Optimisation and Learning with Randomly Compressed Gradient Updates

Huang, Zhanliang; Lei, Yunwen; Kaban, Ata

DOI:
10.1162/neco_a_01588

License:
None: All rights reserved

Document Version
Peer reviewed version

Citation for published version (Harvard):
Huang, Z, Lei, Y & Kaban, A 2023, 'Optimisation and Learning with Randomly Compressed Gradient Updates', *Neural Computation*. https://doi.org/10.1162/neco_a_01588

Optimisation and Learning with Randomly Compressed Gradient Updates

Zhanliang Huang[1]✉, Yunwen Lei[2], and Ata Kabán[1]

[1]School of Computer Science, University of Birmingham, United Kingdom
[2]Department of Mathematics, Hong Kong Baptist University, Hong Kong, China
ZXH898@student.bham.ac.uk yunwen@hkbu.edu.hk A.Kaban@bham.ac.uk

Abstract. Gradient descent methods are simple and efficient optimisation algorithms with widespread applications. To handle high-dimensional problems, we study compressed stochastic gradient descent (SGD) with low-dimensional gradient updates. We provide a detailed analysis in terms of both optimisation rates and generalisation rates. To this end, we develop uniform stability bounds for CompSGD for both smooth and non-smooth problems, based on which we develop almost optimal population risk bounds. Then, we extend our analysis to two variants of SGD – batch and mini-batch gradient descent. Furthermore, we show these variants achieve almost optimal rates compared to their high-dimensional gradient setting. Thus, our results provide a way to reduce the dimension of gradient updates without affecting the convergence rate in the generalisation analysis. Moreover, we show that the same result also holds in the differentially private setting, which allows us to reduce the dimension of added noise with "almost free" cost.

Keywords: Gradient descent, random projection, generalisation bounds, differential privacy.

1 Introduction

Stochastic gradient descent (SGD) is a popular optimisation algorithm that has gained much attention for decades [Bottou et al., 2018]. For instance, it is well-known in convex optimisation that SGD can optimise convex functions over a convex domain with guaranteed convergence rates [Zhang, 2004]. Furthermore, it is known that the error bound of SGD can be dimension-independent which makes it favorable for high-dimensional optimisation [Hardt et al., 2016, Charles and Papailiopoulos, 2018, Chen et al., 2018, Lei and Ying, 2020, Kuzborskij and Lampert, 2018, Liu et al., 2017].

More recently, in the context of distributed optimisation, there is an increasing interest in sketching methods in SGD, where a sketch of the gradient is transmitted to the server instead of the original full gradient in order to reduce communication costs. Existing compression of gradients can mostly be categorised into two categories, 1) by sparsification that finds a sparse representation of the gradient which reduces communication costs by only transmitting the non-zero coefficients (e.g. [Wang et al., 2018, Stich et al., 2018]); 2) by quantisation that finds a less precise approximation of the gradient to reduce transmitted bits (e.g. [Alistarh et al., 2017, Agarwal et al., 2018]) such as taking the sign of values or rounding to the nearest unit integer. Besides these two categories, another approach is to consider gradient updates with reduced-dimensions while preserving accuracy performance.

A particularly useful and innovative recent approach proposed by [Kasiviswanathan, 2021] is CompSGD, which employs random projection (RP) for this purpose. Random projection is a dimensionality reduction technique that achieves a low-dimensional representation of vectors such that their distances can be preserved. The authors showed that, contrary to previous approaches (where sketching comes at the expense of an increase in the variance of the gradient), compression by random projection is able to exploit the geometry of the parameter space imposed by regularisation to make the approach lossless. In other words, CompSGD can achieve the same convergence rate as classical SGD up to logarithmic factors (in expectation). Furthermore, the authors also demonstrated empirically that the run time of CompSGD is almost the same as that of classical SGD. Hence, one can use the lower dimensional (compressed) gradient for almost 'zero cost' in its performance. Furthermore, CompSGD lends itself to applications in privacy-related optimisation, as we only need to add noise in the low dimensional space of compressed gradients [Kasiviswanathan, 2021], which then reduces the dimensionality of the injected noise.

The existing theoretical analysis of CompSGD only provides its optimisation convergence rate [Kasiviswanathan, 2021] in specific non-private settings, and has not considered differential privacy. Moreover, an analysis of optimisation can only guarantee the performance of models on training examples. However, the object of primary interest in machine learning is the generalisation error, or population risk, of the learned models. It is therefore imperative to find out to what extent the use of compressed gradients would affect the generalisation guarantees of learning algorithms. A positive finding on this question will provide a solid theoretical footing for applications in large-scale distributed and federated learning with low communication cost, such as the systems described in [Maurya and Toshniwal, 2018], and applications in differentially private learning.

In this paper, we set out to study these questions for the first time, starting with convex problems. These have a fundamental role in both learning and optimisation [Bartlett et al., 2006], and apply naturally to a variety of high dimensional sparsity-based models [Jaggi, 2011, Liu et al., 2009, Tan et al., 2018, Maurya and Toshniwal, 2018] and structure learning [Bach et al., 2012, Gonçalves et al., 2014]. Insights from the study of convex problems are indispensable to advance our understanding further. We will consider differentially private settings in both optimisation and learning problems. To tackle the latter, following a line of research on SGD [Hardt et al., 2016, Lei and Ying, 2020], we will leverage the fundamental concept of algorithmic stability to study the generalisation performance of CompSGD. This will shed light on the effects of gradient update compression in algorithm-dependent bounds while the analysis itself is independent on the particular form of predictors.

Contributions. We provide a rigorous analysis of CompSGD [Kasiviswanathan, 2021] in terms of optimisation convergence rates, as well as generalisation convergence rates. These quantify the effect of random compression of gradient updates. As a key ingredient, we employ a stability-based analysis, providing the first stability and generalisation guarantees for SGD with low-dimensional gradients. We consider both smooth and non-smooth problems, with and without privacy constraints. Furthermore, we also give the first optimisation and stability convergence analysis for two variants of CompSGD in both private and non-private settings. Our main contributions and findings are summarized as follows:

1. We prove the first uniform stability bounds of CompSGD for both smooth and non-smooth problems. Based on this, we show that CompSGD can achieve the same population risk bounds as regular SGD up to logarithmic factors. Our bound of the order $\widetilde{O}(1/\sqrt{n})$ is optimal up to a logarithmic factor, where n is the sample size. Here we use $\widetilde{O}$ to hide logarithmic factors.
2. We prove the first optimisation bounds of batch and mini-batch compressed gradient descent and show the convergence can be quicker with a larger step size in the smooth case.
3. We further extend our stability analysis to batch and mini-batch variants of CompSGD and show that these variants can achieve the exact same population risk bounds as CompSGD with fewer iterations.
4. We prove the first optimisation bound for CompSGD in the differentially private setting and show that the dimensionality of the injected noise can be significantly reduced from $\mathcal{O}(d)$ to $\mathcal{O}(\log(d))$, where d is the dimension.
5. Finally, by our stability analysis in the differentially private setting, we also prove the first generalisation bound of DP-CompSGD and show the same generalisation convergence also holds while the dimensionality is reduced.

Outline. The remainder of this paper is organised as follows. In Section 1.1 we discuss the related literature to our work. We review the CompSGD algorithm in Section 2.1, and discuss our analysis strategy in Section 2.2. We prove optimisation and generalisation bounds for the CompSGD with and without the smoothness assumption in Sections 3.1 and 3.2 respectively. In Section 4, we discuss the batch and mini-batch variants of CompSGD and present their corresponding optimisation and generalisation bounds. Finally, we present the differentially private algorithms in Section 5, where we will prove the optimisation and generalisation guarantees of DP-CompGD and DP-CompSGD with mini-batch in Sections 5.2 and 5.3 respectively.

In Table 1 we provide a summary of optimisation error rates obtained for the CompSGD algorithm and related iterative algorithms considered in this paper, in comparison with classical SGD. In Table. 2 we give the generalisation error rates for the same algorithms when employed to trained predictors, again in comparison with classical SGD. Section 6 presents the proofs of our theoretical results, and we conclude our study in Section 7.

Table 1. A summary of optimisation error rates for iterative algorithms with randomly compressed gradients with simplified parameters: The loss function is 1-Lipschitz, and the diameter of the parameter set $\mathcal{C}$ is 1. Refer to the indicated Theorems for results with general parameters. For the differentially private (DP) algorithms (last two rows), m_T denotes the maximum projection dimension used, and σ^2 is the variance of the noise added for privacy.

	Optimisation errors			
	Smooth case		Non-smooth case	
	Convergence rate	Step size	Convergence rate	Step size
Classical SGD	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ ([Shamir and Zhang, 2013, Thm. 2])	$\eta_t = \frac{\eta}{\sqrt{t}}$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ ([Shamir and Zhang, 2013, Thm.2])	$\eta_t = \frac{\eta}{\sqrt{t}}$
CompSGD	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ ([Kasiviswanathan, 2021, Thm. 2.3])	$\eta_t = \frac{\eta}{\sqrt{t}}$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ ([Kasiviswanathan, 2021, Thm. 2.3])	$\eta_t = \frac{\eta}{\sqrt{t}}$
CompGD	$\mathcal{O}\left(\frac{\log T}{T}\right)$ (Thm. 4)	$\eta_t = \eta$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ (Thm. 6)	$\eta_t = \frac{\eta}{\sqrt{t}}$
CompMinibatch	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ (Thm. 8)	$\eta_t = \frac{\eta}{\sqrt{t}}$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$ (Thm. 8)	$\eta_t = \frac{\eta}{\sqrt{t}}$
DP-CompGD	$\mathcal{O}\left(\frac{\log T\sqrt{m_T}}{T} + \frac{T\sqrt{m_T}\log(1/\delta)}{n^2\epsilon^2}\right)$ (Thm. 12)	$\eta_t = \frac{1}{\sqrt{m_T}}$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}} + \frac{\sqrt{m_T\log(1/\delta)}}{n\epsilon}\right)$ (Thm. 11)	$\eta_t = \frac{1}{\sqrt{t(1+m_T\sigma^2)}}$
DP-CompMinibatch	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}} + \frac{\log T\sqrt{m_T\log(T/n\delta)}}{n\epsilon}\right)$ (Thm. 14)	$\eta_t = \frac{1}{\sqrt{t(1+m_T\sigma^2)}}$	$\mathcal{O}\left(\frac{\log T}{\sqrt{T}} + \frac{\log T\sqrt{m_T\log(T/n\delta)}}{n\epsilon}\right)$ (Thm. 14)	$\eta_t = \frac{1}{\sqrt{t(1+m_T\sigma^2)}}$

Table 2. A summary of generalisation error rates obtained with the same algorithms as in Table 1.

	Generalisation errors			
	Smooth case		Non-smooth case	
	Convergence rate	Parameters	Convergence rate	Parameters
Classical SGD	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ ([Hardt et al., 2016, Prop.5.4])	$\eta_t = \frac{\eta}{\sqrt{T}}, T \asymp n$	$\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ ([Lei and Ying, 2020, Thm.7])	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$
CompSGD	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 2)	$\eta_t = \frac{\eta}{\sqrt{t}}, T \asymp n$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 3)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$
CompGD	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 5)	$\eta_t = \eta, T \asymp \sqrt{n}$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 7)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$
CompMinibatch	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 9)	$\eta_t = \frac{\eta}{\sqrt{t}}, T \asymp n$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$ (Thm. 10)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$
DP-CompGD	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + \frac{\log(1/\delta)}{n^{3/2}\epsilon^2}\right)$ (Thm. 13)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + \frac{\log(1/\delta)}{n^{3/2}\epsilon^2}\right)$ (Thm. 13)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$
DP-CompMinibatch	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + \frac{\log(T/n\delta)}{n^{3/2}\epsilon^2}\right)$ (Thm. 15)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + \frac{\log(T/n\delta)}{n^{3/2}\epsilon^2}\right)$ (Thm. 15)	$\eta_t = \frac{\eta}{T^{3/4}}, T \asymp n^2$

Remark 1. Note that the parameter choice can be different in Table 1 and 2 to give the best error rate. For example, the optimisation error of DP-CompGD in Table 1 is minimized by choosing a constant step size, which leads to fast convergence in the training error. However, we have to choose a smaller step size parameter for the generalisation guarantee because we need to balance the optimisation error and the estimation error terms.

1.1 Related work

The concept of algorithmic stability has existed for over thirty years [Devroye and Wagner, 1979]. The modern framework of stability analysis was established in [Bousquet and Elisseeff, 2002], where the important uniform stability was introduced and was demonstrated for regularisation schemes. The notion of uniform stability was extended to study randomised algorithms in [Elisseeff et al., 2005]. Further work by [Shalev-Shwartz et al., 2010] studied the relation between stability, uniform convergence, and learnability. An influential paper [Hardt et al., 2016] applies uniform stability to study the generalisation of SGD for convex and smooth problems, which inspires a lot of work on the stability and generalisation analysis of iterative algorithms [Chen et al., 2018, Lei and Ying, 2020, Kuzborskij and Lampert, 2018, Liu et al., 2017, Richards and Kuzborskij, 2021, Nikolakakis et al., 2022]. The smoothness assumption was removed recently by balancing stability and optimisation with small step sizes [Lei and Ying, 2020, Bassily et al., 2020]. For non-convex problems, it was shown any global minimiser would generalize well under a Polyak-Łojasiewicz condition [Charles and Papailiopoulos, 2018, Lei and Ying, 2021]. Other applications of stability can be found in structured prediction [London et al., 2016], transfer learning [Kuzborskij and Lampert, 2018], minimax optimisation [Zhang et al., 2021], hyperparameter optimisation [Bao et al., 2021] and adversarial training [Xing et al., 2021]. The notion of differential privacy has strong relations with stability as discussed in [Dwork and Roth, 2014], in the sense that private algorithms are also stable randomised algorithms. Several private SGD algorithms have been developed in the past decade [Song et al., 2013, Agarwal et al., 2018, Bassily et al., 2020, Wang et al., 2022].

The algorithm of our interest to analyse in this paper is the CompSGD proposed by [Kasiviswanathan, 2021], which we review in Section 2.1. It uses randomly compressed low-dimensional gradients to reduce the communication cost of transmitting the gradients in distributed optimisation. The random compression is implemented as a random projection (RP) technique [Dasgupta and Gupta, 2003, Gordon, 1988], which is a popular dimensionality reduction tool. RP has been previously applied to many learning algorithms in various contexts [Showkatbakhsh et al., 2018, Kabán, 2016], including privacy-related learning [Xu et al., 2017, Kenthapadi et al., 2012]. CompSGD was shown to overcome the loss of information encountered in earlier approaches of reducing the communication costs that used a sparsified or quantised version of the gradient [Alistarh et al., 2017, Alistarh et al., 2018, Wang et al., 2018, Agarwal et al., 2018, Stich et al., 2018] as an encoding process.

