]\right), \tag{12}$$

Remark 2. Consider that b_t, b'_t tend to be small and negligible, compared with Theorem 3.2, Theorem 3.3 suggests that RS could impede the convergence of SGD with gradient clipping as $1/\kappa > 1$ for $r > 0$.

3.3 Applying random sparsification to DP-SGD

Although there is seemingly no advantage of using RS for noisy SGD or SGD with gradient clipping, respectively, we show that in case of DP-SGD, i.e. when noisification and gradient clipping are both presented, RS can induce a trade-off between internal components of the convergence bound and possibly achieve a smaller upper bound.

Lemma 3.4. *Consider DP-SGD on G-smooth function $\mathcal{L}$ with clipping bound C, isotropic Gaussian noise $\xi_\sigma \sim \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I}_d)$, learning rate γ, batch size B, sparsification rate r, we have:*

$$\mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] \leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \gamma \frac{GC^2}{2} + (1-r)\gamma \Delta_\sigma, \tag{13}$$

where we have defined $\Delta_\sigma := \frac{C^2\sigma^2 dG}{2B^2}$. Now we provide Theorem 3.4 to characterize the convergence rate of DP-SGD with RS.

Theorem 3.4. *Combine Lemma 3.3, Lemma 3.4, and Equations 9, 10, consider T iterations, then $\exists\, \kappa \in (1-r, 1)$, such that:*

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t\sim\tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \le \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma\frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b'_t]) + \frac{1-r}{\kappa}\gamma\Delta_\sigma. \tag{14}$$

It is worth noting that in Theorem 3.4 on the r.h.s. the noise term Δ_σ has a factor $1-r$ which is induced by sparsification, but unlike under noisy SGD (see Theorem 3.1), this factor is not fully canceled out as $\kappa > 1-r$. Also note that with no sparsification ($r = 0$), we have $\kappa = 1$, Theorem 3.4 has the following form:

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t\sim\tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \le \frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma\left(\frac{GC^2}{2} + \Delta_\sigma\right) - \frac{1}{T}\sum_{t=1}^{T} b_t, \tag{15}$$

which recovers the convergence bound of vanilla DP-SGD (Chen et al., 2020), implying that our convergence bound over RS is as tight as before. Comparing Equation 15 with Theorem 3.4, we observe:

Remark 3. RS introduces two factors on the r.h.s.: $1/\kappa, (1-r)/\kappa$. The noise term Δ_σ is reduced by $(1-r)/\kappa$ while the remaining terms are enlarged by by $1/\kappa$. The respective magnitude of Δ_σ and other terms determines whether it is a gain or loss to apply RS: a trade-off is established. It is worthwhile to note that Δ_σ scales with $C^2\sigma^2$ and the gradient dimension d.

We also present the upper bound using privacy budget variables (ε, δ) in Appendix C with Corollary C.1.1.

4 Implementation of random sparsification for DP-SGD

Theorem 3.4 reveals that RS induces a trade-off between the internal components of the convergence bound and has a chance to accelerate the convergence. However, the best sparsification rate is infeasible to be computed *a priori* and possibly varying during optimization. In this section, we present a practically efficient and lightweight RS approach. *The additional cost of running RS is negligible.*

Algorithm 1 outlines our approach. We apply RS to optimization with or without gradient momentum. In case of non-zero momentum, coordinates that have been selected can still be updated as long as their velocity has not decayed to zero. To make the algorithm efficient in practice, we adopt gradual cooling (Line 3) and per-epoch randomization (Line 5), which we discuss in the sequel.

4.1 Gradual cooling

In practice we find that if we initiate the training with a constant large sparsification rate r, the network converges slowly and performs poorly when the privacy budget has been fully consumed. According to Theorem 3.4 and Remark 3, we see sparsification is beneficial once Δ_σ is significant compared with the other terms in the convergence bound. At early training stages the network converges fast: $\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}]$ is large, while during training the optimization reaches a plateau: $\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}]$ decays. In contrast, Δ_σ is always fixed and therefore its relative significance grows progressively. To fit this dynamics, we use a simple gradual cooling strategy which linearly ramps up the sparsification rate from 0 to r^* during training, i.e. $r = r^* \cdot \frac{e}{E-1}$. We provide a more detailed discussion in Appendix G.

4.2 Per-epoch randomization

RS alternates the optimization direction. As SGD takes several steps to reach the local minimum, frequently alternating optimization direction might be harmful, which we have also observed in the experiments. Considering that tuning over the number of iterations for refreshing the random mask can be expensive, in this work we set the refresh to be per-epoch. Another strength of per-epoch randomization is that for one epoch

Algorithm 1 DP-SGD with Random Sparsification

Input: Initial parameters w_0; Epochs E; Batch size B; Sparsification rate: r^*; Clipping bound: C; Noise multiplier σ; Momentum: μ; Learning rate γ.

```
for $e = 0$ to $E - 1$ do
    ▷ Gradual cooling ◁
    $r(e) = r^* \cdot \frac{e}{E-1}$;
    ▷ Generate a random mask every epoch ◁
    $m \in \{0, 1\}^d$, s.t. $\|m\|_1 = d \cdot (1 - r(e))$;
    for $t = 0$ to $T - 1$ do
        ▷ For each $x_i$ in the minibatch of size $B$ ◁
        $g_{t,i} = \nabla \ell(w_t, x_i)$;
        ▷ Sparsify gradient ◁
        $g'_{t,i} = m \odot g_{t,i}$;
        ▷ Clip each individual gradient ◁
        $\hat{g}_t = \frac{1}{B} \sum_i g'_{t,i} \cdot \min(1, C/\|g'_{t,i}\|)$;
        ▷ Add sparsified noise ◁
        $\tilde{g}_t = \hat{g}_t + m \odot \mathcal{N}(0, \frac{C^2\sigma^2}{B^2} \boldsymbol{I}_d)$;
        ▷ Update parameters ◁
        $v_{t+1} = \mu \cdot v_t + \tilde{g}_t, w_{t+1} = w_t - \gamma v_{t+1}$;
```

there are certainly $(1-r)d$ coordinates updated, which is favorable in distributed learning as communication overhead is a key issue, and data are not transmitted every iteration. For per-iteration randomization, the cumulative number of updated parameters depends on the sparsification rate r and iterations between two communication rounds, resulting in higher communication overheads than per-epoch randomization.

5 Advantages of sparsified gradients

Sparse representation of the gradient generally offers additional advantages. Our analysis in Section 3 demonstrates that DP-SGD provides an inherent potential for trivial random gradient compression. In the sequel, we discuss the benefits of sparsified gradients in terms of private machine learning.

Reduced communication overhead of DP federated learning A line of work studies how to incorporate differential privacy in federated learning, which is a distributed learning scheme with the interest of protecting the privacy of participants (McMahan et al., 2017; Yang et al., 2019; Shokri & Shmatikov, 2015; Liu et al., 2020). While a major issue in federated learning is the communication bottleneck, sparse representation produced by RS can be transferred in form of non-zero values and indices. The cost of indexing is logarithmic in the number of parameters, consider that $log_2 10^9 < 32$ and for per-epoch randomization multiple rounds of communication could share the same indices, the cost of indexing is negligible, and communication overhead by RS is reduced to $\tilde{O}(1 - r)$.

Reduced cost of projected DP-SGD RS can also reduce the computational cost and memory footprint of Projected DP-SGD (Zhou et al., 2021; Yu et al., 2021a) which intend to add less noise by compressing the gradients, but it induces a significant computation cost by running the power method and projecting gradients into and out of the subspace. For the power method, the basic operation is $WW^\top V$, where $W \in \mathbb{R}^{d\times s}$ denotes sample gradients, $V \in \mathbb{R}^{d\times b}$ denotes eigenvectors, the computational cost is $O(dbs)$. Applying RS, a selection of rows of V, W can be deleted, as no information exits in that subspace, while b, s might also be able to be reduced. Overall, the computational cost is between $O(1 - r)$ and $O\left((1 - r)^3\right)$.

Another issue of projected DP-SGD is the memory footprint of V. Saving sparse V by RS can be achieved by storing non-zero values and indices of zeros or non-zero values. The cost of indexing is logarithmic in the number of parameters, consider that $log_2 10^9 < 32$, we can decrease the memory footprint by removing a single 32 bit float gradient. We note that RS uses the same mask for all eigenvectors. Therefore, the cost of indexing is negligible, i.e. the memory footprint of V after applying RS is reduced to $\tilde{O}(1 - r)$.

Strengthened privacy against gradient reconstruction attacks A gradient reconstruction attack is one of the most hazardous privacy attacks, which assumes that the adversary has access to victim's gradient and intends to recover the victim's training data through gradient matching (Zhu et al., 2019; Geiping et al., 2020; Zhu & Blaschko, 2021; Yin et al., 2021). This is a common scenario in distributed learning schemes, as participants need to share their local update when computing the global model. Zhu & Blaschko (2021) provide a closed-form solution of a gradient reconstruction attack: for a certain layer one can form a constraint matrix K over the input x given g, so that $Kx = g$, where x, g are flattened vectors (Zhu & Blaschko, 2021, Equation 15). When K is overdetermined, after receiving the gradient g, the input x can be reconstructed through least squares: $x = (K^\top K)^{-1} K^\top g$. To prevent such an attack, DP-SGD can be introduced so that g will be perturbed by Gaussian noise ξ_σ, which leads to a squared reconstruction error:

$$\|x - \hat{x}\|^2 = \|(K^\top K)^{-1} K^\top \xi_\sigma\|^2. \tag{16}$$

If we add more noise, the expected squared error $\mathbb{E}[\|x - \hat{x}\|^2]$ will be increased, but at the cost that the network will lose performance. However, under a certain noise level, applying RS can further amplify the reconstruction error (although the noise will also be sparsified).