2 Preliminaries

In this section, we describe the problem setup and the notations used. We consider the following general setting of supervised learning. Let $\mathcal{D}$ be a probability distribution defined over some sample space $\mathcal{Z} \subseteq \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is an input domain, $\mathcal{Y} \subseteq \mathbb{R}$ is the target or label set, so each $z \in \mathcal{Z}$ consists of d attributes and a label. We draw an i.i.d. sample set $S = \{z_1, \ldots, z_n\}$ from $\mathcal{D}$. Let $f : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ be a loss function that quantifies the quality of outputs for a hypothesis represented by the parameter vector $\mathbf{w} \in \mathcal{C}$, where $\mathcal{C}$ is the parameter set, assumed to be a convex set. Given some loss function f, we aim to find a $\mathbf{w} \in \mathcal{C}$ that minimises the risk (expected loss) defined as $F(\mathbf{w}) = \mathbb{E}_{z \sim \mathcal{D}}[f(\mathbf{w}, z)]$. Since the distribution $\mathcal{D}$ is typically unknown, we work with its empirical analogue, defined as

$$F_S(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{w}, z_i). \tag{1}$$

Given a finite sample set $S \subset \mathcal{Z}$, we run an iterative gradient-based optimisation algorithm to minimise (1) over the parameter set $\mathcal{C}$, such as the Stochastic Gradient Descent (SGD), where at each step we update our weight vector $\mathbf{w} \in \mathcal{C}$ using a sample-based estimate of the gradient of f.

We note that this is a general setting, as it is not tied to any specific hypothesis class, or model, nor any properties of the underlying data distribution. It applies whenever the learning proceeds through an iterative gradient-based optimisation procedure.

We are interested in a compressive approach, whereby the optimisation is carried out in a dimensionality-reduced parameter space [Kasiviswanathan, 2021], as this naturally lends itself to private and distributed applications. This compressive optimiser will be described in the next subsection. To ensure minimal loss of information, both the algorithm and the analysis will make use of a geometric measure of complexity of the parameter set $\mathcal{C}$, namely the *Gaussian width*, defined as

$$w(\mathcal{C}) = \mathbb{E} \sup_{x \in \mathcal{C}} \langle g, x \rangle \quad \text{where } g \sim N(0, I_n), \tag{2}$$

and I_n is the $n \times n$ identity matrix.

Notation conventions. We use $\|\cdot\|$ to denote the Euclidean norm. We use the notation $[n] := \{1, \ldots, n\}$. We denote by A a (randomised) optimisation algorithm. Expectations $\mathbb{E}[\cdot]$ are taken with respect to the random sampling of S and the randomness in the algorithm A, unless otherwise specified. For a set $\mathcal{C}$, we define its diameter as $\|\mathcal{C}\| = \sup_{\mathbf{w},\mathbf{v} \in \mathcal{C}} \|\mathbf{w} - \mathbf{v}\|$. Given a linear transform $\varPhi \in \mathbb{R}^{m \times d}$, we define the transformed set $\varPhi\mathcal{C} := \{\varPhi\mathbf{w} : \mathbf{w} \in \mathcal{C}\}$. For our purposes, $\varPhi$ will be an $m \times d, m \leq d$ random matrix with i.i.d. Gaussian entries having 0 mean and variance $1/m$, commonly referred to in the dimensionality reduction literature as "random projection" (although not a projection in the geometric sense). Furthermore, we define the orthogonal projection operator $\varPi$ in the usual way as follows: for a set $\mathcal{C}$ and vector $\mathbf{w}$, the projection of $\mathbf{w}$ onto $\mathcal{C}$ is denoted by $\varPi_{\mathcal{C}}\mathbf{w}$; this is the vector $\mathbf{w}' \in \mathcal{C}$ such that $\mathbf{w}'$ has a minimal distance to $\mathbf{w}$. We use the notation $B \asymp \widetilde{B}$ if there exist universal constants $c_1, c_2 > 0$ such that $c_1\widetilde{B} \leq B \leq c_2\widetilde{B}$.

2.1 SGD optimisation with compressed gradient updates

In this section, we briefly review SGD with compressed gradient updates, as proposed by [Kasiviswanathan, 2021] – see Algorithm 1. At each iteration, this algorithm uses a random projection (RP) $\varPhi$ to compress both the weight vector $\mathbf{w}$ and the gradient vector estimated using a randomly sampled training point z, i.e. $\nabla f(\mathbf{w}_t, z)$, to a lower dimension that depends on the Gaussian width of the parameter set $\mathcal{C}$ (lines 4-6). It then takes a step in the direction of the negative gradient in the reduced parameter space, and orthogonally maps the updated parameter vector into the set $\varPhi\mathcal{C}$ (line 7). Finally, it lifts this updated parameter back into the original higher dimensional parameter set $\mathcal{C}$ (line 8). We use the weighted average output from the algorithm, as common in SGD, defined as $\bar{\mathbf{w}}_T = (\sum_{t=1}^{T} \eta_t \mathbf{w}_t)/\sum_{t=1}^{T} \eta_t$, where η_t and $\mathbf{w}_t$ are the learning rate and output of the algorithm at each iteration.

Remark 2. The compression operator $\varPhi$ used in CompSGD is a random projection matrix, which is different from compression operators used in sparsification and quantization as

Algorithm 1 CompSGD [Kasiviswanathan, 2021]

```
Inputs: Sample set $S$ of $n$ points in $\mathcal{Z}$, convex set $\mathcal{C}$, learning rate parameters $\{\eta_t\}$,
    and projection dimension parameters $\{\beta_t\}$, number of iterations $T$.
initialize $\mathbf{w}_0$ as any point in $\mathcal{C}$.
for $t = 1$ to $T$ do
    $m_t = \mathcal{O}(\min\{d, \omega(\mathcal{C})^2/\beta_t^2\})$
    set $\Phi_t \in \mathbb{R}^{m_t \times d}$ to be an i.i.d. random projection matrix
    set $\nabla f(\mathbf{w}_t, z_{i_t})$ as the gradient where $i_t$ is uniformly drawn from $[n]$
    set $\theta_t = \Pi_{\Phi_t \mathcal{C}}(\Phi_t \mathbf{w}_t - \eta_t \Phi_t \nabla f(\mathbf{w}_t, z_{i_t}))$
    pick $\mathbf{w}_{t+1}$ as any element from the set $\{\mathbf{w} \in \mathcal{C} : \Phi_t \mathbf{w} = \theta_t\}$.
end for
Output: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$
```

mentioned in the introduction. The performance of random projection is highly dependent on the geometry of the set $\mathcal{C}$ which is captured by the Gaussian width $w(\mathcal{C})$ that we use here. Note that this compression approach can be applied in conjunction with existing approaches in sparification and quantization. For example, [Kasiviswanathan, 2021] has applied CompSGD with the quantization method by [Alistarh et al., 2017] to achieve further reduction on communication costs. Furthermore, by the guarantee of Gordon's theorem [Gordon, 1988], the distance distortion due to random projection can be quantified and bounded tightly, as long as we set the appropriate projection dimension relatively to the complexity of $\mathcal{C}$.

A key result of [Kasiviswanathan, 2021] showed that the optimisation convergence of SGD with compressed gradients is the same as that of regular SGD up to logarithmic factors. They also demonstrated experimentally that the run time of using low-dimensional gradients is almost as quick as regular SGD. In turn, the benefits of using compressed gradient updates may include a reduction of the communication costs in distributed optimisation problems [Kasiviswanathan, 2021], and potentially a reduction of the dimensionality of randomised noise required in differentially private learning.

However, the existing results of the analysis of this algorithm only address optimisation convergence rates in a non-private setting. Differentially private optimisation convergence rates have not been attempted previously. Beyond optimisation, from the perspective of using this optimisation method in machine learning, we need to know what can we say about generalisation – in particular, what is the effect of random compression of gradient updates. This has not been addressed previously. To get a handle on this problem while maintaining generality, we shall appeal to stability theory, which we describe next. One of our main results will establish that the stability and generalisation bound of learning with

CompSGD are nearly the same as that of learning with regular SGD, up to logarithmic factors.

2.2 Generalisation via algorithmic stability

Denote by $A(S)$ the output of a stochastic learning algorithm A run on a sample set S. We denote by $\mathbf{w}_S^* \in \mathcal{C}$ an empirical risk minimiser (ERM), that is a hypothesis with the lowest empirical error, $\arg\min_{\mathbf{w}\in\mathcal{C}} F_S(\mathbf{w})$. We are interested in the *excess risk* of $A(S)$, which is $F(A(S)) - F(\mathbf{w}^*)$ where $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{C}} F(\mathbf{w})$ is the unknown best performing hypothesis whose parameters live in $\mathcal{C}$. The expected excess risk can be broken down into two terms, noting $\mathbb{E}[F_S(\mathbf{w}^*)] = F(\mathbf{w}^*)$:

$$\mathbb{E}_{S,A}[F(A(S)) - F(\mathbf{w}^*)] = \mathbb{E}[F(A(S)) - F_S(A(S))] + \mathbb{E}[F_S(A(S)) - F_S(\mathbf{w}^*)]. \quad (3)$$

The first term of the decomposition is called the *estimation error* due to sampling of S and the second term is the *optimisation error* induced by minimising the empirical risk. Stability properties of the algorithms are known to have a strong connection to their generalisation, as stability allows us to understand the scale of the estimation error. The notion of stability that we will employ is *uniform stability.* Uniform stability is a widespread notion of stability that drives powerful analysis [Elisseeff et al., 2005, Hardt et al., 2016, Bousquet and Elisseeff, 2002].

Definition 1 (Uniform stability). *An algorithm A is said to be ϵ-uniformly stable if we have for all $S, S' \in \mathcal{Z}^n$ that differ by at most one example,*

$$\sup_z \mathbb{E}_A[f(A(S), z) - f(A(S'), z)] \leq \epsilon. \quad (4)$$

The following powerful theorem connects uniform stability and generalisation:

Theorem 1 ([Shalev-Shwartz et al., 2010]). *If A is ϵ-uniformly stable, then*

$$|\mathbb{E}_{S,A}[F(A(S)) - F_S(A(S))]| \leq \epsilon.$$

Our general strategy will be to investigate the two terms of (3) individually and combine the results together to obtain an excess risk bound. The second term, i.e. the optimisation error is already available in some settings from [Kasiviswanathan, 2021], while the first term, i.e. the estimation error will need some work and will be obtained by establishing a suitable stability bound.

In this analysis framework, bounding the optimisation error is part of bounding the generalisation error. However, we are also interested in optimisation convergence rates, since optimisation is useful in many other application areas. Thus, we produce two sets of results: optimisation error rates, and generalisation error rates. The difference is that, for optimisation we choose the step size that achieves the best optimisation error rates, while for generalisation we choose the step size and the number of iterations to balance out the optimisation error and the estimation error in order to achieve the best generalisation error rates. Consequently, the rates can be different depending on whether our goal is optimisation or generalisation, and indeed we see examples in our summary Tables 1-2 (last two rows), where the optimisation convergence rate of an algorithm differs from its generalisation convergence rates.

In this paper, we will mainly consider the weighted average output model

$$\bar{\mathbf{w}} = \left(\sum_{t=1}^{T} \eta_t \right)^{-1} \sum_{t=1}^{T} \eta_t \mathbf{w}_t.$$

However, many parts of our results will also hold for the final output model $\mathbf{w}_T$ or other similar averaging models (e.g. average of last 10 iterations).
Throughout this paper, we will assume that the loss function f is convex and Lipschitz in its first argument. Recall, a function $\varphi : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is L-Lipschitz on $\mathcal{C} \subset \mathbb{R}^d$ with respect to the norm $\|\cdot\|$ if $\forall \mathbf{w}, \mathbf{w}' \in \mathcal{C}, \forall z \in \mathcal{Z}$ we have $|\varphi(\mathbf{w}, z) - \varphi(\mathbf{w}', z)| \leq L\|\mathbf{w} - \mathbf{w}'\|$. A differentiable function $\varphi : \mathbb{R}^d \to \mathbb{R}$ is convex over $\mathcal{C} \subset \mathbb{R}^d$ if $\forall \mathbf{w}, \mathbf{w}' \in \mathcal{C}$ we have $\varphi(\mathbf{w}) \geq \varphi(\mathbf{w}') + \langle \mathbf{w} - \mathbf{w}', \nabla \varphi(\mathbf{w}') \rangle$.

We note that, despite the popular use of SGD on non-convex problems, the theoretical analysis in the non-convex case is still very limited (e.g. privacy-related applications [Wang et al., 2019]). There are currently no generalisation analyses of SGD with compressed gradients at all, to the best of our knowledge. Therefore our aim is to provide the first insights in compressed gradient descent methods that yield low generalisation error. This will eventually lead to new insights in privacy applications and complex non-convex problems.

3 Generalisation of CompSGD

3.1 Generalisation bound under smoothness assumption

In this section, we assume that the loss function f, is convex, L-Lipschitz, and also μ-smooth. A function $\varphi : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is μ-smooth on $\mathcal{C} \subset \mathbb{R}^d$ if $\forall \mathbf{v}, \mathbf{w} \in \mathcal{C}, \forall z \in \mathcal{Z}$ we have $\|\nabla\varphi(\mathbf{v}, z) - \nabla\varphi(\mathbf{w}, z)\| \leq \mu\|\mathbf{w} - \mathbf{v}\|$, i.e. its gradient is μ-Lipschitz in its first argument on $\mathcal{C}$.

These assumptions are common and critical for convex optimisation problems as they lead to bounds on the divergence or expansiveness of the gradient updates when the algorithm is running on neighbouring sample sets [Hardt et al., 2016]. Examples of common loss functions that satisfy these assumptions are logistic loss, Huber loss and exponential loss (assuming bounded input samples). Later we also provide analysis without the smoothness assumption, which applies to e.g. the Hinge loss.

However, the bottleneck here is to account for the effects of random compressions that operate at each iteration of Alg. 1. This is a form of sketching, which creates a perturbation that was not present in classical SGD. It is not at all obvious as to whether this sketched parameter update rule is sufficiently well behaved, moreover since at each iteration the updated parameter vector $\mathbf{w}_{t+1}$ depends on the random matrix of the previous iteration, Φ_t, and these perturbations accumulate from iteration to iteration. Fortunately, it turns out that the Gaussian width defined in (2) allows us to estimate an appropriate projection dimension based on the structural complexity of the parameter set, such that the parameters learned by CompSGD from two neighbouring sets still do not diverge too much. This allows us to carry out a useful stability analysis similar to that of classical SGD, while incurring just an extra log factor.

Theorem 2 (Stability and generalisation of CompSGD under smoothness). *Assume that the loss function f is convex, μ-smooth and L-Lipschitz on $\mathbf{w} \in \mathcal{C}$, for every $z \in \mathcal{Z}$. Suppose that we run the CompSGD with step sizes $\eta_t = \frac{\eta}{\sqrt{t}} \leq 2/\mu$ for $T \asymp n$ iterations. Let $\beta_t = \frac{1}{t+1}$, $\eta_t = \frac{\eta}{\sqrt{t}}$ for some absolute constant η.*

1. *Then, the CompSGD algorithm (both $\mathbf{w}_T$ and $\bar{\mathbf{w}}_T$) is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\big(L^2 \log(n)/\sqrt{n}\big)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ of CompSGD satisfies the following generalisation bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}(L^2 \log(n)/\sqrt{n}). \tag{5}$$

Remark 3. For the vanilla SGD, excess risk bounds of the order $O(1/\sqrt{n})$ were established for SGD with $\eta_t \asymp 1/\sqrt{n}$ and $T \asymp n$ [Hardt et al., 2016]. Theorem 2 shows that CompSGD is able to achieve the same generalisation bounds (up to a log factor) by updating with compressed stochastic gradient.