To demonstrate this, we consider a single layer network with a one digit input $x \in \mathbb{R}$, a constraint vector $k \in \mathbb{R}^d$ and corresponding gradient vector $g \in \mathbb{R}^d$:

Theorem 5.1. *Conduct an attack using Equation (16) on the target data x, consider that the gradient g is perturbed by isotropic Gaussian noise $\xi_\sigma \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$ and further sparsified with sparsification rate $r \in (0, 1)$. The expected squared error of reconstruction is:*

$$\mathbb{E}[\|x - \hat{x}\|^2] \geq \frac{\sigma^2}{(1-r)\|k\|^2}. \tag{17}$$

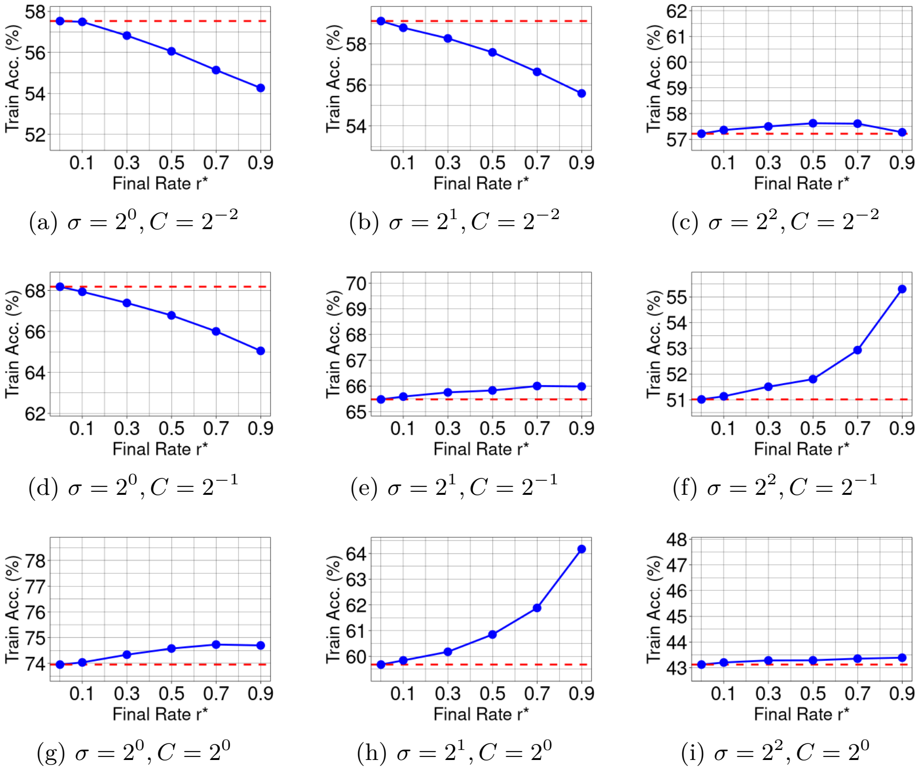
We see the expected squared error has a lower bound which increases monotonically with the sparsification rate r. We also verify this phenomenon under a more general attacking scenario in Section 6.3.

6 Experiments

Setup: Our code is implemented in PyTorch (Paszke et al., 2019b). To compute the gradients of an individual example in a mini-batch we use the BackPACK package (Dangel et al., 2020). Cumulative privacy loss has been tracked with the Opacus package, which adopts Rényi differential privacy (Mironov, 2017; Balle et al., 2020). Shuffling and partitioning the dataset into batches is generally assumed as subsampling, which implies privacy amplification (Balle et al., 2018; Wang et al., 2019; Mironov et al., 2019). We focus on DP image classification and run all experiments on a cluster within the same container environment 5 times using the same group of 5 random seeds. The supplementary material contains our code.

6.1 Evidence of the trade-off and its uniqueness for DP-SGD

Theorem 3.4 implies that when the noise term Δ_σ is dominant in the convergence bound, the trade-off induced by RS can achieve a smaller upper bound on the convergence rate. In the context of DP training, the number of iterations is limited, so faster convergence means better accuracy. Since $\Delta_\sigma \propto \sigma^2 C^2$, we conduct experiments on various combinations of σ and C to verify our analysis. When conducting RS, we always adopt gradual cooling as discussed in Section 4.1 to reach the final sparsification rate. Based on Figure 2, except the last Figure 2i, where due to excessive noise the network does not converge after a few training epochs, we see: (i) from left to right, as σ goes up, more RS is preferred; (ii) from top to bottom, as C increases, more can be gained from RS; (iii) the trade-off has the same tendency for the same value of $\sigma^2 C^2$. These observations verify our analysis. We note that as $C^2\sigma^2$ increases to 2^2, i.e. Figure 2f and 2h, RS can improve the performance of DP-SGD by more than 4%. Additionally, when $C^2\sigma^2 = 1$, i.e. Figure 2c, 2e, 2g, The performance is maintained under RS (or slightly increased). It should be noted that such a flat trend is also desirable as sparse gradients produced by RS are beneficial for private learning as discussed in Section 5.

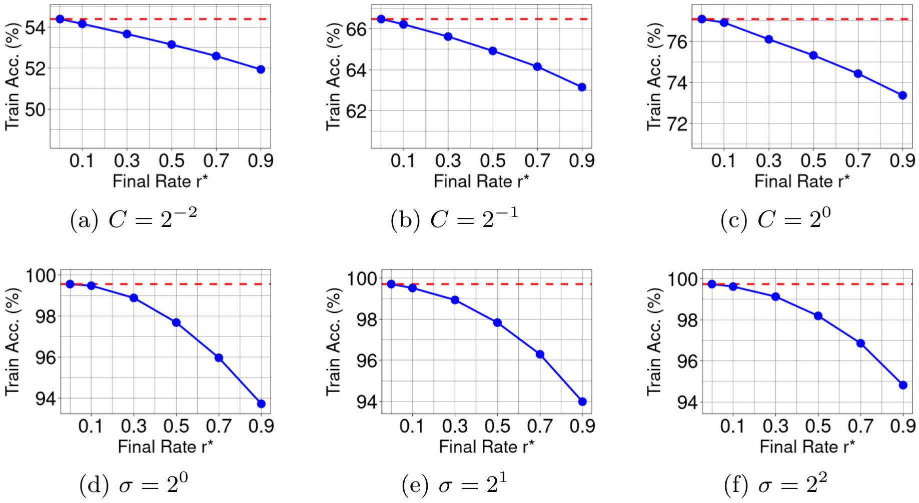


(a) $\sigma = 2^0, C = 2^{-2}$ (b) $\sigma = 2^1, C = 2^{-2}$ (c) $\sigma = 2^2, C = 2^{-2}$

(d) $\sigma = 2^0, C = 2^{-1}$ (e) $\sigma = 2^1, C = 2^{-1}$ (f) $\sigma = 2^2, C = 2^{-1}$

(g) $\sigma = 2^0, C = 2^0$ (h) $\sigma = 2^1, C = 2^0$ (i) $\sigma = 2^2, C = 2^0$

Figure 2: Train accuracy vs. final sparsification rate r^* over various combinations of noise multiplier σ and clipping bound C, red dashed line marks the performance of no sparsification. We set batch size to 1000 and train for 100 epochs, the network is DP-CNN. The result shows that the trade-off favors RS as $\Delta_\sigma \propto \sigma^2 C^2$ increasing, except the last Figure 2i, where due to excessive noise the network degrades after a few epochs.



(a) $C = 2^{-2}$ (b) $C = 2^{-1}$ (c) $C = 2^0$

(d) $\sigma = 2^0$ (e) $\sigma = 2^1$ (f) $\sigma = 2^2$

Figure 3: Train accuracy vs. final sparsification rate r^* over SGD with gradient clipping and noisy SGD (see Algorithms 2 and 3 for pseudo code), other settings keep the same as Figure 2, red dashed line marks the performance of no sparsification. Without noisification or gradient clipping RS only leads to utility drop, which matches our theoretical analysis and indicates that the trade-off is a unique property of DP-SGD.

Dataset	Approaches	ε	d	Baseline	RS (ours)	Difference
CIFAR10	H-CNN	3.0	187K	69.7 ± 0.19	**70.0 ± 0.11**	+0.3
		1.0		62.4 ± 0.13	**63.2 ± 0.10**	+0.8
	DP-CNN	3.0	550K	62.8 ± 0.10	**64.3 ± 0.17**	+1.5
		1.0		52.5 ± 0.25	**55.1 ± 0.11**	+2.6
SVHN	H-CNN	3.0	187K	85.9 ± 0.06	**86.8 ± 0.13**	+0.9
		1.0		80.7 ± 0.15	**82.2 ± 0.19**	+1.5
	DP-CNN	3.0	550K	83.4 ± 0.11	**84.5 ± 0.14**	+1.1
		1.0		76.0 ± 0.05	**79.1 ± 0.13**	+3.1
FMNIST	H-CNN/S	3.0	33K	88.9 ± 0.09	**89.2 ± 0.07**	+0.3
		1.0		85.8 ± 0.17	**87.0 ± 0.08**	+1.2
	DP-CNN/S	3.0	26K	86.6 ± 0.09	**87.4 ± 0.10**	+0.8
		1.0		83.2 ± 0.10	**84.5 ± 0.16**	+1.3

Table 2: Test accuracy (% ± SEM) before and after adopting random sparsification. The difference of mean accuracy is presented in the last column.

Theorems 3.1 and 3.3 suggest that the trade-off of DP-SGD is jointly established by noisification and clipping, as with either absent, RS just reduces performance. To verify this, we conduct experiments on two ablation variants of DP-SGD: noisy SGD and SGD with gradient clipping (pseudo code is given in Appendix E). Figure 3 demonstrates that in the absence of either noisification or gradient clipping, RS always makes the convergence slower, which indicates that the trade-off is a unique property of DP-SGD.

6.2 Improving performance of DP networks

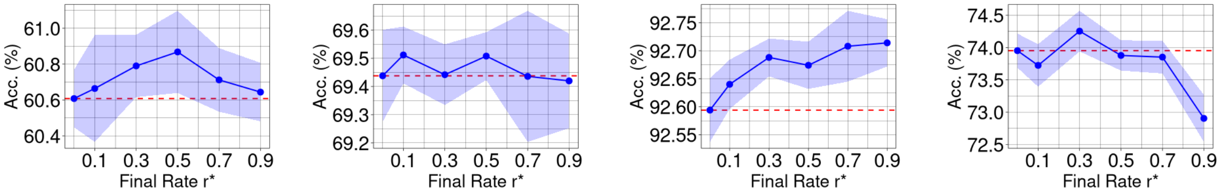
In practice, the clipping bound C and noise multiplier σ are tuned as hyperparameters for the networks to achieve the best performance under a certain privacy budget. To investigate whether RS has a chance to achieve better performance in practical settings, we conduct experiments on following baselines of DP image classification: (i) DP-CNN (Papernot et al., 2021): a SOTA network for training from scratch; (ii) H-CNN (Tramer & Boneh, 2021): a SOTA network for training solely on a private dataset, which uses handcrafted features.

We adopt the best hyperparameters provided in previous works, then we do grid search for baselines and RS over the clipping bound $C \in \{0.1, 0.5, 1\}$ where $C = 0.1$ is given in previous works, and epochs $E \in \{E^*,\ 1.2 \cdot E^*,\ 1.5 \cdot E^*\}$ (σ is adapted accordingly), where E^* is the best given in previous works for different frameworks. When conducting RS we use gradual cooling and search for the final sparsification rate $r^* \in \{0.5,\ 0.7,\ 0.9\}$. We also find an interesting scaling rule to efficiently find the optimal hyperparameters for DP frameworks which is given in Appendix H. As a result, some baseline performances we give are higher than reported in previous works. As in the previous section, we observe that RS is beneficial when Δ_σ is dominant, thus we conduct experiments at high privacy regimes, i.e. $\varepsilon \in \{1, 3\}$.

Table 2 shows that RS improves the performance of baselines and the gain is significant for smaller ε, where more noise is added. We further note that DP-CNN generally gains more from RS, as it has comparatively more parameters and Δ_σ scales with the gradient dimension d.

6.3 Advantages of sparse gradients

Beyond the improvement in performance, sparse gradients also offer additional advantages in the area of private machine learning as discussed in Section 5. To investigate the sparsity we can achieve, we directly plugged RS into various frameworks, including: DP-TL (Tramer & Boneh, 2021): a transfer learning framework, and GEP (Yu et al., 2021a): a projected DP-SGD framework, as well as DP-CNN and H-CNN, using the hyperparameters of no sparsification given in previous works (so the same iterations). Figure 4 shows that we can achieve up to 0.7 or 0.9 final sparsification rate, i.e. 0.35 or 0.45 average sparsity under gradual cooling, while maintaining the same performance.



(a) DP-CNN, $\varepsilon = 3$ (b) H-CNN, $\varepsilon = 3$ (c) DP-TL, $\varepsilon = 2$ (d) GEP, $\varepsilon = 8$

Figure 4: Test accuracy (% ± SEM) vs. final sparsification rate r^* when random sparsification is directly plugged into different frameworks using the hyperparameters of no sparsification given in previous works.

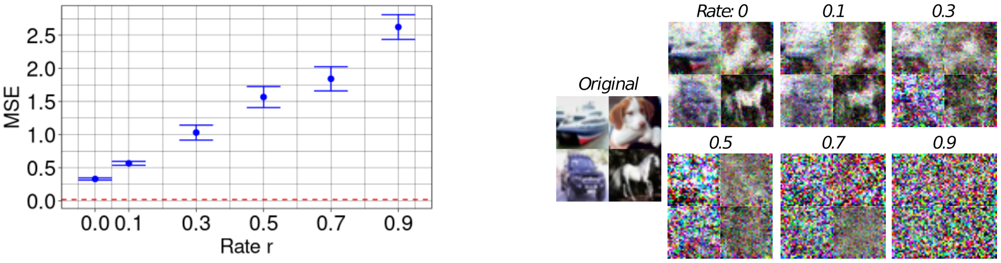


Figure 5: MSE of reconstruction using IG (Geiping et al., 2020) vs. sparsification rate r. In the non-private learning scheme, the MSE is 0.02 ± 0.02 represented by the red dashed line. The victim's network is ResNet34 and added noise is sampled from $\mathcal{N}(0, 10^{-3})$. The results are computed on 200 images from CIFAR10, four example images are: ship, dog, car, horse.

In Section 5 we have theoretically proven that with sparse representation, the communication overhead of DP federated learning and additional cost of projected DP-SGD are reduced to $\tilde{O}(1 - r)$. We also provide Theorem 5.1 to show that the sparse gradients produced by RS strengthen the privacy against gradient reconstruction attacks through an example of reconstruction using the linear relation between data and gradient. To verify this advantage under a general setting, we conduct experiments on ResNet34 (He et al., 2016) using the attack approach Inverting Gradients (IG) (Geiping et al., 2020), see Figure 5.

7 Discussion and conclusion

During DP training, it is possible to save the perturbed gradients and accordingly find the coordinates whose gradients were small and possibly continue being insignificant in subsequent iterations. One might therefore expect better performance from a ranked sparsification algorithm, which for example ranks the mean of the perturbed gradients of last epoch and sparsifies accordingly. However, we find that due to the dominant noise in our experimental settings, such ranking is in fact random, while on the other hand the resulting random sparsification still improves the performance. Similarly, other methods that select unimportant coordinates more precisely, e.g. DP selection or methods with public datasets, also cannot fully eliminate randomness. However, such random sparsification may still be beneficial for DP-SGD, which was not realized previously. More discussion are provided in Appendix I.

In this work we theoretically analyze DP-SGD with random sparsification and identify a special trade-off between internal components in the convergence bound, as the same trade-off is not presented in other popular SGD schemes. Our analyses are supported by the empirical evidence. Although the analyses do not imply that RS is always better than no sparsification, it is possible and appears to result in significant improvement in a range of realistic settings. With the proposed efficient and lightweight approach of RS, we successfully improve the performance of SOTA works, while the produced sparse gradients also provide additional advantages for key problems in private learning.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

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

Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 2496–2506. PMLR, 2020.

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

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Xiangyi Chen, Steven Z. Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. In *NeurIPS*, 2020.

Georgios Damaskinos, Celestine Mendler-Dünner, Rachid Guerraoui, Nikolaos Papandreou, and Thomas Parnell. Differentially private stochastic coordinate descent. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7176–7184, May 2021. doi: 10.1609/aaai.v35i8.16882.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. In *International Conference on Learning Representations*, 2020.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9:211–407, 2014.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, pp. 16937–16947, 2020.

Mary Gooneratne, Khe Chai Sim, Petr Zadrazil, Andreas Kabel, Françoise Beaufays, and Giovanni Motta. Low-rank gradient approximation for memory-efficient on-device training of deep neural network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3017–3021, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private erm with adagrad rates
via publicly estimated subspaces. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134, pp. 2717–2746, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*, pp. 190–198, 2020.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners, 2021.

Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated SGD under local differential privacy with top-k dimension selection. In Yunmook Nah, Bin Cui, Sang-Won Lee, Jeffrey Xu Yu, Yang-Sae Moon, and Steven Euijong Whang (eds.), *Database Systems for Advanced Applications*, 2020.

Paul Mangold, Aurélien Bellet, Joseph Salmon, and Marc Tommasi. Differentially private coordinate descent for composite empirical risk minimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14948–14978. PMLR, 17–23 Jul 2022.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Ilya Mironov. Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism, 2019.

Opacus. Opacus PyTorch library. Available from opacus.ai.

Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9312–9321, 2021.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019a.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019b.

Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X. Yu, Sashank J. Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd, 2019.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 909–910, 2015.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1139–1147, 2013.

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

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. *PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization.* 2019.

Di Wang and Jinhui Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1182–1189, 2019.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled renyi differential privacy and analytical moments accountant. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 2019.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 30th International Conference on Machine Learning*, 2011.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16337–16346, 2021.

Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021a.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning (ICML)*, 2021b.

Huanyu Zhang, Ilya Mironov, and Meisam Hejazinia. Wide network learning with differential privacy, 2021.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020.

Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *International Conference on Learning Representations*, 2021.

Junyi Zhu and Matthew B. Blaschko. R-GAP: Recursive gradient attack on privacy. In *International Conference on Learning Representations*, 2021.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019.

Appendix

In this appendix we provide full statements and proofs of our analyses (Appendix A-D). The pseudo code for noisy SGD, SGD with gradient clipping and DP-SGD with ranked sparsification are presented in Appendix E. Experimental results to verify the approximately symmetric property of the gradient deviation distribution $p(\xi_t)$ are demonstrated in Appendix F. More insights of gradual cooling are presented in Appendix G. Scaling rule for efficient hyperparameter tuning is discussed in Appendix H. More details of DP-SGD with ranked sparsifcation and connection between RS and selective sparsification or compression methods are given in Appendix I.