Remark 4. The choice of β_t in Thm. 2 makes the projection dimension in Alg.1 scales inversely with t. This is needed for the theoretical analysis and will appear similarly for later analyses. However, in practice this appears to be not crucial, as demonstrated in [Kasiviswanathan, 2021, Appendix B.3] on relatively complex and high-dimensional real-world problems. CompSGD can be implemented in the distributed learning setting which requires less iterations to converge. Furthermore, many real world problem have very high dimensions while their intrinsic dimension may be much smaller, allowing m_t to be smaller than it is required here. Our results presented here considers the worst case scenario.

3.2 Generalisation bound without smoothness

The smoothness assumption is commonly used in the analysis for regular SGD since it simplifies the analysis for both optimisation and generalisation. However, in practice, we often encounter learning problems with non-smooth loss functions (e.g. the Hinge loss). Recently, [Lei and Ying, 2020] showed for SGD without the smoothness assumption (relaxed Hölder-continuous assumption) enjoys stability and generalisation bounds (up to constant factors) similar to that with the smooth assumption by choosing an appropriate choice of parameters. We will adapt parts of their technique here to prove the generalisation convergence for CompSGD in the non-smooth case. We show that we can obtain the same convergence as in the smooth case for CompSGD up to log factors by choosing suitable projection and learning parameters.

Theorem 3 (Stability and generalisation of CompSGD without smoothness). *Assume that the loss function f is convex and L-Lipschitz over the convex set $\mathcal{C}$. Suppose that we run the CompSGD with step sizes $\eta_t = \frac{\eta}{T^{3/4}}$ for some absolute constant η for T steps. Furthermore we let $\beta_t = \frac{1}{t+1}$ and $T \asymp n^2$.*

1. *Then CompSGD is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\big(L^2\sqrt{\log(n)}/\sqrt{n}\big)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ of CompSGD satisfies the following generalisation bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}}\right). \qquad (6)$$

From Theorem 3 we have chosen a relatively smaller step size parameter $\eta_t = \eta/T^{3/4}$ comparing to $\eta_t = \eta/\sqrt{t}$ in the smooth case. The choice of η_t here is needed for our result to have the same convergence as in the smooth case. An intuitive explanation for the smaller step size parameter is that the problem is harder without the smoothness hence we need to take more careful steps towards the optima.

Remark 5. The choice of the parameter in the non-smooth case matches with the choice for classical SGD in the same setting [Lei and Ying, 2020]. Hence Theorem 3 shows that CompSGD achieves the same generalisation bounds in the non-smooth setting with the same parameters (up to logarithmic factors).

4 Variants of CompSGD

In this section, we present the convergence guarantee of the variants of the CompSGD - batch gradient descent that uses the full gradient (section 4.1) and mini-batch gradient descent that uses the gradient of the mini-batch (section 4.2). These variants of SGD are also widely used in practice, especially in cases where we can compute the full gradient easily to make more informative updates in each iteration. In the differentially private setting, it is also desirable to use the batch gradient instead of the stochastic gradient due to the random noise injected and to limit the number of iterations required. In this section, we show that the risk bounds of these variants are the same as the rates for CompSGD. Hence, we can choose an appropriate method suited to our needs without affecting its generalisation.

4.1 Compressed gradient descent

The first variant we will analyze is the classical batch gradient descent.

Batch gradient descent utilizes the most information from the sample set S at each iteration to make accurate updates. Hence we can usually converge close to the optima using much less iterations compared to SGD which can be beneficial in many cases (e.g. private optimisations). The detail is outlined in Algorithm 2.

Since batch gradient utilises the most information from the sample set at each iteration, batch gradient descent is a special case where we can use a constant step size η to obtain a quicker convergence rate in its optimisation bound. We show that this is also the case for compressed batch gradient descent.

Algorithm 2 Compressed Gradient Descent (CompGD)

```
Inputs: Sample set $S$ of $n$ points in $\mathcal{Z}$, convex set $\mathcal{C}$, learning rate parameters $\{\eta_t\}$,
and projection dimension parameters $\{\beta_t\}$.
initialize $\mathbf{w}$ as any point in $\mathcal{C}$.
for $t = 1$ to $T$ do
    Set $m_t = \mathcal{O}(\min\{d, \omega(\mathcal{C})^2/\beta_t^2\})$
    Let $\Phi_t \in \mathbb{R}^{m_t \times d}$ be an i.i.d. random projection matrix
    compute $\hat{g}_t = \frac{1}{n}\sum_{z \in S} \nabla f(\mathbf{w}_t, z)$ as the gradient
    set $\theta_t = \Pi_{\Phi_t \mathcal{C}}(\Phi_t \mathbf{w}_t - \eta_t \Phi_t \hat{g}_t)$
    pick $\mathbf{w}_{t+1}$ to be any element from the set $\{\mathbf{w} \in \mathcal{C} : \Phi_t \mathbf{w} = \theta_t\}$.
end for
Output: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$
```

Theorem 4 (Optimisation with CompGD, smooth case). *Let f be a convex function over a convex set $\mathcal{C}$, and satisfy L-Lipschitz condition and μ-smooth condition. Then with $\eta_t = \eta \leq 1/(2\mu)$ for some absolute constant η and $\beta_t = \frac{1}{t+1}$, the compressed gradient descent satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2 \log(T)}{T}\right). \tag{7}$$

We now study the risk bounds of CompGD. Note that here we have $O(1/T)$ for its optimisation bound which is better than SGD. We also note that we have chosen a constant step size η for batch gradient because each gradient update is accurate enough to take larger steps. Hence, we only required $T \asymp \sqrt{n}$ to achieve the same generalisation convergence compared to SGD.

Theorem 5 (Stability and generalisation of CompGD, smooth case). *Assume the loss function f is convex, μ-smooth, and L-Lipschitz over the convex set $\mathcal{C}$. Suppose we run the CompGD with $\eta_t = \eta \leq 1/(2\mu)$ for $T \asymp \sqrt{n}$ steps where η is an absolute constant and $\beta_t = 1/(t+1)$.*

1. *Then CompGD satisfies uniform stability with $\epsilon_{stab} = \mathcal{O}\left(L^2/\sqrt{n}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ of CompGD satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}}\right). \tag{8}$$

While we can obtain a faster convergence rate for batch gradient descent in optimisation, the stability guarantee of CompGD is the same as CompSGD. This is also the case for classical SGD, because the samples we use for $\mathbf{w}_{t+1}, \mathbf{w}'_{t+1}$ will differ in one point every iteration with probability 1. Hence we do not obtain an improvement in the expected excess generalisation risk. However, the smoothness case will allow us to choose a larger learning rate compared with the non-smooth case.

Theorem 6 (Optimisation with CompGD, non-smooth case). *Let f be a convex function over a convex set $\mathcal{C}$ and satisfy L-Lipschitz condition. Then with $\eta_t = \eta/\sqrt{t}$ for some absolute constant η and $\beta_t = \frac{1}{t+1}$, the compressed gradient descent satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(T)}{\sqrt{T}}\right). \tag{9}$$

Theorem 7 (Stability and generalisation of CompGD, non-smooth case). *Assume the loss function f is convex, μ-smooth and L-Lipschitz over the convex set $\mathcal{C}$. Suppose we run the CompGD with $\eta_t = \eta/T^{3/4}$ for $T \asymp n^2$ steps where η is an absolute constant and $\beta_t = 1/(t+1)$.*

1. *Then CompGD satisfies uniform stability with $\epsilon_{stab} = \mathcal{O}\left(L^2\log(n)/\sqrt{n}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ of CompGD satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}}\right). \tag{10}$$

4.2 CompSGD with Mini-batch

In this section, we study the stability and generalisation of CompSGD with a mini-batch strategy. Mini-batch SGD is considered as a semi-stochastic version of gradient descent and is widely used in various applications [Konečnỳ et al., 2015, Zhao and Zhang, 2014]. Different sampling techniques may be used to sample a mini-batch depending on the application and preference. Here we will use the following sampling method for the CompSGD with mini-batch: For each iteration we sample a mini-batch B_t of size b from the sample set S without replacements, then we will sample a fresh batch from S at the next iteration so that the sampling set S is consistent.

Theorem 8 (Optimisation with Mini-batch SGD). *Let f be a convex function over a convex set $\mathcal{C}$, and satisfy L-Lipschitz condition. Then with $\eta_t = \frac{\eta}{\sqrt{t}}$ and $\beta_t = \frac{1}{t+1}$, the CompSGD with mini-batch satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(T)}{\sqrt{T}}\right). \tag{11}$$

Algorithm 3 CompSGD with Mini-batch (CompMinibatch)

```
Inputs: Sample set $S$ of $n$ points in $\mathcal{Z}$, batch size $b$, convex set $\mathcal{C}$, learning rate parameters $\{\eta_t\}$, and projection dimension parameters $\{\beta_t\}$.
initialize $\mathbf{w}$ as any point in $\mathcal{C}$.
for $t = 1$ to $T$ do
    Set $m_t = \mathcal{O}(\min\{d, \omega(\mathcal{C})^2/\beta_t^2\})$
    Let $\Phi_t \in \mathbb{R}^{m_t \times d}$ be an i.i.d. random projection matrix
    Sample a mini-batch $B_t$ of size $b$ uniformly from $S$
    compute $\hat{g}_t = \frac{1}{b}\sum_{z \in B_t} \nabla f(\mathbf{w}_t, z)$ as the gradient of the mini-batch
    set $\theta_t = \Pi_{\Phi_t \mathcal{C}}(\Phi_t \mathbf{w}_t - \eta_t \Phi_t \hat{g}_t)$
    pick $\mathbf{w}_{t+1}$ to be any element from the set $\{\mathbf{w} \in \mathcal{C} : \Phi_t \mathbf{w} = \theta_t\}$.
end for
Output: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$
```

Theorem 9 (Stability and generalisation of Mini-batch SGD, smooth case). *Assume that the loss function f is μ-smooth, convex, and L-Lipschitz for every z. Suppose that we run the CompSGD with mini-batch of size b and step sizes $\eta_t \leq 2/\mu$ for T iterates. Then with $\eta_t = \eta/\sqrt{t}$ for some absolute constant η, $\beta_t = \frac{1}{t+1}$ and $T \asymp n$, we have*

1. *The algorithm CompMinibatch is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\left(L^2 \log(n)/\sqrt{n}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}}\right). \tag{12}$$

We note that the optimisation and stability of the mini-batch achieve the same convergence rate as SGD, which is optimal up to logarithmic factors. The main advantage of mini-batch is to perform stochastic gradient descent while preventing over-randomised convergence to the optima. In the best case we will obtain the same convergence as for batch gradient descent in section 4.1. Since the mini-batch selected can be either quite small or large depending on the computational complexity desired, the upper bound for its generalisation error is identical to SGD as we show below.

Theorem 10 (Stability and generalisation of Mini-batch SGD, non-smooth case). *Assume that the loss function f is convex and L-Lipschitz for every z. Suppose that we run the CompSGD with mini-batch of size b and step sizes $\eta_t = \eta/T^{3/4}$ for some absolute constant η, $\beta_t = \frac{1}{t+1}$ and $T \asymp n^2$.*

1. *The CompSGD with mini-batch is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\left(L^2 \log(n)/\sqrt{n}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}}\right). \tag{13}$$

5 Reducing the Noise for Differentially Private Applications

5.1 A Brief Background on Differential Privacy

We provide a brief introduction of differential privacy here as a preliminary. Differential privacy (DP) is a rigorous theoretical privacy guarantee that is introduced by [Dwork, 2006], and since then DP has been a popular concept that has been widely applied to many common algorithms in computer science. Roughly speaking, DP guarantees that the participation of a particular sample will not affect the output of the algorithm, hence adversaries cannot recover particular samples from the output of a DP algorithm. We denote two datasets $S \sim S'$ if they differ by at most an example.

Definition 2 (Differential privacy [Dwork, 2006]). *A randomised algorithm M with domain $\mathcal{X}$ is (ϵ, δ)-differentially private if for all $B \subseteq Range(M)$ and for all $S \sim S' \subset \mathcal{X}$:*

$$\Pr[M(S) \in B] \leq \exp(\epsilon) \Pr[M(S') \in B] + \delta.$$

There have been many ways of achieving differential privacy for algorithms since there is a vast amount of research related to differential privacy [Dwork and Roth, 2014]. One of the common mechanism is the *Gaussian mechanism*:

Definition 3 (Gaussian mechanism [Dwork et al., 2006]). *Let $F : \mathcal{X}^n \to \mathbb{R}^d$. The algorithm with input $S \in \mathcal{X}^n$ outputs $F(S) + \mathbf{e}$ where $\mathbf{e} \sim N(0, 2\Delta^2 \log(1.25/\delta)/\epsilon^2 I_{d\times d})$ where Δ denotes ℓ_2-global sensitivity of the function F defined as $\sup_{S\sim S'} \|F(S) - F(S')\|$, is (ϵ, δ)-differentially private. Here $I_{d\times d}$ denotes the identity matrix in $\mathbb{R}^{d\times d}$.*

The Gaussian mechanism works by analysing the global sensitivity of an algorithm F and adding Gaussian noise (w.r.t. its sensitivity) to guarantee privacy. This is the most common mechanism used for differentially private gradient descent methods, since its combination with the strong composition theorem provides decent privacy guarantees while preserving good accuracy. For our purpose, the algorithm F will be our gradient computation that takes samples in $S \subset \mathcal{Z}$ as inputs and outputs a gradient in d-dimension.

5.2 Differentially Private Compressed Gradient Descent

In this section, we demonstrate and analyse the DP-SGD with compressed gradients. To guarantee differential privacy we impose the Gaussian mechanism to add Gaussian noise to the gradient updates. For classical DP gradient updates, we need to add noises in the original d-dimensional space which could be of very large size if d is large. We impose the compressed gradient updates in the private setting to add noise in a much lower dimension instead. The algorithm is introduced in the appendix of [Kasiviswanathan, 2021]. However, no convergence analysis has been done for the private setting. The detailed algorithm is outlined in Algorithm 4.

The algorithm uses a standard application of the *Gaussian mechanism* [Dwork, 2006] to guarantee (ϵ, δ)-differential privacy. The main idea of the mechanism is to perturb the gradient update at each iteration by injecting noise. A similar approach has also been taken in [Song et al., 2013] in the classical case (without projections).