Contents

A Proof of applying random sparsification to noisy SGD **17**

A.1 Proof of Theorem 3.1 17

B Proof of applying random sparsification SGD with gradient clipping **18**

B.1 Proof of Lemma 3.1 18

B.2 Proof of Equation 9 and Equation 10 19

B.3 Proof of Lemma 3.3 19

C Proof of applying random sparsification to DP-SGD **21**

C.1 Proof of Lemma 3.4 21

C.2 Proof of Theorem 3.4 21

C.3 Convergence bound with privacy budget variables 21

D Proof of privacy against gradient reconstruction attacks **22**

D.1 Proof of Theorem 5.1 22

E Additional algorithms **24**

F Approximately symmetric distribution of the gradient deviation **26**

G Further discussion of gradual cooling **27**

H Scaling rule for hyperparameter tuning **28**

I DP-SGD with ranked sparsification **30**

A Proof of applying random sparsification to noisy SGD

Lemma A.1. *u, $v \in \mathbb{R}^d$ are two arbitrary vectors, $m \in \{0,1\}^d$ is a random mask with sparsification rate r. We have the following expectation, $\forall\, k \in \mathbb{Z}^+$,*

$$\mathbb{E}[\langle m^k \odot u, v\rangle] = (1-r)\mathbb{E}[\langle u, v\rangle]. \tag{18}$$

Proof.

$$\begin{aligned}\mathbb{E}[\langle m^k \odot u, v\rangle] &= \mathbb{E}[\langle m \odot u, v\rangle] &(19)\\ &= \sum_i \mathbb{E}[m_i]\mathbb{E}[u_i v_i] &(20)\\ &= (1-r)\mathbb{E}[\langle u, v\rangle]. &(21)\end{aligned}$$

□

Corollary A.2. *Following the notations of Lemma A.1, we have the expectation below:*

$$\mathbb{E}[\langle m \odot u, m \odot v\rangle] = (1-r)\mathbb{E}[\langle u, v\rangle]. \tag{22}$$

Proof.

$$\begin{aligned}\mathbb{E}[\langle m \odot u, m \odot v\rangle] &= \mathbb{E}[\langle m^2 \odot u, v\rangle] &(23)\\ &\overset{18}{=} (1-r)\mathbb{E}[\langle u, v\rangle]. &(24)\end{aligned}$$

□

A.1 Proof of Theorem 3.1

Proof. From the G-Lipschitz smoothness, and noting that Gaussian noise is added to the aggregated gradient and will be divided by batch-size B before updating the model, we have:

$$\begin{aligned}\mathcal{L}_{t+1} &\le \mathcal{L}_t + \langle \nabla w_t, w_{t+1} - w_t\rangle + \frac{G}{2}\|w_{t+1} - w_t\|^2 &(25)\\ &= \mathcal{L}_t - \gamma\langle \nabla w_t, m \odot (\nabla w_t + \frac{1}{B}(\sum_i \xi_{t,i} + \xi_\sigma))\rangle + \frac{G\gamma^2}{2}\|m \odot (\nabla w_t + \frac{1}{B}(\sum_i \xi_{t,i} + \xi_\sigma))\|^2. &(26)\end{aligned}$$

Taking expectations on both sides and rearranging:

$$\begin{aligned}&\gamma\mathbb{E}[\langle \nabla w_t, m \odot (\nabla w_t + \frac{1}{B}(\sum_i \xi_{t,i} + \xi_\sigma))\rangle]\\ &\le \mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{G\gamma^2}{2}\mathbb{E}[\|m \odot (\nabla w_t + \frac{1}{B}(\sum_i \xi_{t,i} + \xi_\sigma))\|^2],\end{aligned} \tag{27}$$

from which we can obtain:

$$\begin{aligned}(1-r)\gamma\|\nabla w_t\|^2 &\overset{18}{\le} \mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{(1-r)G\gamma^2}{2}(\|\nabla w_t\|^2 + \mathbb{E}[\|\frac{1}{B}\sum_i \xi_{t,i}\|^2] + \frac{d\sigma^2}{B^2}) &(28)\\ &\le \mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{(1-r)G\gamma^2}{2}(\|\nabla w_t\|^2 + \frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}). &(29)\end{aligned}$$

Combining $\|\nabla w_t\|^2$, and plugging $\gamma = 1/G$ into the inequality, we have:

$$(1-r)(\frac{1}{G} - \frac{1}{2G})\|\nabla w_t\|^2 \le \mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{1-r}{2G}(\frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}). \tag{30}$$

After rearranging:

$$\|\nabla w_t\|^2 \leq \frac{2G}{1-r}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}. \tag{31}$$

It is worth emphasizing that the factor $1-r$ on the r.h.s of Equation 30 is canceled out by the same factor from the l.h.s. Consider the average over T steps:

$$\begin{aligned}
\frac{1}{T}\sum_t \|\nabla w_t\|^2 &\leq \frac{2G}{(1-r)T}\sum_t \mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{1}{T}\sum_t(\frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}) &(32)\\
&= \frac{2G}{(1-r)T}\mathbb{E}[\mathcal{L}_1 - \mathcal{L}_{T+1}] + \frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2} &(33)\\
&\leq \frac{2G}{(1-r)T}\mathbb{E}[\mathcal{L}_1 - \mathcal{L}^*] + \frac{\sigma_g^2}{B} + \frac{d\sigma^2}{B^2}, &(34)
\end{aligned}$$

□

B Proof of applying random sparsification SGD with gradient clipping

Lemma B.1. *$u \in \mathbb{R}^d$ is an arbitrary vector, $m \in \{0,1\}^d$ is a random mask with sparsification rate r. We have the following inequality:*

$$\mathbb{E}[\|m \odot u\|] \geq (1-r)\|u\|. \tag{35}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\|m \odot u\|] &= \frac{1}{\|u\|}\mathbb{E}[\|m \odot u\|\|u\|] &(36)\\
&\geq \frac{1}{\|u\|}\mathbb{E}[\|m \odot u\|^2] &(37)\\
&\overset{22}{=} \frac{1}{\|u\|}(1-r)\mathbb{E}[\langle u, u\rangle] &(38)\\
&= (1-r)\|u\|. &(39)
\end{aligned}$$

Equality holds when $r = 0$. □

B.1 Proof of Lemma 3.1

Proof. Following from the G-smoothness assumption, we have:

$$\begin{aligned}
\mathcal{L}_{t+1} &\leq \mathcal{L}_t + \langle \nabla w_t, w_{t+1} - w_t\rangle + \frac{G}{2}\|w_{t+1} - w_t\|^2 &(40)\\
&= \mathcal{L}_t - \gamma\langle \nabla w_t, \bar{g}_t\rangle + \frac{G\gamma^2}{2}\|\bar{g}_t\|^2. &(41)
\end{aligned}$$

Taking expectations on both sides and rearranging, we have:

$$\begin{aligned}
\mathbb{E}[\langle \nabla w_t, \bar{g}_t\rangle] &\leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{G\gamma}{2}\mathbb{E}[\|\bar{g}_t\|^2] &(42)\\
&\leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \gamma\frac{GC^2}{2}, &(43)
\end{aligned}$$

□

B.2 Proof of Equation 9 and Equation 10

Similar to Lemma 3.1, consider gradient sparsification then clipping, we can obtain:

$$\mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] \leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \gamma \frac{GC^2}{2}. \tag{44}$$

Now focusing on the l.h.s. we have:

$$\mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] = \langle \nabla w_t, \mathbb{E}[\hat{g}_t] \rangle \tag{45}$$

$$= \langle \nabla w_t, \frac{1}{B}\sum_i \mathbb{E}[m \odot g_{t,i} \cdot \min(1, \frac{C}{\|m \odot g_{t,i}\|})] \rangle \tag{46}$$

$$= \mathbb{E}[\langle \nabla w_t, m \odot g_{t,i} \cdot \min(1, \frac{C}{\|m \odot g_{t,i}\|}) \rangle] \tag{47}$$

$$= \mathbb{E}[\langle m \odot \nabla w_t, m \odot g_{t,i} \cdot \min(1, \frac{C}{\|m \odot g_{t,i}\|}) \rangle] \tag{48}$$

$$= \mathbb{E}_m\left[\mathbb{E}_{\xi_t}[\langle \nabla w'_t, g'_{t,i} \cdot \min(1, \frac{C}{\|g'_{t,i}\|}) \rangle]\right] \tag{49}$$

$$= \mathbb{E}_m\left[\mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w'_t, g'_{t,i} \cdot \min(1, \frac{C}{\|g'_{t,i}\|}) \rangle]\right] + \mathbb{E}_m[b'_t], \tag{50}$$

where we define $\nabla w'_t := m \odot \nabla w_t$, $\xi'_t := m \odot \xi_t$ with corresponding true distribution p'_t and proxy $\tilde{p}'_t$ which are projected from p_t and $\tilde{p}_t$, $b'_t := \mathbb{E}_{\xi'_t \sim p'_t}[\langle \nabla w', \hat{g}_t \rangle] - \mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w', \hat{g}_t \rangle]$. Equation 49 holds because ξ_t and m are independent. Since the projection of a symmetric distribution to a subspace is symmetric, i.e. $\tilde{p}'_t(\xi'_t) = \tilde{p}'_t(-\xi'_t)$. We can adapt Theorem 3.2 for the first term:

$$\mathbb{E}_m\left[\mathbb{E}_{\xi'_t \sim \tilde{p}'_t}[\langle \nabla w'_t, g'_{t,i} \cdot \min(1, \frac{C}{\|g'_{t,i}\|}) \rangle]\right] \geq \mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) h(\nabla w'_t) \|\nabla w'_t\|]. \tag{51}$$

B.3 Proof of Lemma 3.3

Proof. Consider the following two cases:

case 1: $\|\nabla w_t\| \leq C/4$, then $P_m(\|\nabla w'_t\| \leq C/4) = 1$,

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) h(\nabla w'_t) \|\nabla w'_t\|] \geq P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \mathbb{E}_m[h(\nabla w'_t)\|\nabla w'_t\|] \tag{52}$$

$$= P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \mathbb{E}_m[\|\nabla w'_t\|^2] \tag{53}$$

$$\overset{22}{=} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)(1-r)\|\nabla w_t\|^2 \tag{54}$$

$$= (1-r) P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t) \|\nabla w_t\|. \tag{55}$$

case 2: $\|\nabla w_t\| > C/4$, consider two events A_1: $\|\nabla w'_t\| \leq C/4$; A_2: $\|\nabla w'_t\| > C/4$ then we have:

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) h(\nabla w'_t) \|\nabla w'_t\|] \geq P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \mathbb{E}_m[h(\nabla w'_t)\|\nabla w'_t\|] \tag{56}$$

$$\overset{54}{\geq} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)(P(A_1)(1-r)\|\nabla w_t\|^2 + P(A_2) C/4 \cdot \mathbb{E}_m[\|\nabla w'_t\|]) \tag{57}$$

$$\overset{35}{\geq} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)(P(A_1)(1-r)\|\nabla w_t\|^2 + P(A_2)\frac{C}{4}(1-r)\|\nabla w_t\|) \tag{58}$$

$$\geq P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)(1-r)\frac{C}{4}\|\nabla w_t\| \tag{59}$$

$$= (1-r) P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t) \|\nabla w_t\|, \tag{60}$$

from which we conclude that:

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4)h(\nabla w'_t)\|\nabla w'_t\|] \geq (1-r)P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\|. \tag{61}$$

The inequality above implies that $\exists\, \kappa_t \geq 1 - r$ such that:

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4)h(\nabla w'_t)\|\nabla w'_t\|] = \kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\|. \tag{62}$$

We note that $\kappa_t = 1$ if $r = 0$ and under natural assumption, e.g. $P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4) > P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)$, we have $\kappa_t > 1 - r$ if $r > 0$.

Now we form the lower bound for κ_t using the condition $P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \geq \sqrt{1-r}$. We emphasize that as the network converges during training, the following property will be satisfied with the true probability distribution p_t:

$$P_{\xi_t \sim p_t}(\|\xi_t\| < 3C/4) \geq \sqrt{1-r}. \tag{63}$$

So the condition of choosing $\tilde{p}_t$ will not lead to significant bias term b_t. Based on the condition, we have:

$$P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w)\|\nabla w\| \geq \sqrt{1-r}h(\nabla w)\|\nabla w\|. \tag{64}$$

Consider two cases:

case 1: $\|\nabla w\| \geq C/4$, r.h.s. of Equation 64 has:

$$\begin{align}
\sqrt{1-r}h(\nabla w)\|\nabla w\| &\geq h(\nabla w)((1-r)\|\nabla w\|^2)^{1/2} \tag{65}\\
&\overset{22}{=} h(\nabla w)(\mathbb{E}_m[\|\nabla w'\|^2])^{1/2} \tag{66}\\
&\geq h(\nabla w)\mathbb{E}_m[\|\nabla w'\|] \tag{67}\\
&= \mathbb{E}_m[h(\nabla w)\|\nabla w'\|] \tag{68}\\
&\geq \mathbb{E}_m[h(\nabla w')\|\nabla w'\|], \tag{69}
\end{align}$$

where Equation 67 is taken according to Jensens' inequality.

case 2: $\|\nabla w\| \leq C/4$, so we have $\|\nabla w'\| \leq C/4$, then:

$$\begin{align}
\sqrt{1-r}h(\nabla w)\|\nabla w\| &= \sqrt{1-r}\|\nabla w\|^2 \tag{70}\\
&\geq (1-r)\|\nabla w\|^2 \tag{71}\\
&\overset{22}{=} \mathbb{E}_m[\|\nabla w'\|^2] \tag{72}\\
&= \mathbb{E}_m[h(\nabla w')\|\nabla w'\|]. \tag{73}
\end{align}$$

Combing two cases, we can obtain the following property:

$$\begin{align}
P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w)\|\nabla w\| &\geq \sqrt{1-r}h(\nabla w)\|\nabla w\| \tag{74}\\
&\geq \mathbb{E}_m[h(\nabla w')\|\nabla w'\|] \tag{75}\\
&\geq \mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4)h(\nabla w')\|\nabla w'\|]. \tag{76}
\end{align}$$

It is worth noting that the above inequality can also be obtained under other conditions, the one we provided, i.e. $P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) \geq \sqrt{1-r}$, is a sufficient but not necessary condition, which will be definitely satisfied as the network converges, i.e. $\|\nabla w\| \to 0$, $\|\xi_t\| \to 0$.

Combining Equation 76 with Equation 61, we obtain: $\exists\, \kappa_t \in (1-r, 1)$,

$$\mathbb{E}_m[P_{\xi'_t \sim \tilde{p}'_t}(\|\xi'_t\| < 3C/4)h(\nabla w'_t)\|\nabla w'_t\|] = \kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\|. \tag{77}$$

κ_t takes 1 when $r = 0$. □

C Proof of applying random sparsification to DP-SGD

C.1 Proof of Lemma 3.4

Proof. Following from the smoothness assumption, we have:

$$\mathcal{L}_{t+1} \leq \mathcal{L}_t + \langle \nabla w_t, w_{t+1} - w_t \rangle + \frac{G}{2}\|w_{t+1} - w_t\|^2 \tag{78}$$

$$= \mathcal{L}_t - \gamma\langle \nabla w_t, \hat{g}_t + m \odot \xi_{DP} \rangle + \frac{G\gamma^2}{2}\|\hat{g}_t + m \odot \xi_{DP}\|^2 \tag{79}$$

$$= \mathcal{L}_t - \gamma\langle \nabla w_t, \hat{g}_t \rangle - \gamma\langle \nabla w_t, m \odot \xi_{DP} \rangle + \frac{G\gamma^2}{2}\|\hat{g}_t + m \odot \xi_{DP}\|^2, \tag{80}$$

taking expectations on both sides and rearranging, we have:

$$\mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] \leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] - \mathbb{E}[\langle \nabla w_t, m \odot \xi_{DP} \rangle] + \frac{G\gamma}{2}\mathbb{E}[\|\hat{g}_t + m \odot \xi_{DP}\|^2] \tag{81}$$

$$= \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] - 0 + \frac{G\gamma}{2}(\mathbb{E}[\|\hat{g}_t\|^2] + \mathbb{E}[\|m \odot \xi_{DP}\|^2] - 0) \tag{82}$$

$$\overset{22}{=} \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{G\gamma}{2}(\mathbb{E}[\|\hat{g}_t\|^2] + (1-r)\frac{C^2\sigma^2 d}{B^2}) \tag{83}$$

$$\leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \frac{G\gamma}{2}(C^2 + (1-r)\frac{C^2\sigma^2 d}{B^2}). \tag{84}$$

□

C.2 Proof of Theorem 3.4

Proof. Combine Equation 9, 10 and Lemma 3.3:

$$\kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| + \mathbb{E}_m[b'_t] \leq \mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle]. \tag{85}$$

Substitute Lemmma 3.4 and rearrange:

$$\kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \mathbb{E}[\langle \nabla w_t, \hat{g}_t \rangle] - \mathbb{E}_m[b_t], \tag{86}$$

$$\kappa_t P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\gamma}\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}] + \gamma\frac{GC^2}{2} + (1-r)\gamma\Delta_\sigma - \mathbb{E}_m[b'_t], \tag{87}$$

$$P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa_t}(\frac{\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}]}{\gamma} + \gamma\frac{GC^2}{2} - \mathbb{E}_m[b'_t]) + \frac{1-r}{\kappa_t}\gamma\Delta_\sigma. \tag{88}$$

Since $\kappa_{1:T} \in (1-r, 1)$, then $\exists \kappa \in (1-r, 1)$, such that consider all T iterations, we have:

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma\frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b'_t]) + \frac{1-r}{\kappa}\gamma\Delta_\sigma, \tag{89}$$

□

C.3 Convergence bound with privacy budget variables

Theorem C.1 (Abadi et al. (2016)). *There exist constants u and v so that give sampling probability $q = B/N$ and the number of steps T, for any $\varepsilon < vq^2T$, DP-SGD is (ε, δ)-differentially private for any $\delta > 0$ if we choose:*

$$\sigma \geq u\frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}. \tag{90}$$

So to achieve (ε, δ)-differential privacy, we choose $\sigma = u\frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}$ and note that for DP-SGD, large batch is preferred in practice, the following corollary proves that the upper bound of true gradient norm ∇w_t decreases w.r.t. iterations T:

Corollary C.1.1. *Follow Theorem 3.4, consider the overall T iterations with learning rate $\gamma = \frac{1}{G\sqrt{T}}$, batch-size $B = \sqrt{T}$, privacy budget $(\varepsilon,\, \delta)$, we have:*

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t\sim\tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}G}{\sqrt{T}} + \frac{C^2}{2\sqrt{T}} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b_t']) + \frac{1-r}{\kappa}\Delta_\varepsilon, \tag{91}$$

where we define $\Delta_\varepsilon := \frac{dC^2u^2q^2\log(1/\delta)}{2\sqrt{T}\varepsilon^2}$.

Proof.