Algorithm 4 Differentially private CompGD (DP-CompGD)

```
Inputs: Sample set $S = \{z_1, \dots, z_n\} \subset \mathcal{Z}^n$, privacy parameters $(\epsilon, \delta)$, step size parameters $\{\eta_t\}$ and projection parameters $\{\beta_t\}$.
initialize $\mathbf{w}_1$ as any point in $\mathcal{C}$
for $t = 1$ to $T$ do
    set $m_t = \min\{d, \omega(\mathcal{C})^2/\beta_t^2\}$
    choose projection matrix $\Phi_t \in \mathbb{R}^{m_t \times d}$ with i.i.d. entries from $\mathcal{N}(0, 1/m_t)$
    set $\sigma^2 = \frac{32L^2T\log(1/\delta)}{n^2\epsilon^2}$
    set $s_t = \frac{\|\nabla F_S(\mathbf{w}_t)\|}{\|\Phi_t \nabla F_S(\mathbf{w}_t)\|}$
    set $\theta_t = \Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}_t - \eta_t(s_t\Phi_t\nabla F_S(\mathbf{w}_t) + \mathbf{e}))$ where $\mathbf{e} \sim \mathcal{N}(0, \sigma^2 I_{m_t})$
    pick $\mathbf{w}_{t+1}$ to be any element from the set $\{\mathbf{w} \in \mathcal{C} : \Phi_t\mathbf{w} = \theta_t\}$
end for
Output: $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_T$
```

We note that the variance of the injected noise σ^2 here depends not only on the privacy parameters ϵ, δ, but also on the number of iterations T and the number of samples n. The dependence on T follows from the iterative property of gradient descent as we need to query the sample set once every epoch. The dependence on n follows from the use of the gradient $\nabla F_S(\mathbf{w}_t)$ for our algorithm. We remark that it is more preferable to use the full gradient in the privacy setting as compared to SGD if we wish to maximise accuracy because we can reduce the variance of the noise. Since differential privacy requires that the sensitivity of the gradient is bounded uniformly, we are required to set the normalization factor s_t as in the algorithm to guarantee this property. Other methods such as gradient clipping as in [Chen et al., 2020] also works similarly to bound the sensitivity uniformly. One of the main challenges in the differentially private setting is to accommodate in the analysis the normalization factor s_t on the gradient updates, which depends on the random projection Φ_t and the gradient ∇F_S. The size of the random projection Φ_t only depends on the Gaussian width of $\mathcal{C}$ (and distortion parameter β_t), meaning we have a uniform norm-preservation guarantee for elements of $\mathcal{C}$, but no guarantee on the distortion of the projected gradients $\Phi_t \nabla F_S$. This makes the convergence of the projected gradient updates difficult to analyse. We exploit the Gaussianity of Φ_t to overcome this bottleneck, and we show that the convergence of DP-CompGD is almost the same as that of high-dimensional SGD (without projection) while reducing the dimensionality of the noise and the gradient.

Theorem 11 (Optimisation with DP-CompGD, non-smooth case). *Assume that the loss function f is convex and L-Lipschitz over the convex set $\mathcal{C}$. Suppose we run the compressed GD with step sizes $\eta_t = \|\mathcal{C}\|/\sqrt{t(L^2 + m_T\sigma^2)}$ for some absolute constant η. For privacy parameters ϵ, δ we let $\sigma^2 = \mathcal{O}(L^2 \log(1/\delta)T/(\epsilon^2 n^2))$ and $\beta_t = 1/(t+1)$. Then we have that the private CompSGD satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{\log T\|\mathcal{C}\|L}{\sqrt{T}} + \frac{\log T\|\mathcal{C}\|L\sqrt{m_T \log(1/\delta)}}{n\epsilon}\right), \tag{14}$$

where $m_T = \max_{t\in[n]} m_t \leq d$.

We note that the second term in (14) has the privacy parameter ϵ in the denominator, which implies that the second term will vanish as ϵ tends to infinity (zero privacy). In that case, we recover the same convergence rate as in the non-private case as expected. We also note that m_T is dependent on the Gaussian width of the constraint set $\mathcal{C}$. This captures the dimensionality reduction from d to $\omega(\mathcal{C})$, which can be much smaller if the set has a low dimensional structure. e.g. if $\mathcal{C}$ is the ℓ_1-ball, then we have $\omega(\mathcal{C}) = \mathcal{O}(\sqrt{\log d})$.

In the case of smooth f, we can obtain a faster convergence in optimisation just as we have observed for the non-private case.

Theorem 12 (Optimisation with DP-CompGD, smooth case). *Assume that the loss function f is convex, μ-smooth and L-Lipschitz over the convex set $\mathcal{C}$. Suppose that we run the compressed GD with step sizes $\eta_t = \frac{\|\mathcal{C}\|}{L\sqrt{m_T}} \leq 1/(4\mu)$ for some absolute constant η. For privacy parameters ϵ, δ we let $\sigma^2 = \mathcal{O}(\log(1/\delta)L^2T/(\epsilon^2 n^2))$ and $\beta_t = 1/(t+1)$. Then we have that the private CompSGD satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{L\|\mathcal{C}\|\log T\sqrt{m_T}}{T} + \frac{LT\sqrt{m_T}\log(1/\delta)}{n^2\epsilon^2}\right),$$

where $m_T = \max_{t\in[n]} m_t \leq d$.

Similar to the non-smooth case, we note that when the privacy parameters ϵ, δ converge to infinity, we will recover the same convergence bound as in the non-private case for compressed gradient descent.

Remark 6. Notice here that we have specified the learning rate η_t needed for the optimisation error bounds in Thm. 11 and Thm. 12. The only necessary dependence in η_t is t, σ^2, and m_T for the dimensionality dependence. Other constants can be chosen freely without affecting the result (up to constant factors) in a similar way as previous results (as in e.g. Thm. 4).

Theorem 13 (Stability and generalisation of DP-CompGD). *Assume that the loss function f is convex and L-Lipschitz for every $\mathbf{w} \in \mathcal{C}, z \in \mathcal{Z}$. Then,*

1. *For $\beta_t = \frac{1}{t+1}$, $\eta_t = \frac{\eta}{T^{3/4}}$, and some absolute constant η and $T \asymp n^2$, the differentially private CompGD is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\left(\frac{L\log(n)}{\sqrt{n}}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}} + \frac{\log(1/\delta)}{n^3\sqrt{n}\epsilon^2}\sum_{t=1}^{n^2} m_t\right). \quad (15)$$

5.3 Differentially Private CompSGD with mini-batch

While we prefer using large batch gradients in the private setting to reduce the global sensitivity of the gradient and improve optimisation with fewer iterations, private mini-batch SGD can be useful for large sample sets. Moreover, the mini-batch act as a trade-off parameter between computational complexity and the accuracy of gradient updates. In this section, we present the differentially private CompSGD algorithm using mini-batch gradient updates, the algorithm is outlined in Alg. 5.

Note that the mini-batch version of DP-CompSGD induces an extra log factor in its variance σ^2, which will lead to an extra multiplicative log factor in the optimisation and generalisation bounds. This is a trade-off in privacy from using a smaller batch of samples in each gradient update (instead of the full batch as in Alg. 4). Fortunately, we are still able to obtain the same convergence guarantees for the mini-batch as in Section. 5.2 when ϵ tends to infinity.

Theorem 14 (Optimisation with DP-CompMiniBatch, non-smooth case). *Assume that the loss function f is convex and L-Lipschitz over the convex set $\mathcal{C}$. Suppose we*

Algorithm 5 Differentially Private CompSGD with mini-batch (DP-CompMiniBatch)

```
Inputs: Sample set $S = \{z_1, \ldots, z_n\}$, batch size $b$, privacy parameters $(\epsilon, \delta)$, step size parameters $\{\eta_t\}$ and projection parameters $\{\beta_t\}$.
initialize $\mathbf{w}_1$ as any point in $\mathcal{C}$
for $t = 1$ to $T$ do
    set $m_t = \min\{d, \omega(\mathcal{C})^2/\beta_t^2\}$
    choose projection matrix $\Phi_t \in \mathbb{R}^{m_t \times d}$ with i.i.d. entries from $\mathcal{N}(0, 1/m_t)$
    Sample a mini-batch $B_t$ of size $b$ uniformly from $S$
    set $\sigma^2 = \frac{16^2 LT \log(1/\delta) \log(2.5Tb/(\delta n))}{n^2 \epsilon^2}$
    set $s_t = \frac{\|\nabla F_{B_t}(\mathbf{w}_t)\|}{\|\Phi_t \nabla F_{B_t}(\mathbf{w}_t)\|}$
    set $\theta_t = \Pi_{\Phi_t \mathcal{C}}(\Phi_t \mathbf{w}_t - \eta_t(s_t \Phi_t \nabla F_{B_t}(\mathbf{w}_t) + \mathbf{e}))$ where $\mathbf{e} \sim \mathcal{N}(0, \sigma^2 I_{m_t})$
    pick $\mathbf{w}_{t+1}$ to be any element from the set $\{\mathbf{w} \in \mathcal{C} : \Phi_t \mathbf{w} = \theta_t\}$
end for
Output: $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$
```

run the compressed GD with step sizes $\eta_t = \|\mathcal{C}\|/\sqrt{t(L^2 + m_T\sigma^2)}$. For privacy parameters $0 \leq \epsilon, \delta \leq 1$ and $\beta_t = 1/(t+1)$. Then the private CompSGD with mini-batch satisfies

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{\log T \|\mathcal{C}\| L}{\sqrt{T}} + \frac{\log T \|\mathcal{C}\| L \sqrt{m_T \log(1/\delta) \log(4TB/(\delta n))}}{n\epsilon}\right).$$

Similar to the non-private case for the mini-batch variance, there is no improvement with the additional smoothness condition, we obtain the same convergence for both cases. The rate we obtained here is the same as the result obtained with non-projected gradients in [Bassily et al., 2020] with a key difference: the dimensionality dependence $\sqrt{d}$ is replaced with the maximum projection dimension m_T. Hence the reduction of dimensionality here comes for "free" compared with using non-projected gradients.

Theorem 15 (Stability and generalisation of DP-CompMiniBatch). *Assume that the loss function f is convex and L-Lipschitz for every $\mathbf{w} \in \mathcal{C}, z \in \mathcal{Z}$.*

1. *Then, for $\beta_t = \frac{1}{t+1}$, $\eta_t = \frac{\eta}{T^{3/4}}$ for some absolute constant η and $T \asymp n^2$, the differentially private mini-batch CompSGD is ϵ_{stab}-uniformly stable with $\epsilon_{stab} = \mathcal{O}\left(\frac{L\log(n)}{\sqrt{n}}\right)$.*
2. *Moreover, the weighted average output $\bar{\mathbf{w}}_T$ satisfies the following excess risk bound*

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] = F(\mathbf{w}^*) + \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2 + L^2)\log(n)}{\sqrt{n}} + \frac{L^2 \log(4Tb/(\delta n)) \log(1/\delta)}{n^3\sqrt{n}\epsilon^2} \sum_{t=1}^{n^2} m_t\right).$$

Remark 7. Note that the generalisation bound obtained here is tight, as we required at least $\mathcal{O}(1/\sqrt{n})$ even in the non-private case. The convergence rate will be almost the same (up to log factors) as the non-private case if the privacy parameter is not too small - $\epsilon \asymp 1/\sqrt{n}$.

6 Proofs

In this section, we present the proofs of the key theorems from Section 3 and Section 5. Supporting lemmas and proofs of intermediate results are deferred to the Appendix to focus on the proofs of our main results only. Similarly, proofs for Sections 4.1, 4.2 and 5.3 are deferred to the Appendix as they use similar techniques as the proofs we present in this section with slight variations.

For simplicity, we first denote our gradient update at iteration t as follows. A gradient update in CompSGD (lines 4-8 in Alg. 1) is a map $G : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ that takes a parameter vector $\mathbf{w}_t$ and a training point z as inputs, and it outputs the updated parameter vector $G(\mathbf{w}_t, z)$, defined as the following

$$G(\mathbf{w}_t, z) = \arg\min_{\mathbf{w}\in\mathcal{C}}\{\|\mathbf{w}\|_1 : \Phi_t\mathbf{w} = \Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}_t - \eta_t\Phi_t\nabla f(\mathbf{w}_t, z))\}, \tag{16}$$

where Φ_t is a RP matrix, and η_t is the step size parameter. We drop the dependence of z when it is clear from the context and just write $G(\mathbf{w})$ for simplicity. We remark that our analysis does not require $\mathbf{w}$ to have the minimum ℓ_1-norm property; this is included only to break ties so that the map G is well defined. Indeed, one can pick the updated element as described in Alg 1.

6.1 Proof of stability & generalisation of CompSGD under smoothness

This section will prove Theorem 2, i.e. the stability of CompSGD, which combined with an existing optimisation bound will give us its generalisation guarantee.

First, we establish some important properties of CompSGD. A key idea in stability analysis is to control the extent to which a sequence of updates starting from neighbouring sample sets diverge, in each iteration - in our case, one update corresponds to one run of lines 4-8 in Alg. 1. The algorithm is more stable if the divergence is smaller. The following result shows how the divergence between two gradient updates in CompSGD is controlled by the projection parameter β_t.

Lemma 1 (Distortion induce by the random projection). *Let $\mathbf{w}_{t+1}, \mathbf{w}'_{t+1} \in \mathcal{C}$ be the parameter vectors at iteration $t+1$ of Alg. 1 when run on two neighbouring sample sets S and S'. For any choices of training points z_{i_t} and z'_{i_t}, we have*

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 \leq \|(\mathbf{w}_t - \mathbf{w}'_t) - (\eta_t\nabla f(\mathbf{w}_t, z_{i_t}) - \eta_t\nabla f(\mathbf{w}'_t, z'_{i_t}))\|^2.$$

Since Lemma 1 implies that the divergence of gradient update with projected gradient is upper bounded by the divergence of regular gradient update (up to the $1-\beta_t$ factor), we can make use of some property analysis from classical SGD. We will approach this using the concepts of expansivity and boundedness introduced in [Hardt et al., 2016].

Definition 4 (Well behaved gradient update). *We say the gradient update $G(\mathbf{w})$ is α-expansive if, for all $\mathbf{v}, \mathbf{w} \in \mathcal{C}$ we have $\|G(\mathbf{v}) - G(\mathbf{w})\| \leq \alpha\|\mathbf{v} - \mathbf{w}\|$. We say the gradient update $G(\mathbf{w})$ is γ-bounded if $\sup_{\mathbf{w}\in\mathcal{C}} \|\mathbf{w} - G(\mathbf{w})\| \leq \gamma$.*

We now show that, for the type of loss functions considered, the update rule of Alg. 1 is well-behaved despite the distortion created by random compression. First we show that, at any iteration t the CompSGD update rule $\mathbf{w}_{t+1} = G(\mathbf{w}_t)$ has limited expansiveness whenever the same training point is chosen for gradient estimation (Lem. 2). Secondly, the CompSGD update is bounded whenever different training points are chosen for gradient estimation (Lem. 3).

Lemma 2 (Limited expansiveness). *Assume that f is convex and μ-smooth. Fix any $t \in \mathbb{N}$, and let $\mathbf{w}_t, \mathbf{w}'_t \in \mathcal{C}$ be the parameter vectors at the t-th iteration of Alg. 1 when run on two neighbouring sample sets S and S'. If $z_{i_t} = z'_{i_t}$ i.e. the same training point is chosen to estimate the gradient at the t-th iteration, then the update rule of CompSGD (Alg. 1) is $\frac{1}{\sqrt{1-\beta_t}}$-expansive for $\eta_t \leq 2/\mu$ – that is, we have $\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\| \leq \frac{1}{\sqrt{1-\beta_t}}\|\mathbf{w}_t - \mathbf{w}'_t\|$.*

Lemma 3 (Boundedness). *Assume that f is L-Lipschitz. Fix any $t \in \mathbb{N}$, and let $\mathbf{w}_t \in \mathcal{C}$ be the parameter vector at the t-th iteration of Alg. 1 when run on S. Then the update rule of the CompSGD (Alg. 1) is $\frac{\eta_t L}{\sqrt{1-\beta_t}}$-bounded – that is, we have $\|\mathbf{w}_{t+1} - \mathbf{w}_t\| \leq \frac{\eta_t L}{\sqrt{1-\beta_t}}$.*

The detailed proofs are these properties are deferred to Appendix A.1 and we focus on the proof of our main result here. With these core properties recorded, we now prove the stability guarantee of the CompSGD under the smoothness setting. The basic idea in the proof is that we note with probability $1 - 1/n$ the sample we select from S and S' will be identical, which allows us to use the expansiveness of CompSGD. We can then bound the low probability case with the γ-bounded property and put the two cases together to obtain our result.

Proof of Theorem 2. Let S and S' be two neighbouring sample sets of size n that differ in one single sample point. Denote $G_t := G(\cdot, z_{i_t})$ and $G'_t := G(\cdot, z'_{i_t})$, with $t \in [T], i_t \in [n]$, the gradient updates induced by running the CompSGD on the neighbouring sample sets

S and S', respectively. Let $\delta_T = \|\mathbf{w}_T - \mathbf{w}'_T\|$, and fix a sample point z. By the Lipschitz condition,

$$\mathbb{E}[|f(\mathbf{w}_T, z) - f(\mathbf{w}'_T, z)|] \leq L\mathbb{E}[\delta_T]. \tag{17}$$

Observe that, at iteration t, with probability $1 - 1/n$, the example z_{i_t} and z'_{i_t} selected from both S and S' is the same. In this case we have $G_t = G'_t$, and we use the limited expansiveness property of the update G_t from Lemma 2. With the remaining probability $1/n$, we have $z_{i_t} \neq z'_{i_t}$, in which case we use the boundedness property of both updates G_t and G'_t cf. Lemma 3. By the linearity of expectation, and the triangle inequality, this yields the following

$$\mathbb{E}[\delta_{t+1}] \leq \left(\frac{1 - 1/n}{\sqrt{1 - \beta_t}}\right) \mathbb{E}[\delta_t] + \frac{1}{n}\left(\mathbb{E}[\delta_t] + \frac{2\eta_t L}{\sqrt{1 - \beta_t}}\right). \tag{18}$$

Now it remains to solve this recursive sequence. We multiply both sides by $\prod_{j=1}^{t-1}\sqrt{1 - \beta_j}$,

$$\left[\prod_{j=1}^{t}\sqrt{1 - \beta_j}\right] \mathbb{E}[\delta_{t+1}] \leq \left[\prod_{j=1}^{t-1}\sqrt{1 - \beta_j}\right] \mathbb{E}[\delta_t] + \frac{2\eta_t L}{n}\prod_{j=1}^{t-1}\sqrt{1 - \beta_j} \tag{19}$$

and sum up the T iterates

$$\left[\prod_{j=1}^{T-1}\sqrt{1 - \beta_j}\right] \mathbb{E}[\delta_T] \leq \sum_{t=1}^{T-1}\frac{2\eta_t L}{n}\prod_{j=1}^{t-1}\sqrt{1 - \beta_j}. \tag{20}$$

Rearranging, we have:

$$\mathbb{E}[\delta_T] \leq \frac{2L}{n}\sum_{t=1}^{T-1}\eta_t\prod_{j=t}^{T-1}(1 - \beta_j)^{-1/2}. \tag{21}$$

In particular, with the choice $\beta_j = \frac{1}{j+1}$, we have $\prod_{j=t}^{T-1}(1 - \beta_j)^{-1/2} = \frac{\sqrt{T}}{\sqrt{t}}$. Furthermore, choosing $\eta_t = \frac{\eta}{\sqrt{t}}$ with some absolute constant η, we have

$$\mathbb{E}[\delta_T] = \frac{2\eta L\sqrt{T}}{n}\sum_{t=1}^{T-1}\frac{1}{t} = \mathcal{O}\left(\frac{L\sqrt{T}\log(T)}{n}\right), \tag{22}$$

where we exploited the fact that the growth rate of the partial sum of a harmonic series is just logarithmic. Finally, we take $T \asymp n$ and plug it back into (17) to conclude our stability bound. □

Note that this proves all results from Theorem 2. The generalisation bound in Theorem 2 is a direct consequence of stability, combined with the optimisation bound from [Kasiviswanathan, 2021, Thm. 2.3] and Theorem 1, using the strategy discussed in section 2.2.

6.2 Proof of optimisation, stability & generalisation of CompSGD without smoothness

We first prove the optimisation bound of CompSGD in the non-smooth case that uses a small step size parameter.

Theorem 16 (Optimisation with CompSGD with small step size). *Let f be a convex and L-Lipschitz function over a convex set $\mathcal{C}$. Then with $\eta_t = \frac{\eta}{T^{3/4}}$ and $\beta_t = \frac{1}{t+1}$, CompSGD satisfies*

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2 \log(T) + L^2}{T^{1/4}}\right). \tag{23}$$

Proof for Theorem 16. We apply Thm 19 with $\mathbf{w} = \mathbf{w}^*$ and have that for all $t \geq 1$:

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 &= \|\mathbf{w}_t - \mathbf{w}^*\|^2 + 2\eta_t\langle \nabla f(\mathbf{w}_t; z_{i_t}), \mathbf{w}^* - \mathbf{w}_t\rangle + \eta_t^2\|\nabla f(\mathbf{w}_t)\|^2 \\ &\leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 + 2\eta_t(f(\mathbf{w}^*; z_{i_t}) - f(\mathbf{w}_t; z_{i_t})) + \eta_t^2 L^2. \end{aligned} \tag{24}$$

Rearranging we have:

$$2\eta_t(f(\mathbf{w}_t; z_{i_t}) - f(\mathbf{w}^*; z_{i_t})) \leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 + \beta_t\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 + \eta_t^2 L^2. \tag{25}$$

We take expectations on both sides, and sum over the T iterates,

$$2\sum_{t=1}^{T} \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)] \leq \|\mathbf{w}_1 - \mathbf{w}^*\|^2 + \sum_{t=1}^{T} \beta_t \mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] + L^2 \sum_{t=1}^{T} \eta_t^2.$$

By choosing $\beta_t = 1/(t+1)$ and using the bound $\|\mathbf{w}_t - \mathbf{w}^*\|^2 \le \|\mathcal{C}\|^2$, we obtain

$$2\sum_{t=1}^{T} \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\|\mathcal{C}\|^2 + \|\mathcal{C}\|^2 \sum_{t=1}^{T} \frac{1}{t+1} + L^2 \sum_{t=1}^{T} \eta_t^2 \right)$$
$$= \mathcal{O}\left(\|\mathcal{C}\|^2 \log(T) + L^2 \sum_{t=1}^{T} \eta_t^2 \right). \tag{26}$$

Finally, choosing $\eta_t = \frac{\eta}{T^{3/4}}$ we obtain our result as

$$\left(\sum_{t=1}^{T} \eta_t\right)^{-1} \sum_{t=1}^{T} \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2 \log(T)}{T^{1/4}} + \frac{L^2}{T^{3/4}} \right). \tag{27}$$

□

One can find from the proof of the optimisation bound that we can use the parameters $\eta = \mathcal{O}(1/\sqrt{t})$ and $T \asymp n$ to yield convergence of $\mathcal{O}(1/\sqrt{n})$. However, we need to balance stability and optimisation so that we obtain the best generalisation convergence overall, hence we choose a smaller learning rate.

Proof of Theorem 3. In the non-smooth setting, we no longer have all the properties of CompSGD proved for the smooth case. However we still have the core result (Lemma 1) and note that the probability that we pick a different sample at an iteration is $1/n$ as in the smooth case. For the case where the selected sample is identical, we can make use of the convexity of f and obtain the same convergence rate by carefully choosing the learning rate η_t.

Let S and S' be two neighbouring sample sets of size n that differ in one single sample. Let $G(\mathbf{w}_t) = \mathbf{w}_{t+1}$ denote the gradient update and let $G_1, \ldots, G_T$ and $G'_1, \ldots, G'_T$ be the updates induced by running the CompSGD on S and S' for T iterates, respectively. Let $\delta_T = \|\mathbf{w}_T - \mathbf{w}'_T\|$, by the Lipschitz condition,

$$\mathbb{E}[|f(\mathbf{w}_T, z) - f(\mathbf{w}'_T, z)|] \le L\mathbb{E}[\delta_T]. \tag{28}$$

If at iteration t, the sample we selected is the same $G_t = G'_t$, then from Lemma 1 we have the following (short notations $\nabla f(\mathbf{w}_t, z_{i_t}) = \nabla f(\mathbf{w}_t)$ for simplicity)

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 \le \|(\mathbf{w}_t - \mathbf{w}'_t) - \eta_t(\nabla f(\mathbf{w}_t, z_{i_t}) - \nabla f(\mathbf{w}'_t, z_{i_t}))\|^2$$
$$= \|\mathbf{w}_t - \mathbf{w}'_t\|^2 - 2\eta_t\langle \nabla f(\mathbf{w}_t) - \nabla f(\mathbf{w}'_t), \mathbf{w}_t - \mathbf{w}'_t\rangle + \eta_t^2\|\nabla f(\mathbf{w}_t) - \nabla f(\mathbf{w}'_t)\|^2.$$

From the convexity of f we have that $\langle \nabla f(\mathbf{w}_t) - \nabla f(\mathbf{w}'_t), \mathbf{w}_t - \mathbf{w}'_t \rangle \geq 0$ and from Lipschitzness of f we also have $\|\nabla f(\mathbf{w}_t) - \nabla f(\mathbf{w}'_t)\| \leq 2L$. Hence we obtain the following bound

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 \leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + 4L^2\eta_t^2. \tag{29}$$

For the case where $G_t \neq G'_t$, we use the inequality $(a+b)^2 \leq (1+p)a^2 + (1+1/p)b^2$ to obtain

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 &\leq \|(\mathbf{w}_t - \mathbf{w}'_t) - \eta_t(\nabla f(\mathbf{w}_t, z_{i_t}) - \nabla f(\mathbf{w}'_t, z_{i'_t}))\|^2 \\ &\leq (1+p)\|\mathbf{w}_t - \mathbf{w}'_t\|^2 + 4(1+1/p)L^2\eta_t^2,\end{aligned}$$

where we have again used the Lipschitz condition of f. Combining the two cases we have

$$\begin{aligned}(1-\beta_t)\mathbb{E}[\delta_{t+1}^2] &\leq \left(1 - \frac{1}{n}\right)(\mathbb{E}[\delta_t^2] + 4L^2\eta_t^2) + \frac{1}{n}\left((1+p)\mathbb{E}[\delta_t^2] + 4(1+1/p)L^2\eta_t^2\right) \\ &\leq (1+p/n)\mathbb{E}[\delta_t^2] + \left(4 + \frac{4(1+1/p)}{n}\right)L^2\eta_t^2. \end{aligned} \tag{30}$$

Denoting $\varDelta_t = \left[\prod_{j=1}^{t-1}(1-\beta_j)\right](1+p/n)^{-t} \cdot \mathbb{E}[\delta_t^2]$ and multiplying both sides by $\left[\prod_{j=1}^{t-1}(1-\beta_j)\right](1+p/n)^{-(t+1)}$, we obtain

$$\varDelta_{t+1} \leq \varDelta_t + \left[\prod_{j=1}^{t-1}(1-\beta_j)\right](1+p/n)^{-(t+1)}\left(4 + \frac{4(1+1/p)}{n}\right)L^2\eta_t^2. \tag{31}$$

Choosing $p = n/T$ and summing over T iterates we have

$$\varDelta_T \leq \sum_{t=1}^{T-1}\left[\prod_{j=1}^{t-1}(1-\beta_j)\right](1+1/T)^{-(t+1)}\left(4 + \frac{4(1+T/n)}{n}\right)L^2\eta_t^2. \tag{32}$$

By rearranging and choosing $\beta_j = \frac{1}{j+1}$, we get $\prod_{j=1}^{t-1}(1-\beta_j) = 1/t$. Hence we have

$$\mathbb{E}[\delta_T^2] \leq \sum_{t=1}^{T-1}\frac{T}{t}\left(1+\frac{1}{T}\right)^{T-(t+1)}\left(4+\frac{4(1+T/n)}{n}\right)L^2\eta_t^2 = \mathcal{O}\left(L^2T\sum_{t=1}^{T-1}\frac{(1+T/n^2)}{t}\eta_t^2\right), \tag{33}$$

where we noted that the factor $(1+1/T)^{T-(t+1)} = 1 + \mathcal{O}((T-(t+1)/T) = \mathcal{O}(1)$.

Using Eq. (28) and (33), letting $\eta_t = \frac{\eta}{T^{3/4}}$ for some absolute constant η and $T \asymp n^2$, we obtain our result by taking the square-root from both sides

$$\mathbb{E}[|f(\mathbf{w}_T, z) - f(\mathbf{w}'_T, z)|] = \mathcal{O}\left(\frac{L^2\sqrt{\log(n)}}{\sqrt{n}}\right). \tag{34}$$

The generalisation result then follows as a direct consequence by combining it with Thm 16 and applying Thm. 1. □

6.3 Proofs for DP-CompGD

Notation note: for simplicity of notation, we denote $\nabla F_S(\mathbf{w}_t)$ by ∇F_t and $\nabla F_{S'}(\mathbf{w}'_t)$ by $\nabla F'_t$ in this section.

Before proving the utility guarantee of the algorithm, we require the following lemma as part of our proof. We will include the proof of this lemma in here due to its importance to our main result.

Lemma 4. *For $\mathbf{w}_t$ in DP CompSGD algorithm, we have for all t,*

$$\mathbb{E}[\langle \Phi_t(\mathbf{w} - \mathbf{w}_t), s_t\Phi_t\nabla f(\mathbf{w}_t)\rangle] = \langle \mathbf{w} - \mathbf{w}_t, \nabla f(\mathbf{w}_t)\rangle \cdot C_{m_t},$$

where $C_{m_t} = \sqrt{\frac{2}{m_t}}\frac{\Gamma((m_t+1)/2)}{\Gamma(m_t/2)} \in \left[\sqrt{\frac{m_t}{m_t+1}}, 1\right]$, and $\Gamma(\cdot)$ is the gamma function.