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t\sim\tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \tag{92}$$

$$\leq \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma\frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b_t']) + \frac{1-r}{\kappa}\gamma\Delta_\sigma \tag{93}$$

$$= \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}G}{\sqrt{T}} + \frac{C^2}{2\sqrt{T}} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b_t']) + \frac{1-r}{\kappa}\gamma\Delta_\sigma. \tag{94}$$

Focusing on $\gamma\Delta_\sigma$:

$$\gamma\Delta_\sigma = \frac{\gamma C^2dG}{2B^2}\cdot\frac{u^2q^2T\log(1/\delta)}{\varepsilon^2} \tag{95}$$

$$= \frac{\gamma C^2dG}{2}\cdot\frac{u^2q^2\log(1/\delta)}{\varepsilon^2} \tag{96}$$

$$= \frac{dC^2u^2q^2\log(1/\delta)}{2\sqrt{T}\varepsilon^2}, \tag{97}$$

Plugging equation 97 into Equation 94:

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t\sim\tilde{p}_t}(\|\xi_t\| < 3C/4)h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa}(\frac{\Delta_{\mathcal{L}}G}{\sqrt{T}} + \frac{C^2}{2\sqrt{T}} - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_m[b_t']) + \frac{1-r}{\kappa}\Delta_\varepsilon. \tag{98}$$

□

D Proof of privacy against gradient reconstruction attacks

D.1 Proof of Theorem 5.1

Proof. Analogous to $Kx = g$ (Zhu & Blaschko, 2021, Equation 15), we have the following system of linear equations:

$$m \odot kx = m \odot g. \tag{99}$$

Assume that $m \odot k \neq 0$ (otherwise the reconstruction will fail, i.e. arbitrary reconstruction error), x can be estimated through least squares:

$$\hat{x} = \left((m \odot k)^\top(m \odot k)\right)^{-1}(m \odot k)^\top(m \odot (g + \xi_\sigma)). \tag{100}$$

The squared error of reconstruction can be expressed as:

$$\|x-\hat{x}\|^2 = \|x - \left((m\odot k)^\top (m\odot k)\right)^{-1}(m\odot k)^\top (m\odot (g+\xi_\sigma))\|^2 \tag{101}$$

$$= \|x - \left((m\odot k)^\top (m\odot k)\right)^{-1}(m\odot k)^\top (g+\xi_\sigma)\|^2 \tag{102}$$

$$= \| \left((m\odot k)^\top (m\odot k)\right)^{-1}(m\odot k)^\top \xi_\sigma\|^2. \tag{103}$$

Take expectations on both sides:

$$\mathbb{E}[\|x-\hat{x}\|^2] = \mathbb{E}[\| \left((m\odot k)^\top (m\odot k)\right)^{-1}(m\odot k)^\top \xi_\sigma\|^2 \tag{104}$$

$$= \mathbb{E}[\xi_\sigma^\top \frac{m\odot k}{\|m\odot k\|^2}\frac{(m\odot k)^\top}{\|m\odot k\|^2}\xi] \tag{105}$$

$$= \mathbb{E}[\mathrm{Tr}\{\xi_\sigma\xi_\sigma^\top \frac{m\odot k}{\|m\odot k\|^2}\frac{(m\odot k)^\top}{\|m\odot k\|^2}\}] \tag{106}$$

$$= \mathrm{Tr}\{\mathbb{E}[\xi_\sigma\xi_\sigma^\top]\mathbb{E}[\frac{m\odot k}{\|m\odot k\|^2}\frac{(m\odot k)^\top}{\|m\odot k\|^2}]\} \tag{107}$$

$$= \mathrm{Tr}\{\sigma^2\boldsymbol{I}_d\mathbb{E}[\frac{m\odot k}{\|m\odot k\|^2}\frac{(m\odot k)^\top}{\|m\odot k\|^2}]\} \tag{108}$$

$$= \sigma^2\mathbb{E}[\mathrm{Tr}\{\frac{m\odot k}{\|m\odot k\|^2}\frac{(m\odot k)^\top}{\|m\odot k\|^2}\}] \tag{109}$$

$$= \sigma^2\mathbb{E}[\frac{1}{\|m\odot k\|^2}]. \tag{110}$$

We see that as more gradients are zeroed out, the expected squared error increases. In particular, according to Jensen's inequality and the convexity of $1/\|m\odot k\|^2$:

$$\mathbb{E}[\|x-\hat{x}\|^2] \geq \frac{\sigma^2}{\mathbb{E}[\|m\odot k\|^2]} \tag{111}$$

$$\overset{22}{=} \frac{\sigma^2}{(1-r)\|k\|^2}. \tag{112}$$

The lower bound of expected squared error increases monotonically with increasing sparsification rate r. □

E Additional algorithms

We provide pseudo codes for Noisy SGD with RS (Algorithm 2), SGD with gradient clipping and RS (Algorithm 3), and DP-SGD with ranked sparsification (Algorithm 4).

Algorithm 2 Noisy SGD with Random Sparsification

```
Input: Initial parameters $w_0$; Epochs $E$; Batch size $B$; Sparsification rate: $r^*$; Momentum: $\mu$; Learning rate $\gamma$; Noise multiplier $\sigma$.
for $e = 0$ to $E - 1$ do
    ▷ Gradual cooling ◁
    $r(e) = r^* \cdot \frac{e}{E-1}$;
    ▷ Generate a random mask every epoch ◁
    $m \in \{0, 1\}^d$, s.t. $\|m\|_1 = d \cdot (1 - r(e))$;
    for $t = 0$ to $T - 1$ do
        ▷ Compute aggregated gradients ◁
        $g_t = \sum_i \nabla \ell(w_t, x_i)$;
        ▷ Sparsify gradient ◁
        $g'_t = m \odot g_t$;
        ▷ Add sparsified noise ◁
        $\hat{g}_t = \frac{1}{B}(g'_t + m \odot \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d))$;
        ▷ Update parameters ◁
        $v_{t+1} = \mu \cdot v_t + \hat{g}_t, w_{t+1} = w_t - \gamma v_{t+1}$;
```

Algorithm 3 SGD with Gradient clipping and Random Sparsification

```
Input: Initial parameters $w_0$; Epochs $E$; Batch size $B$; Sparsification rate: $r^*$; Clipping bound: $C$; Momentum: $\mu$; Learning rate $\gamma$.
for $e = 0$ to $E - 1$ do
    ▷ Gradual cooling ◁
    $r(e) = r^* \cdot \frac{e}{E-1}$;
    ▷ Generate a random mask every epoch ◁
    $m \in \{0, 1\}^d$, s.t. $\|m\|_1 = d \cdot (1 - r(e))$;
    for $t = 0$ to $T - 1$ do
        ▷ For each $x_i$ in the minibatch of size $B$ ◁
        $g_{t,i} = \nabla \ell(w_t, x_i)$;
        ▷ Sparsify gradient ◁
        $g'_{t,i} = m \odot g_{t,i}$;
        ▷ Clip each individual gradient ◁
        $\hat{g}_t = \frac{1}{B} \sum_i g'_{t,i} \cdot \min(1, C/\|g'_{t,i}\|)$;
        ▷ Update parameters ◁
        $v_{t+1} = \mu \cdot v_t + \hat{g}_t, w_{t+1} = w_t - \gamma v_{t+1}$;
```

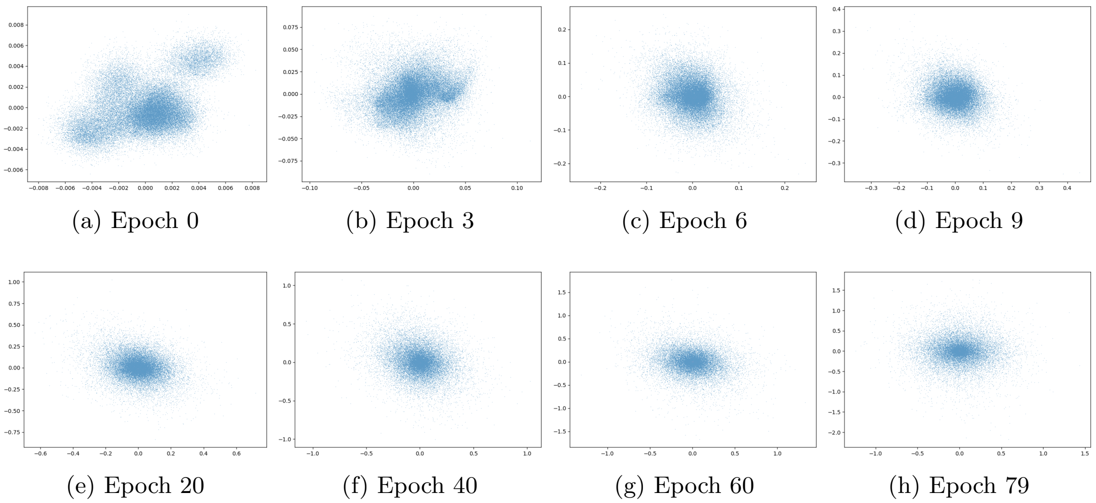
Algorithm 4 DP-SGD with Ranked Sparsification

Input: Initial parameters w_0; Epochs E; Batch size B; Sparsification rate: r^*; Clipping bound: C; Momentum: μ; Learning rate γ; Noise multiplier σ.

```
for $e = 0$ to $E - 1$ do
    $z_e = \{0\}^d$;
    ▷ Gradual Cooling ◁
    $r(e) = r^* \cdot \frac{e}{E-1}$;
    ▷ Generate a ranked mask every epoch ◁
    if $e$ is 0 then
        $m = \{1\}^d$;
    else
        Sort the indices $[1, ..., d]$ with respect to the magnitude of aggregated gradient of the last epoch
        $|z^{e-1}|$ in ascending order then set the first $d \cdot r(e)$ positions in mask $m$ to 0 and the rest to 1;
    for $t = 0$ to $T - 1$ do
        ▷ For each $x_i$ in the minibatch of size B ◁
        $g_{t,i} = \nabla \ell(w_t, x_i)$;
        ▷ Sparsify gradient ◁
        $g'_{t,i} = m \odot g_{t,i}$;
        ▷ Clip each individual gradient ◁
        $\hat{g}_{t,i} = g'_{t,i} \cdot \min(1, C/\|g'_{t,i}\|)$;
        ▷ Add sparsified noise ◁
        $\tilde{g}_t = \frac{1}{B}(\sum_i \hat{g}_{t,i} + m \odot \mathcal{N}(0, C^2\sigma^2 \boldsymbol{I}_d))$;
        ▷ Update parameters ◁
        $v_{t+1} = \mu \cdot v_t + g_t$, $w_{t+1} = w_t - \gamma v_{t+1}$;
        ▷ Save perturbed gradients ◁
        $z^e = z^e + \tilde{g}_t + \frac{1}{B}(1 - m) \odot \mathcal{N}(0, C^2\sigma^2 \boldsymbol{I}_d)$;
```