Lemma 4 shows that the inner product between the projected weight vector and the normalized projected gradient vector is almost identical to their inner product before projection. Furthermore, lemma 4 also implies that the sign of the inner product is preserved under Gaussian random projection in expectation. This lemma is a key observation that allows us to prove a similar convergence result with projected gradients. We also state the definition of Chi-distribution for the completeness of the proof:

Definition 5 (Chi-distribution). *The probability density function of chi-distribution is*

$$f(x;k) = \begin{cases} \frac{x^{k-1}e^{-x^2/2}}{2^{k/2-1}\Gamma\left(\frac{k}{2}\right)}, & \textit{if } x \geq 0, \\ 0, & \textit{otherwise}, \end{cases} \tag{35}$$

where $\Gamma(z)$ is the gamma function. It is known that the expected value of the chi-distribution is $\frac{\sqrt{2}\Gamma((k+1)/2)}{\Gamma(k/2)}$.

Proof of Lemma 4. Note that $\mathbf{e}$ is independent from the rest of parameters and $\mathbb{E}[\mathbf{e}] = 0$. Denote $\nabla F_S(\mathbf{w}_t)$ by ∇F_t, we have

$$\begin{aligned}\mathbb{E}_{\Phi_t,\mathbf{e}}[\langle \Phi_t(\mathbf{w}-\mathbf{w}_t), s_t\Phi_t\nabla F_t + \mathbf{e}\rangle] &= \mathbb{E}_{\Phi_t,\mathbf{e}}[\langle \Phi_t(\mathbf{w}-\mathbf{w}_t), s_t\Phi_t\nabla F_t\rangle + \langle \Phi_t\mathbf{w} - \Phi\mathbf{w}_t, \mathbf{e}\rangle] \\ &= \mathbb{E}_{\Phi_t}[\langle \Phi_t(\mathbf{w}-\mathbf{w}_t), s_t\Phi_t\nabla F_t\rangle]. \end{aligned} \tag{36}$$

For simplicity let us denote $(\mathbf{w}-\mathbf{w}_t)$ by $\mathbf{v}$. To bound the above quantity, we first consider two special cases of ∇F_t: 1. ∇F_t is a scale multiple of $\mathbf{v}$; 2. ∇F_t is perpendicular to $\mathbf{v}$. For the first case, $\nabla F_t = c\mathbf{v}$ for some constant c. We have

$$\begin{aligned}\|\nabla F_t\|\mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t\mathbf{v}, \frac{\Phi_t\nabla F_t}{\|\Phi_t\nabla F_t\|}\right\rangle\right] &= sign(c)\|c\mathbf{v}\|\mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t\mathbf{v}, \frac{\Phi_t\mathbf{v}}{\|\Phi_t\mathbf{v}\|}\right\rangle\right] \\ &= sign(c)\|c\mathbf{v}\|\mathbb{E}_{\Phi_t}\left[\|\Phi_t\mathbf{v}\|\right] \\ &= sign(c)\frac{|c|\|\mathbf{v}\|^2}{\sqrt{m_t}}\mathbb{E}_{\Phi_t}\left[\frac{\|\Phi_t\mathbf{v}\|\sqrt{m_t}}{\|\mathbf{v}\|}\right]. \end{aligned} \tag{37}$$

Since Φ_t's entries are randomly drawn from distribution $\mathcal{N}(0, 1/m_t)$, this implies that $\Phi_t(\mathbf{v}/\|\mathbf{v}\|)\sqrt{m_t} \sim \mathcal{N}(0, I_{m_t})$. Hence the norm $\|\Phi_t(\mathbf{v}/\|\mathbf{v}\|)\sqrt{m_t}\|$ is Chi-distributed with m_t degrees of freedom. The expectation of a Chi-distributed random variable is

$$\mathbb{E}_{\Phi_t}\left[\frac{\|\Phi_t\mathbf{v}\|\sqrt{m_t}}{\|\mathbf{v}\|}\right] = \frac{\sqrt{2}\Gamma((m_t+1)/2)}{\Gamma(m_t/2)}.$$

With $C_{m_t} = \frac{\sqrt{2}\Gamma((m_t+1)/2)}{\Gamma(m_t/2)\sqrt{m_t}}$ we get from equation (37) that

$$\|\nabla F_t\|\mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t\mathbf{v}, \frac{\Phi_t\nabla F_t}{\|\Phi_t\nabla F_t\|}\right\rangle\right] = C_{m_t}c\|\mathbf{v}\|^2. \tag{38}$$

The second special case of interest is when ∇F_t is perpendicular to $\mathbf{v}$. Note that this condition implies that $\Phi_t\nabla F_t$ is independent to $\Phi_t\mathbf{v}$. Indeed, if we consider their covariance:

$$\mathrm{cov}_{\Phi_t}(\Phi_t\mathbf{v}, \Phi_t\nabla F_t) = \mathbb{E}_{\Phi_t}[\langle \Phi_t\mathbf{v}, \Phi_t\nabla F_t\rangle] = \langle \mathbf{v}, \nabla F_t\rangle, \tag{39}$$

which equals to zero when $\mathbf{v}$ is perpendicular to ∇F_t. Hence we have

$$\|\nabla F_t\|\mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t\mathbf{v}, \frac{\Phi_t\nabla F_t}{\|\Phi_t\nabla F_t\|}\right\rangle\right] = \|\nabla F_t\|\langle \mathbb{E}_{\Phi_t}\left[\Phi_t\mathbf{v}\right], \mathbb{E}_{\Phi_t}\left[\Phi_t\nabla F_t\right]\rangle = 0. \tag{40}$$

Now for any vector $\mathbf{v}$, we can write $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$ where $\mathbf{v}_1$ is a vector perpendicular to ∇F_t and $\mathbf{v}_2$ is a scalar multiple of ∇F_t. In this case we can split-up the inner product as follows

$$\begin{aligned} &\|\nabla F_t\| \mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t \mathbf{v}, \frac{\Phi_t \nabla F_t}{\|\Phi_t \nabla F_t\|}\right\rangle\right] \\ &\qquad = \|\nabla F_t\| \left(\mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t \mathbf{v}_1, \frac{\Phi_t \nabla F_t}{\|\Phi_t \nabla F_t\|}\right\rangle\right] + \mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t \mathbf{v}_2, \frac{\Phi_t \nabla F_t}{\|\Phi_t \nabla F_t\|}\right\rangle\right]\right). \qquad (41) \end{aligned}$$

Now, using the properties of $\mathbf{v}_1$ and $\mathbf{v}_2$, we have

$$\begin{aligned} eq.\ (41) &= \|\nabla F_t\| \mathbb{E}_{\Phi_t}\left[\left\langle \Phi_t \mathbf{v}_2, \frac{\Phi_t \nabla F_t}{\|\Phi_t \nabla F_t\|}\right\rangle\right] \\ &= \|\nabla F_t\| \left\langle \mathbf{v}_2, \frac{\nabla F_t}{\|\nabla F_t\|}\right\rangle \cdot C_{m_t} \qquad (42) \\ &= \|\nabla F_t\| \left\langle \mathbf{v}, \frac{\nabla F_t}{\|\nabla F_t\|}\right\rangle \cdot C_{m_t}, \qquad (43) \end{aligned}$$

where (42) used that $\mathbf{v}_2$ is a scalar multiple of ∇F_t with scalar multiple $c = \pm 1$ being sufficient to consider (since otherwise we can divide and multiply with $\|\mathbf{v}_2\|$), and the last equality holds because $\mathbf{v}_1$ is perpendicular to ∇F_t. Hence we have for all $\mathbf{v} = \mathbf{w} - \mathbf{w}_t$ that

$$\mathbb{E}_{\Phi_t}[\langle \Phi_t(\mathbf{w} - \mathbf{w}_t), s_t \Phi_t \nabla F_t\rangle] = \langle (\mathbf{w} - \mathbf{w}_t), \nabla F_t\rangle \cdot C_{m_t}. \qquad (44)$$

□

Using Lemma 4, we now show the following optimisation bound for the differentially private compressed gradient descent.

Proof of Theorem 11. Since after each update, $\mathbf{w}_t \in \mathcal{C}$ for all t, we can recall equation (74) in the proof of Thm. 19, we have

$$(1 - \beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}\|^2 \leq \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_{t+1} - \mathbf{w})\|^2\right]. \qquad (45)$$

Hence, by expanding out the RHS, we have (here we will denote $\nabla F_S(\mathbf{w}_t)$ by ∇F_t)

$$\begin{aligned}
(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}\|^2 &\le \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}_t-\eta_t(s_t\Phi_t\nabla F_t+\mathbf{e}))-\Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w})\|^2\right]\\
&\le \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Phi_t\mathbf{w}_t-\eta_t(s_t\Phi_t\nabla F_t+\mathbf{e})-\Phi_t\mathbf{w}\|^2\right]\\
&= \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Phi_t\mathbf{w}_t-\Phi_t\mathbf{w}\|^2\right]+2\eta_t\mathbb{E}_{\Phi_t,\mathbf{e}}[\langle\Phi_t\mathbf{w}-\Phi_t\mathbf{w}_t,s_t\Phi_t\nabla F_t+\mathbf{e}\rangle]\\
&\quad+\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|s_t\Phi_t\nabla F_t+\mathbf{e}\|^2]\\
&= \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t\langle\mathbf{w}-\mathbf{w}_t,\nabla F_t\rangle+\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|s_t\Phi_t\nabla F_t\|^2]\\
&\quad+2\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\langle s_t\Phi_t\nabla F_t,\mathbf{e}\rangle]+\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|\mathbf{e}\|^2]\\
&= \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t\langle\mathbf{w}-\mathbf{w}_t,\nabla F_t\rangle+\eta_t^2(L^2+m_t\sigma^2)\\
&\le \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t(F_S(\mathbf{w})-F_S(\mathbf{w}_t))+\eta_t^2(L^2+m_t\sigma^2), \qquad (46)
\end{aligned}$$

where we have used Lemma 4 between the fourth and fifth line and the convexity of f in the last step. Also note that since $\mathbf{e}$ is i.i.d. fresh Gaussian noise, the expectation of $\mathbf{e}$ is 0, hence the expected inner product with $\mathbf{e}$ is also zero. Rearranging the last inequality and let $\mathbf{w}=\mathbf{w}^*$ we have:

$$2C_{m_t}\eta_t(F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*))\le\|\mathbf{w}_t-\mathbf{w}^*\|^2-(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2+\eta_t^2(L^2+m_t\sigma^2). \quad (47)$$

Taking expectation and summing over T iterations we have

$$2C_{m_t}\sum_{t=1}^T\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)]\le\|\mathbf{w}_1-\mathbf{w}^*\|^2+\sum_{t=1}^T\beta_t\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2]+\sum_{t=1}^T\eta_t^2(L^2+m_t\sigma^2).$$

Choosing $\beta_t=1/(t+1)$ we obtain that

$$\frac{\sum_{t=1}^T\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)]}{\sum_{t=1}^T\eta_t}=\mathcal{O}\left(\frac{\|\mathcal{C}\|^2+\log T\|\mathcal{C}\|^2+(L^2+m_T\sigma^2)\sum_{t=1}^T\eta_t^2}{\sum_{t=1}^T\eta_t}\right), \quad (48)$$

where we have used $m_T=\max_{t\in[T]}m_t$.

Finally, let $\eta_t=\|\mathcal{C}\|/\sqrt{t(L^2+m_T\sigma^2)}$ and $\sigma^2=\mathcal{O}(TL^2\log(1/\delta)/(\epsilon^2n^2))$ we have

$$\begin{aligned}
\frac{\sum_{t=1}^T\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)]}{\sum_{t=1}^T\eta_t}&=\mathcal{O}\left(\frac{\log T\|\mathcal{C}\|\sqrt{L^2+m_T\sigma^2}}{\sqrt{T}}\right)\\
&\le\mathcal{O}\left(\frac{\log T\|\mathcal{C}\|L}{\sqrt{T}}+\frac{\log T\|\mathcal{C}\|L\sqrt{m_TT\log(1/\delta)}}{n\epsilon\sqrt{T}}\right)\\
&=\mathcal{O}\left(\frac{\log T\|\mathcal{C}\|L}{\sqrt{T}}+\frac{\log T\|\mathcal{C}\|L\sqrt{m_T\log(1/\delta)}}{n\epsilon}\right). \qquad (49)
\end{aligned}$$

□

Proof of Theorem 12. Since after each update, $\mathbf{w}_t \in \mathcal{C}$ for all t, by equation (76) in the proof of lemma 1 (replacing $\mathbf{w}'_{t+1}$ with $\mathbf{w}$ and taking expectation w.r.t. $\mathbf{e}$) we have the following

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}\|^2 &\leq \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Phi_t(\mathbf{w}_{t+1}-\mathbf{w})\|^2\right] \\ &= \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}_t-\eta_t(s_t\Phi_t\nabla F_t+\mathbf{e}))-\Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w})\|^2\right] \\ &\leq \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Phi_t\mathbf{w}_t-\eta_t(s_t\Phi_t\nabla F_t+\mathbf{e})-\Phi_t\mathbf{w}\|^2\right] \\ &= \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Phi_t\mathbf{w}_t-\Phi_t\mathbf{w}\|^2\right]+\mathbb{E}_{\Phi_t,\mathbf{e}}[\langle\Phi_t\mathbf{w}-\Phi_t\mathbf{w}_t, s_t\Phi_t\nabla F_t+\mathbf{e}\rangle] \\ &\qquad + \eta^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|s_t\Phi_t\nabla F_t+\mathbf{e}\|^2].\end{aligned}\tag{50}$$

Hence using Lemma 4 and expanding out the last term, we have

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}\|^2 &= \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t\langle\mathbf{w}-\mathbf{w}_t,\nabla F_t\rangle+\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|s_t\Phi_t\nabla F_t\|^2] \\ &\qquad +2\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\langle s_t\Phi_t\nabla F_t,\mathbf{e}\rangle]+\eta_t^2\mathbb{E}_{\Phi_t,\mathbf{e}}[\|\mathbf{e}\|^2] \\ &= \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t\langle\mathbf{w}-\mathbf{w}_t,\nabla F_t\rangle+\eta_t^2(\|\nabla F_t\|^2+m_t\sigma^2) \\ &\leq \|\mathbf{w}_t-\mathbf{w}\|^2+2C_{m_t}\eta_t(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+\eta_t^2(\|\nabla F_t\|^2+m_t\sigma^2),\end{aligned}\tag{51}$$

where we have used the convexity of f in the second to last step. Also note that since $\mathbf{e}$ is i.i.d. fresh Gaussian noise, the expectation of $\mathbf{e}$ is 0, hence the expected inner product with $\mathbf{e}$ is also zero. Now we substitute $\mathbf{w}=\mathbf{w}_S^*$. Since $\mathbf{w}_S^*$ is an minimiser we have $\nabla F_S(\mathbf{w}_S^*)=0$. Hence by smoothness, we have

$$\|\nabla F_S(\mathbf{w}_t)\|^2=\|\nabla F_S(\mathbf{w}_t)-\nabla F_S(\mathbf{w}_S^*)\|^2\leq 2\mu(F_S(\mathbf{w}_t)-F_S(\mathbf{w}_S^*)).\tag{52}$$

Substituting equation (52) into (51) we have

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}_S^*\|^2 &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+(\eta_t-2\eta_t^2\mu)(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+\eta_t^2m_t\sigma^2 \\ \text{(assuming } \eta_t\leq 1/(4\mu)\text{)} \quad &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+\frac{\eta_t}{2}(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+\eta_t^2m_t\sigma^2.\end{aligned}\tag{53}$$

Rearranging we have:

$$\frac{\eta_t}{2}(F_S(\mathbf{w}_t) - F_S(\mathbf{w}_S^*)) \le \|\mathbf{w}_t - \mathbf{w}_S^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|^2 + \beta_t\|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|^2 + \eta_t^2 m_t\sigma^2.$$

Taking expectation and summing over T iterates and choosing $\beta_t = 1/(t+1)$ we have:

$$\begin{aligned}\sum_{t=1}^T \frac{\eta_t}{2}\mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}_S^*)] &\le \|\mathbf{w}_1 - \mathbf{w}_S^*\|^2 + \sum_{t=1}^T \beta_t \mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|^2] + \sum_{t=1}^T \eta_t^2 m_t\sigma^2 \\ &= \mathcal{O}\left(\|\mathcal{C}\| + \|\mathcal{C}\|^2 \sum_{t=1}^T \beta_t + \sum_{t=1}^T \eta_t^2 m_t \sigma^2\right) \\ &= \mathcal{O}\left(\|\mathcal{C}\| + \|\mathcal{C}\|^2 \log(T) + \frac{\log(1/\delta)T}{n^2\epsilon^2}\sum_{t=1}^T \eta_t^2 m_t\right), \qquad (54)\end{aligned}$$

where we have used $\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|]^2 \le \|\mathcal{C}\|^2$.

Finally, taking $\eta_t = \frac{\|\mathcal{C}\|}{L\sqrt{m_T}}$ where m_T is the maximum projection dimension, we have our final result

$$\left(\sum_{t=1}^T \eta_t\right)^{-1} \sum_{t=1}^T \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}_S^*)] = \mathcal{O}\left(\frac{L\|\mathcal{C}\| \log T\sqrt{m_T}}{T} + \frac{LT\sqrt{m_T}\log(1/\delta)}{n^2\epsilon^2}\right).$$

The proof is completed. □

We now turn to the stability analysis of CompGD in the private setting. The analysis begins similarly to the non-private setting, however DP introduces new challenges due to the normalisation factor s_t we introduced to keep the projected gradient bounded uniformly.

Proof of Theorem 13. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. W.l.o.g. assume that they differ on the j-th point denoted z_j, z_j' for S and S', respectively. Fix a sample z, by the Lipschitz condition we get

$$\mathbb{E}[|f(\mathbf{w}_T) - f(\mathbf{w}_T')|] \le L\mathbb{E}[\delta_T], \qquad (55)$$

where $\delta_T = \|\mathbf{w}_T - \mathbf{w}_T'\|$.

Since after each update, $\mathbf{w}_t \in \mathcal{C}$ for all t, by equation (76) in the proof of lemma 1 we have

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}_{t+1}'\|^2 \le \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_{t+1} - \mathbf{w}_{t+1}')\|^2\right]. \qquad (56)$$

Hence by expanding out the RHS, we have (here we will denote $\nabla F_S(\mathbf{w}_t)$ by ∇F_t and $\nabla F_{S'}(\mathbf{w}'_t)$ by $\nabla F'_t$)

$$\begin{aligned}
(56) &= \mathbb{E}_{\Phi_t,\mathbf{e}}\left[\|\Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}_t - \eta_t(s_t\Phi_t\nabla F_t + \mathbf{e})) - \Pi_{\Phi_t\mathcal{C}}(\Phi_t\mathbf{w}'_t - \eta_t(s'_t\Phi_t\nabla F'_t + \mathbf{e}))\|^2\right] \\
&\leq \mathbb{E}_{\Phi_t}\left[\|\Phi_t\mathbf{w}_t - \eta_t(s_t\Phi_t\nabla F_t) - (\Phi_t\mathbf{w}'_t - \eta_t(s'_t\Phi_t\nabla F'_t))\|^2\right] \\
&= \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_t - \mathbf{w}'_t)\|^2\right] + 2\eta_t\mathbb{E}_{\Phi_t}[\langle\Phi_t(\mathbf{w}'_t - \mathbf{w}_t), \Phi_t(s_t\nabla F_t - s'_t\nabla F'_t)\rangle] \\
&\quad + \eta_t^2\mathbb{E}_{\Phi_t}[\|\Phi_t(s_t\nabla F_t - s'_t\nabla F'_t)\|^2] \\
&= \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_t - \mathbf{w}'_t)\|^2\right] + 2\eta_t\mathbb{E}_{\Phi_t}[\langle\Phi_t(\mathbf{w}'_t - \mathbf{w}_t), \Phi_t s_t\nabla F_t\rangle - \langle\Phi_t(\mathbf{w}'_t - \mathbf{w}_t), \Phi_t s'_t\nabla F'_t\rangle] \\
&\quad + \eta_t^2\mathbb{E}_{\Phi_t}[\|\Phi_t(s_t\nabla F_t - s'_t\nabla F'_t)\|^2] \\
&= \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + 2C_{m_t}\eta_t\langle\mathbf{w}'_t - \mathbf{w}_t, \nabla F_t - \nabla F'_t\rangle + \eta_t^2\mathbb{E}_{\Phi_t}[\|\Phi_t(s_t\nabla F_t - s'_t\nabla F'_t)\|^2] \quad (57) \\
&= \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + 2C_{m_t}\eta_t\langle\mathbf{w}'_t - \mathbf{w}_t, \nabla F_t - \nabla F'_t\rangle + 4\eta_t^2L^2, \quad (58)
\end{aligned}$$

where the second-to-last line (57) follows by applying lemma 4 twice, the last line (58) follows from the Lipschitz assumption.

Since S, S' only differs on the j-th point, we have (C_{m_t} omitted here since $C_{m_t} \leq 1$.)

$$\begin{aligned}
\langle\mathbf{w}'_t - \mathbf{w}_t, \nabla F_t - \nabla F'_t\rangle &= \langle\mathbf{w}'_t - \mathbf{w}_t, \nabla F_{S\cup\{z'_j\}}(\mathbf{w}_t) - \nabla F_{S'\cup\{z_j\}}(\mathbf{w}'_t)\rangle \\
&\quad + \frac{1}{n}\langle\mathbf{w}'_t - \mathbf{w}_t, \nabla f(\mathbf{w}'_t, z_j) - \nabla f(\mathbf{w}_t, z'_j)\rangle \\
&\leq \frac{1}{n}\langle\mathbf{w}'_t - \mathbf{w}_t, \nabla f(\mathbf{w}'_t, z_j) - \nabla f(\mathbf{w}_t, z'_j)\rangle \\
&\leq \frac{2L}{n}\|\mathbf{w}_t - \mathbf{w}'_t\|, \quad (59)
\end{aligned}$$

where the second line holds because the first term is negative by the convexity of F_S, and the last line follows from the Lipschitz condition.

We substitute the inequality (59) into equation (58) and multiply both sides by $\prod_{j=1}^{t-1}(1-\beta_j)$. It then follows that

$$\left[\prod_{j=1}^{t}(1-\beta_j)\right]\delta_{t+1}^2 \leq \left[\prod_{j=1}^{t-1}(1-\beta_j)\right]\delta_t^2 + \frac{4\eta_t L\delta_t}{n}\prod_{j=1}^{t-1}(1-\beta_j) + 4\eta_t^2L^2\left[\prod_{j=1}^{t-1}(1-\beta_j)\right]. \quad (60)$$

By summing over T iterates we have:

$$\left[\prod_{j=1}^{T-1}(1-\beta_j)\right]\delta_T^2 \leq \sum_{t=1}^{T-1}\frac{4\eta_t L\delta_t}{n}\prod_{j=1}^{t-1}(1-\beta_j) + \sum_{t=1}^{T-1}4\eta_t^2L^2\left[\prod_{j=1}^{t-1}(1-\beta_j)\right]. \quad (61)$$

Taking $\beta_t = 1/t + 1$ and rearranging we have:

$$\delta_T^2 \leq \frac{4LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t \delta_t}{t} + 4L^2 T \sum_{t=1}^{T-1} \frac{\eta_t^2}{t}. \tag{62}$$

Claim: The following inequality holds for all T:

$$\delta_T \leq 2L\sqrt{T} \sqrt{\sum_{t=1}^{T-1} \frac{\eta_t^2}{t}} + \left(\frac{2LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} \right). \tag{63}$$

We prove this claim by induction: The base case $T = 0$ clearly holds as the right-hand side is always positive. For the inductive step, if $\delta_T \leq \max_{t \in [T]} \delta_t$, then by the inductive hypothesis we have

$$\delta_T \leq \delta_{T-1} \leq 2L\sqrt{T} \sqrt{\sum_{t=1}^{T-2} \frac{\eta_t^2}{t}} + \left(\frac{2LT}{n} \sum_{t=1}^{T-2} \frac{\eta_t}{t} \right) \leq 2L\sqrt{T} \sqrt{\sum_{t=1}^{T-1} \frac{\eta_t^2}{t}} + \left(\frac{2LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} \right). \tag{64}$$

For the other case where $\delta_T > \max_{t \in [T]} \delta_t$, we have from (62):

$$\delta_T^2 \leq \frac{4LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t \delta_t}{t} + 4L^2 T \sum_{t=1}^{T-1} \frac{\eta_t^2}{t} \leq \frac{4LT\delta_T}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} + 4L^2 T \sum_{t=1}^{T-1} \frac{\eta_t^2}{t}. \tag{65}$$

Which after rearranging is equivalent to:

$$\left(\delta_T - \frac{2LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} \right)^2 \leq \left(\frac{2LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} \right)^2 + 4L^2 T \sum_{t=1}^{T-1} \frac{\eta_t^2}{t}. \tag{66}$$

Taking square roots from both sides and the result follows from the sub-additivity of square roots. The inductive step is complete.

Finally using the choice $\eta_t = \mathcal{O}(1/T^{3/4})$ and $T \asymp n^2$ together with our proved claim, we have:

$$\begin{aligned} \mathbb{E}[\delta_T] &\leq 2L\sqrt{T} \sqrt{\sum_{t=1}^{T-1} \frac{\eta_t^2}{t}} + \left(\frac{2LT}{n} \sum_{t=1}^{T-1} \frac{\eta_t}{t} \right) \\ &= \mathcal{O}\left(\frac{L\sqrt{\log T}}{T^{1/4}} + \frac{LT^{1/4} \log T}{n} \right) = \mathcal{O}\left(\frac{L \log n}{\sqrt{n}} \right). \end{aligned}$$

For the excess risk bound, we have from Thm. 11

$$\frac{\sum_{t=1}^{T} \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)]}{\sum_{t=1}^{T} \eta_t} = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2 + \log T \|\mathcal{C}\|^2 + \sum_{t=1}^{T} \eta_t^2 (L^2 + m_t \sigma^2)}{\sum_{t=1}^{T} \eta_t} \right). \tag{67}$$

Using the choice of $\eta_t = \eta/T^{3/4}$ and $T \asymp n^2$ we have

$$\frac{\sum_{t=1}^{T} \eta_t \mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)]}{\sum_{t=1}^{T} \eta_t} = \mathcal{O}\left(\frac{\log n(\|\mathcal{C}\|^2 + L^2)}{\sqrt{n}} + \frac{\log(1/\delta)\sum_{t=1}^{n^2} m_t}{\sqrt{n}n^3\epsilon^2}\right). \tag{68}$$

Combining with the stability bound we obtain our final result. □

7 Conclusions

We presented a rigorous analysis of the stability and generalisation guarantee of SGD with compressed gradients. Our result shows that we can obtain almost optimal generalisation convergence with compressed gradients in both smooth and non-smooth cases. We also extend the analysis to the batch and mini-batch variants of CompSGD, and showed that the same convergence can be achieved with these variants. In particular, the batch variant achieves significantly better convergence rates compared to CompSGD with a constant step size. Furthermore, we have presented two differentially private gradient descent algorithms using compressed gradient only. Our result shows that we can significantly reduce the dimensionality dependence in their optimisation and generalisation bounds if the constraint set has a simple structure. A natural extension of the research is whether CompSGD and its variants can achieve similar results in the non-convex setting. Using the knowledge from the convex setting and extending the analysis to non-convex settings will be an interesting open research problem.

Acknowledgments

The work was done when Yunwen was with the School of Computer Science, University of Birmingham.

References

[Agarwal et al., 2018] Agarwal, N., Suresh, A. T., Yu, F., Kumar, S., and Mcmahan, H. B. (2018). cpsgd: Communication-efficient and differentially-private distributed sgd. *arXiv preprint arXiv:1805.10559*.

[Alistarh et al., 2017] Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. (2017). Qsgd: Communication-efficient sgd via gradient quantization and encoding. *NeurIPS*, 30.

[Alistarh et al., 2018] Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. (2018). The convergence of sparsified gradient methods. *In NeurIPS*, pages 5973–5983.

[Bach et al., 2012] Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Structured Sparsity through Convex Optimization. *Statistical Science*, 27(4):450 – 468.

[Balle et al., 2018] Balle, B., Barthe, G., and Gaboardi, M. (2018). Privacy amplification by subsampling: Tight analyses via couplings and divergences. *NeurIPS*, 31.

[Bao et al., 2021] Bao, F., Wu, G., Li, C., Zhu, J., and Zhang, B. (2021). Stability and generalization of bilevel programming in hyperparameter optimization. *NeurIPS*, 34:4529–4541.

[Bartlett et al., 2006] Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

[Bassily et al., 2020] Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. (2020). Stability of stochastic gradient descent on nonsmooth convex losses. *NeurIPS*, 33:4381–4391.

[Bassily et al., 2014] Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE.

[Bottou et al., 2018] Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311.

[Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.

[Charles and Papailiopoulos, 2018] Charles, Z. and Papailiopoulos, D. (2018). Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learning*, pages 745–754. PMLR.

[Chen et al., 2020] Chen, X., Wu, S. Z., and Hong, M. (2020). Understanding gradient clipping in private sgd: A geometric perspective. *NeurIPS*, 33:13773–13782.

[Chen et al., 2018] Chen, Y., Jin, C., and Yu, B. (2018). Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*.

[Dasgupta and Gupta, 2003] Dasgupta, S. and Gupta, A. (2003). An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65.

[Devroye and Wagner, 1979] Devroye, L. and Wagner, T. (1979). Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604.

[Dwork, 2006] Dwork, C. (2006). Differential privacy. *Automata, Languages and Programming*, pages 1–12.

[Dwork et al., 2006] Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

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

[Elisseeff et al., 2005] Elisseeff, A., Evgeniou, T., and Pontil, M. (2005). Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(55-79).

[Gonçalves et al., 2014] Gonçalves, A. R., Das, P., Chatterjee, S., Sivakumar, V., Von Zuben, F. J., and Banerjee, A. (2014). Multi-task sparse structure learning. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 451–460.