F Approximately symmetric distribution of the gradient deviation

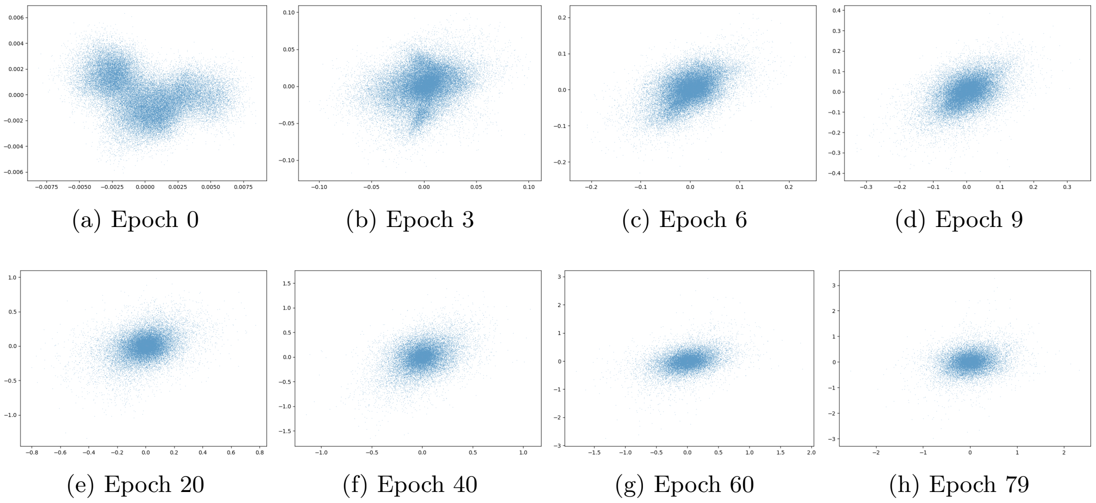
As the gradient deviation distribution $p(\xi_t)$ is high-dimensional, demonstrating and verifying its symmetricity is in general intractable. We therefore adopt random projection onto a 2D for the visualization, which is also adopted by Chen et al. (2020). Our experiments reproduce their observation that the gradient deviation distribution approximates symmetry during training with DP-SGD (see Figure 6 and 7). We verify that this property is also valid for DP-SGD with RS (see Figure 8 and 9). We have repeated the experiments 10 times, all results are qualitatively the same as presented here.



(a) Epoch 0 (b) Epoch 3 (c) Epoch 6 (d) Epoch 9

(e) Epoch 20 (f) Epoch 40 (g) Epoch 60 (h) Epoch 79

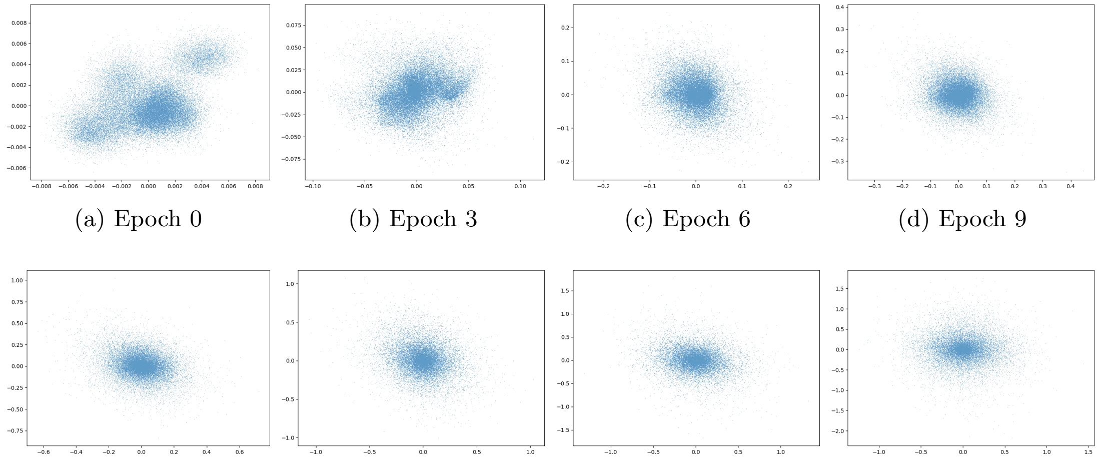
Figure 6: Gradient deviation distribution during training, projected onto a 2D using a random matrix. Network is DP-CNN, method is DP-SGD, dataset is CIFAR10.



(a) Epoch 0 (b) Epoch 3 (c) Epoch 6 (d) Epoch 9

(e) Epoch 20 (f) Epoch 40 (g) Epoch 60 (h) Epoch 79

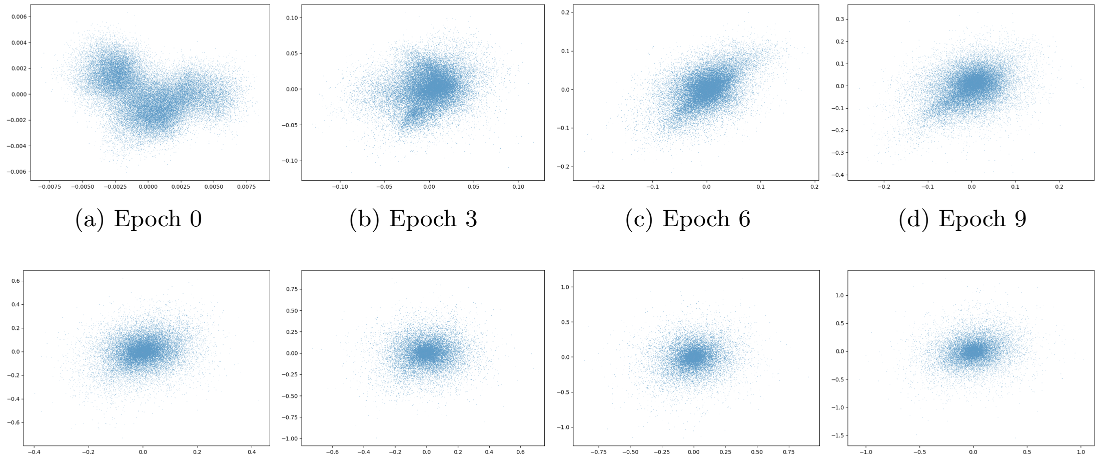
Figure 7: Gradient deviation distribution during training, projected onto a 2D using a random matrix. Network is DP-CNN, method is DP-SGD, dataset is CIFAR10.



(a) Epoch 0 (b) Epoch 3 (c) Epoch 6 (d) Epoch 9

(e) Epoch 20 (f) Epoch 40 (g) Epoch 60 (h) Epoch 79

Figure 8: Gradient deviation distribution during training, projected onto a 2D using a random matrix. Network is DP-CNN, method is DP-SGD with RS, dataset is CIFAR10.



(a) Epoch 0 (b) Epoch 3 (c) Epoch 6 (d) Epoch 9

(e) Epoch 20 (f) Epoch 40 (g) Epoch 60 (h) Epoch 79

Figure 9: Gradient deviation distribution during training, projected onto a 2D using a random matrix. Network is DP-CNN, method is DP-SGD with RS, dataset is CIFAR10.

G Further discussion of gradual cooling

For the convenience we restate Equation (14) here, and for the clarity we define $\Delta_C := \frac{\Delta_{\mathcal{L}}}{\gamma T} + \gamma \frac{GC^2}{2} - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_m[b_t']$ to denote the terms that are enlarged by RS:

$$\frac{1}{T}\sum_{t=1}^{T} P_{\xi_t \sim \tilde{p}_t}(\|\xi_t\| < 3C/4) h(\nabla w_t)\|\nabla w_t\| \leq \frac{1}{\kappa}\Delta_C + \frac{1-r}{\kappa}\gamma\Delta_\sigma. \tag{113}$$

We see $\Delta_\sigma = C^2\sigma^2 dG/2B^2$ can be computed given the constants, assume Δ_C is also given, to deduce the optimal sparsification rate, we need to relate κ to r. However, beyond giving a range $\kappa \in (1-r, 1)$, it is not possible to represent κ by r without supposing an additional assumption over the shape of the distribution of the gradient deviation, which is difficult to make. We therefore start from the empirical evidence in Figure 3.

Recall that the factor introduced by RS for noisy SGD is $1/(1-r)$ (see Theorem 3.1) and for SGD with gradient clipping is $1/k$ (see Theorem 3.3). From Figures 3a, 3b, 3c, we see that as r increases SGD with gradient clipping also obviously converges slower, which implies that $1/k$ also grows with r. From Figure 3d, 3e, 3f, we find that the performance of noisy SGD tends to drop more dramatically in a high sparsification regime, like a parabolic shape, which implies $1/k$ is comparatively more stable than $1/(1-r)$ as r changes.

Based on these observations from Figure 2, we suppose that $\kappa = \sqrt{1-r}$, which complies with the theoretical analysis that $\kappa = 1$ when $r = 0$ and $\kappa \in (1-r, 1)$. Using this relation we can express the convergence bound as a function of sparsification rate r:

$$U(r) := \frac{1}{\sqrt{1-r}}\Delta_C + \sqrt{1-r}\Delta_\sigma. \tag{114}$$

To find the optimal r^*, we can simply compute the derivative of U:

$$U'(r) = 1/2(1-r)^{-3/2}\Delta_C - 1/2(1-r)^{-1/2}\Delta_\sigma \tag{115}$$

Consider the first order condition, we have:

$$r^* = 1 - \Delta_C/\Delta_\sigma. \tag{116}$$

So when $\Delta_C > \Delta_\sigma$, i.e. the noise term is insignificant in the convergence bound, we have $r^* = 0$ is optimal. While when $\Delta_\sigma > \Delta_C$, we have $r^* = 1 - \Delta_C/\Delta_\sigma$ is optimal, and as Δ_σ becomes dominant, higher sparsification rate is preferred. We see this conclusion matches our empirical evidence in Figure 2.

Although in practice Δ_C is unknown and r^* therefore cannot be precisely estimated, the observation and conclusion above reason and support the gradual cooling. At early training stages, the network converges fast: $\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}]$ is large, while at late training stages, the optimization reaches a plateau: $\mathbb{E}[\mathcal{L}_t - \mathcal{L}_{t+1}]$ decays, in contrast Δ_σ is constant and becomes relatively large, the best sparsification rate r^* thus should be increasing during training.

H Scaling rule for hyperparameter tuning

First order momentum is commonly adopted in the optimization with DP-SGD, because momentum can alleviate oscillation and accelerate gradient descent (Sutskever et al., 2013), it is therefore believed to reduce the number of iterations of training and therefore achieve less privacy loss. However, for privacy-preserving training, momentum will also exaggerate the added Gaussian noise by incorporating current and all historical noise.

Denote velocity update: $v_{t+1} = \mu \cdot v_t + a \cdot \tilde{g}_{t+1}$, where v, μ, $\tilde{g}$ denote perturbed velocity, momentum and perturbed gradients, respectively, common implementation of SGD with momentum includes $a = 1$, e.g. for Pytorch (Paszke et al., 2019a) or $a = -\gamma$, e.g. for Tensorflow (Abadi et al., 2015). Using the expression of one step noise in Equation 4 and denoting by $\hat{v}_t$ the velocity after separating the noise, we have:

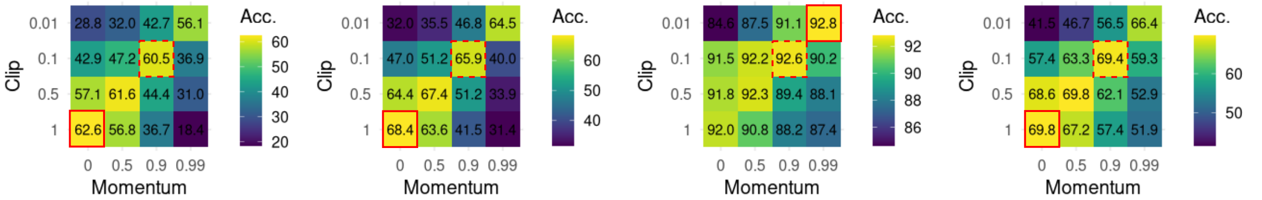
$$v_{t+1} - \hat{v}_{t+1} = (1 + \mu + \mu^2 + ... + \mu^t) \cdot a\mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_d). \tag{117}$$

After many iterations, the scalar approximates a geometric series, i.e.:

$$v_{t+1} - \hat{v}_{t+1} \approx \frac{1}{1-\mu} \cdot a\mathcal{N}(0, C^2\sigma^2\boldsymbol{I}_d), \tag{118}$$

with a residual decaying exponentially. Pulling the clipping bound C out and forming the noise as $C(1-\mu) \cdot a\mathcal{N}(0, \sigma^2\boldsymbol{I}_d)$, we present a scaling rule for adjusting C and μ, namely the local optimal values of C, μ satisfy the same $C/(1-\mu)$: the same amount of noise. As shown in Figure 10, the local optima of every column or row is located on the diagonal, where $C/(1-\mu) = C^*/(1-\mu^*)$ (C^*, μ^* are given in previous works).

The scaling rule suggests a workload saving pipeline for tuning, as it decomposes a 2D search into a line search: i). Set either C or μ fixed and search for the other (search in a row or column); ii). Adjust C, μ jointly according to the scaling rule to find the optimal combination (searching along the diagonal).



(a) DP-CNN, $\varepsilon = 3$ (b) DP-CNN, $\varepsilon = 7.53$ (c) DP-Transfer Learning, $\varepsilon = 2$ (d) Handcrafted CNN, $\varepsilon = 3$

Figure 10: Accuracy (%) under different combinations of clipping bound C and momentum μ. The diagonals represent combinations adjusted by the scaling rule based on values of C and μ given in previous works. The red dashed square marks the given hyperparameters, and the red solid square the optimal hyperparameters.

We note that the scaling rule assumes noise being directly added to velocity, for projected DP-SGD where the noise has been projected, this rule may not be applied. Additionally, for some adaptive optimizers, e.g. Adam (Kingma & Ba, 2017), computing of velocity is implemented as follows:

$$v_{t+1} = \mu \cdot v_t + (1-\mu)\tilde{g}_t, \tag{119}$$

$$v^*_{t+1} = v_{t+1}/(1-\mu^t), \tag{120}$$

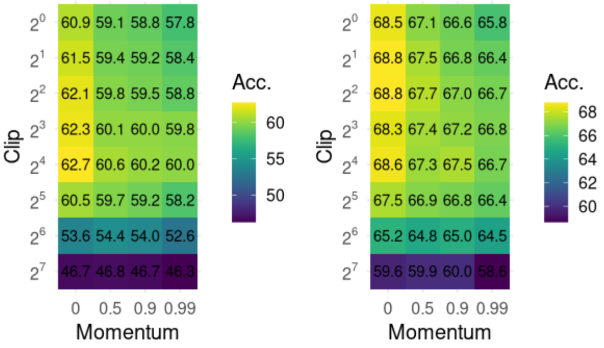
If we ignore the scaling in Equation 120 where the factor $1-\mu^t$ increases rapidly to 1, we have:

$$v_{t+1} - \hat{v}_{t+1} = (1+\mu+\mu^2+...+\mu^t)\cdot(1-\mu)\mathcal{N}(0, C^2\cdot\sigma^2\boldsymbol{I}_d) \tag{121}$$

$$\approx \frac{1}{1-\mu}\cdot(1-\mu)\mathcal{N}(0, C^2\cdot\sigma^2\boldsymbol{I}_d) \tag{122}$$

$$= \mathcal{N}(0, C^2\cdot\sigma^2\boldsymbol{I}_d). \tag{123}$$

Here we see C and μ are decoupled in term of the noise amount, which suggests that they can be tuned independently. So searching for the best combination can be decomposed into searching in a row followed by its column or vice versa, as the best μ is the same for different C and vice versa, see Figure 11.



(a) $\varepsilon = 3$ (b) $\varepsilon = 7.53$

Figure 11: Accuracy (%) under different combinations of clipping bound and first order momentum. The network is DP-CNN and the optimizer is Adam (Kingma & Ba, 2017).

I DP-SGD with ranked sparsification

For ranked sparsification, we consider a straightforward implementation which ranks the (absolute) mean of the perturbed gradients of last epoch and sparsifies accordingly. As DP is robust to post-processing, such ranked sparsification does not consume the privacy budget.

Algorithm 4 describes the proposed ranked sparsification. Note that the noise added to the gradient mean estimation z^e is not sparsified (Line 22), otherwise the coordinates that get masked out at the first iteration will receive gradient mean estimation as 0 and never get updated for all the remaining iterations, which will degrade the network. Adding noise to sparsified coordinates can give these coordinates a chance to be ranked in higher positions, while in turn the coordinates being updated but with low magnitude of their true gradient may get masked out in the next iteration.

However, it turns out that ranked sparsification *cannot* outperform random sparsification, more precisely it performs the same as RS. We further find that it seems the ranking is fundamentally random as the ranking is dominated by Gaussian noise. To demonstrate this, we run random sparsification and ranked sparsification for 100 epochs, then statistically analyze the distribution of how many times a parameter is masked out. The result shows the equivalency between these two strategies, which implies even averaging the perturbed gradients over a full epoch cannot sufficiently mitigate the noise added in gradient perturbation, see Figure 12.

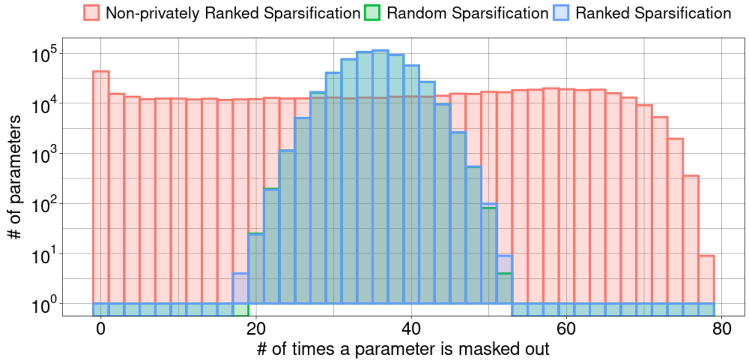


Figure 12: Histogram of the number of parameters vs. the number of times a parameter is masked out during 100 epochs training. For comparison we present non-privately ranked sparsification, i.e. ranking after excluding noise. Random sparsification and ranked sparsification mostly overlap.

This indicates that in the context of DP-SGD selecting the unimportant coordinates is difficult and stochasticity cannot be fully eliminated. Although there exist options to increase the precision of ranking, for example using public dataset (Zhou et al., 2021; Yu et al., 2021a) or via sparse vector techniques (SVT) and DP selection (Zhang et al., 2021; Dwork & Roth, 2014), there is probably a mismatch of the empirical distributions of public dataset and private dataset, while SVT and DP selection definitely contains randomness. However, as we show with RS, under such randomness, sparsification could still be beneficial for the optimization, which was not realized before.