[Gordon, 1988] Gordon, Y. (1988). On milman's inequality and random subspaces which escape through a mesh in $\mathbb{R}^n$. In *Geometric aspects of functional analysis*, pages 84–106. Springer.

[Hardt et al., 2016] Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR.

[Jaggi, 2011] Jaggi, M. (2011). *Sparse convex optimization methods for machine learning.* PhD thesis, ETH Zurich.

[Kabán, 2016] Kabán, A. (2016). A new look at nearest neighbours: Identifying benign input geometries via random projections. In *Asian Conference on Machine Learning*, pages 65–80. PMLR.

[Kasiviswanathan, 2021] Kasiviswanathan, S. P. (2021). Sgd with low-dimensional gradients with applications to private and distributed learning. In *Uncertainty in Artificial Intelligence*, pages 1905–1915. PMLR.

[Kenthapadi et al., 2012] Kenthapadi, K., Korolova, A., Mironov, I., and Mishra, N. (2012). Privacy via the johnson-lindenstrauss transform. *arXiv preprint arXiv:1204.2606*.

[Konečnỳ et al., 2015] Konečnỳ, J., Liu, J., Richtárik, P., and Takáč, M. (2015). Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255.

[Kuzborskij and Lampert, 2018] Kuzborskij, I. and Lampert, C. (2018). Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 2815–2824. PMLR.

[Lei and Ying, 2020] Lei, Y. and Ying, Y. (2020). Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pages 5809–5819. PMLR.

[Lei and Ying, 2021] Lei, Y. and Ying, Y. (2021). Sharper generalization bounds for learning with gradient-dominated objective functions. In *ICLR*.

[Liu et al., 2009] Liu, J., Chen, J., and Ye, J. (2009). Large-scale sparse logistic regression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 547–556, New York, NY, USA.

[Liu et al., 2017] Liu, T., Lugosi, G., Neu, G., and Tao, D. (2017). Algorithmic stability and hypothesis complexity. In *International Conference on Machine Learning*, pages 2159–2167. PMLR.

[London et al., 2016] London, B., Huang, B., and Getoor, L. (2016). Stability and generalization in structured prediction. *The Journal of Machine Learning Research*, 17(1):7808–7859.

[Maurya and Toshniwal, 2018] Maurya, C. K. and Toshniwal, D. (2018). Large-scale distributed sparse class-imbalance learning. *Information Sciences*, 456:1–12.

[Nesterov, 2003] Nesterov, Y. (2003). *Introductory Lectures on Convex Optimization*, volume 87. Springer Science & Business Media.

[Nikolakakis et al., 2022] Nikolakakis, K. E., Haddadpour, F., Karbasi, A., and Kalogerias, D. S. (2022). Beyond lipschitz: Sharp generalization and excess risk bounds for full-batch gd. *arXiv preprint arXiv:2204.12446*.

[Richards and Kuzborskij, 2021] Richards, D. and Kuzborskij, I. (2021). Stability & generalisation of gradient descent for shallow neural networks without the neural tangent kernel. *NeurIPS*, 34:8609–8621.

[Shalev-Shwartz et al., 2010] Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2010). Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670.

[Shamir and Zhang, 2013] Shamir, O. and Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR.

[Showkatbakhsh et al., 2018] Showkatbakhsh, M., Karakus, C., and Diggavi, S. (2018). Privacy-utility trade-off of linear regression under random projections and additive noise. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 186–190. IEEE.

[Song et al., 2013] Song, S., Chaudhuri, K., and Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE.

[Stich et al., 2018] Stich, S. U., Cordonnier, J.-B., and Jaggi, M. (2018). Sparsified sgd with memory. *NeurIPS*, 31:4447–4458.

[Tan et al., 2018] Tan, K. M., Wang, Z., Zhang, T., Liu, H., and Cook, R. D. (2018). A convex formulation for high-dimensional sparse sliced inverse regression. *Biometrika*, 105(4):769–782.

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

[Wang et al., 2018] Wang, H., Sievert, S., Liu, S., Charles, Z., Papailiopoulos, D., and Wright, S. (2018). Atomo: Communication-efficient learning via atomic sparsification. *NeurIPS*, 31:9850–9861.

[Wang et al., 2022] Wang, P., Lei, Y., Ying, Y., and Zhang, H. (2022). Differentially private sgd with non-smooth losses. *Applied and Computational Harmonic Analysis*, pages 306–336.

[Xing et al., 2021] Xing, Y., Song, Q., and Cheng, G. (2021). On the algorithmic stability of adversarial training. *NeurIPS*, 34:26523–26535.

[Xu et al., 2017] Xu, C., Ren, J., Zhang, Y., Qin, Z., and Ren, K. (2017). Dppro: Differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security*, 12(12):3081–3093.

[Zhang et al., 2021] Zhang, J., Hong, M., Wang, M., and Zhang, S. (2021). Generalization bounds for stochastic saddle point problems. In *International Conference on Artificial Intelligence and Statistics*, pages 568–576. PMLR.

[Zhang, 2004] Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116.

[Zhao and Zhang, 2014] Zhao, P. and Zhang, T. (2014). Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*.

A Missing Details and Proofs

In this section, we present the missing proofs for Section 3 and the proofs of our results in Section 4.1 and Section 4.2. We first state the preliminary Theorems that we will use during some steps of our analysis.

Theorem 17 ([Nesterov, 2003]). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and μ-smooth function. We have $\forall x, y \in \mathbb{R}^d$:*

1. *(upper bound) $f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|^2$;*
2. *(co-coercivity) $\frac{1}{\mu}\|\nabla f(x) - \nabla f(y)\|^2 \leq \langle \nabla f(x) - \nabla f(y), x - y \rangle$;*
3. *(lower bound) $f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2\mu}\|\nabla f(x) - \nabla f(y)\|^2$.*

Lemma 5 (Non-expansitivity of convex-smooth gradient updates). *Let f be a loss function that is convex and μ-smooth. For gradient updates of the form $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t, z_{i_t})$, assume that two gradient updates $\mathbf{w}_{t+1}, \mathbf{w}'_{t+1}$ uses the same samples for update, i.e. $\mathbf{w}'_{t+1} = \mathbf{w}'_t - \eta_t \nabla f(\mathbf{w}'_t, z_{i_t})$. Then we have $\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\| \leq \|\mathbf{w}_t - \mathbf{w}'_t\|$ for all $\eta_t \leq 1/(2\mu)$.*

Proof. We have

$$\begin{aligned}\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 &= \|\mathbf{w}_t-\eta_t\nabla f(\mathbf{w}_t,z_{i_t})-(\mathbf{w}'_t-\eta_t\nabla f(\mathbf{w}'_t,z_{i_t}))\|^2\\
&= \|\mathbf{w}_t-\mathbf{w}'_t\|^2-2\eta_t\langle\mathbf{w}_t-\mathbf{w}'_t,\nabla f(\mathbf{w}_t,z_{i_t})-\nabla f(\mathbf{w}'_t,z_{i_t})\rangle\\
&\qquad\qquad\qquad+\eta_t^2\|\nabla f(\mathbf{w}_t,z_{i_t})-\nabla f(\mathbf{w}'_t,z_{i_t})\|^2\\
&\le \|\mathbf{w}_t-\mathbf{w}'_t\|^2-2\eta_t\frac{1}{\mu}\|\nabla f(\mathbf{w}_t,z_{i_t})-\nabla f(\mathbf{w}'_t,z_{i_t})\|\\
&\qquad\qquad\qquad+\eta_t^2\|\nabla f(\mathbf{w}_t,z_{i_t})-\nabla f(\mathbf{w}'_t,z_{i_t})\|^2\\
&= \|\mathbf{w}_t-\mathbf{w}'_t\|^2+(\eta_t^2-\frac{2\eta_t}{\mu})\|\nabla f(\mathbf{w}_t,z_{i_t})-\nabla f(\mathbf{w}'_t,z_{i_t})\|^2, \qquad (69)\end{aligned}$$

where the last inequality is by applying the co-coercivity of convex and smooth functions. Now, by assuming that $\eta_t \le \frac{1}{2\mu}$ we eliminated the last term, we conclude our result $\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\| \le \|\mathbf{w}_t-\mathbf{w}'_t\|$. □

Remark 8. From the proof of Lemma 5, we note we do not require the gradient to be stochastic. Hence non-expansitivity holds for batch and mini-batch gradients using the same argument, as long as the condition for η_t holds.

Theorem 18 (Gordon's Theorem [Gordon, 1988]). *Let $m, d \in \mathbb{N}$, let $\Phi \in \mathbb{R}^{m\times d}$ be a random matrix with independent $\mathcal{N}(0,1/m)$ entries. Let $B \subset \mathbb{S}^{d-1}$ be a subset of the unit sphere in d dimensions. If $m = \Theta(\omega(B)^2/\beta^2)$, then*

$$\mathbb{E}_{\Phi}\left[\sup_{\mathbf{x}\in B}|\|\Phi\mathbf{x}\|^2-1|\right] \le \beta, \qquad (70)$$

where $\omega(B)$ is the Gaussian width of B and the expectation $\mathbb{E}_{\Phi}[\cdot]$ is over the randomness in Φ.

Gordon's Theorem is a key result to bound the expected norm of projected points with respect to the norm of original points. We note that the projection dimension must increase as β decreases, implying we need to project onto a higher dimension if we wish to decrease the distortion.

The following result bounds the norm of the gradient update in CompSGD with a fixed point $\mathbf{w} \in \mathcal{C}$ which will be useful for the analysis of optimisation step. We provide the proof here for completeness.

Theorem 19 ([Kasiviswanathan, 2021]). *In Algorithm 1 CompSGD, for any $t \in [T]$, we have for all $\mathbf{w} \in \mathcal{C}$*

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}\|^2 \leq \|\mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t, z) - \mathbf{w}\|^2. \tag{71}$$

Remark 9. Note that from the proof of Theorem 19, there are no requirements on the gradient used (∇f) being stochastic. Hence the same result will hold for batch (∇F_S) and mini-batch gradients (∇F_B).

Proof of Theorem 19. At iteration t, fix a RP matrix Φ_t. Define the normalizing map $u : \mathbb{R}^d \to \mathbb{R}^d$ as $u(\mathbf{w}) = \frac{\mathbf{w}}{\|\mathbf{w}\|}$. Let $\mathbf{w} \in \mathcal{C}$ be any vector. To simplify notation, note that $\mathbf{w}_{t+1} - \mathbf{w} \in \mathcal{C} + \mathcal{C}$ (the Minkowski sum) and denote $\mathcal{C}' = \{u(\mathbf{w}) \mid \mathbf{w} \in \mathcal{C} + \mathcal{C}\}$. Since $u(\mathbf{w}_{t+1} - \mathbf{w}) \in \mathcal{C}'$, we have

$$|\|\Phi_t u(\mathbf{w}_{t+1} - \mathbf{w})\|^2 - 1| \leq \sup_{\mathbf{w}\in\mathcal{C}'} |\|\Phi_t \mathbf{w}\|^2 - 1|. \tag{72}$$

Eq. (72) holds for all Φ_t. Taking expectation with respect to Φ_t, Gordon's theorem implies

$$\mathbb{E}_{\Phi_t}\left[|\|\Phi_t u(\mathbf{w}_{t+1} - \mathbf{w})\|^2 - 1|\right] \leq \mathbb{E}_{\Phi_t}\left[\sup_{\mathbf{w}\in\mathcal{C}'} |\|\Phi_t \mathbf{w}\|^2 - 1|\right] \leq \beta_t. \tag{73}$$

The above inequality can be rearranged as

$$\begin{aligned}(1-\beta_t) &\leq \mathbb{E}_{\Phi_t}\left[|\|\Phi_t u(\mathbf{w}_{t+1} - \mathbf{w})\|^2|\right] \leq (1+\beta_t),\\ \Rightarrow (1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}\|^2 &\leq \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_{t+1} - \mathbf{w})\|^2\right]. \end{aligned} \tag{74}$$

Hence we obtain

$$\begin{aligned}(74) &= \mathbb{E}_{\Phi_t}\left[\left\|\underset{\Phi_t\mathcal{C}}{\Pi}(\Phi_t\mathbf{w}_t - \eta_t\Phi_t\nabla f(\mathbf{w}_t, z_{i_t}) - \underset{\Phi_t\mathcal{C}}{\Pi}(\Phi_t\mathbf{w})\right\|^2\right]\\ &\leq \mathbb{E}_{\Phi_t}\left[\|(\Phi_t\mathbf{w}_t - \eta_t\Phi_t\nabla f(\mathbf{w}_t, z_{i_t}) - (\Phi_t\mathbf{w})\|^2\right]\\ &= \|(\mathbf{w}_t - \eta_t\nabla f(\mathbf{w}_t, z_{i_t})) - \mathbf{w}\|^2, \end{aligned} \tag{75}$$

where we have used the fact that the projection map $\Pi_{\Phi\mathcal{C}}$ is contractive in the second step, i.e. distance between two points will not be larger after projection onto $\Phi\mathcal{C}$; and the final step follows since Φ_t is independent from all the remaining variables, $\mathbf{w}_t, \mathbf{w}, \eta_t, z_{i_t}$. □

A.1 Proofs for CompSGD in Section 3

Proof of Lemma 1. The proof of lemma 1 follows similar derivation as for Theorem 19 using Gordon's Theorem. Except that we are bounding the distortion between two gradient updates rather than a fixed point $\mathbf{w} \in \mathcal{C}$.

We start by replacing $\mathbf{w}$ with $\mathbf{w}'_{t+1}$ from equation (74), we have

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 &\leq \mathbb{E}_{\Phi_t}\left[\|\Phi_t(\mathbf{w}_{t+1}-\mathbf{w}'_{t+1})\|^2\right] \\ &= \mathbb{E}_{\Phi_t}\left[\left\|\underset{\Phi_t\mathcal{C}}{\Pi}\left(\Phi_t\mathbf{w}_t-\eta_t\Phi_t\nabla f(\mathbf{w}_t,z_{i_t})-\left(\Phi_t\mathbf{w}'_t-\eta_t\Phi_t\nabla f(\mathbf{w}'_t,z'_{i_t})\right)\right)\right\|^2\right] \\ &\leq \mathbb{E}_{\Phi_t}\left[\left\|(\Phi_t\mathbf{w}_t-\eta_t\Phi_t\nabla f(\mathbf{w}_t,z_{i_t})-(\Phi_t\mathbf{w}'_t-\eta_t\Phi_t\nabla f(\mathbf{w}'_t,z'_{i_t}))\right\|^2\right] \\ &= \|(\mathbf{w}_t-\mathbf{w}'_t)-(\eta_t\nabla f(\mathbf{w}_t,z_{i_t})-\eta_t\nabla f(\mathbf{w}'_t,z'_{i_t}))\|^2, \qquad (76)\end{aligned}$$

where we have used the fact that the projection map $\Pi_{\Phi\mathcal{C}}$ is contractive in the second step, i.e. distance between two points will not be larger after projection onto $\Phi\mathcal{C}$; and the final step follows since Φ_t is independent from all the remaining variables, $\mathbf{w}_t, \mathbf{w}'_t, \eta_t, z_{i_t}, z'_{i_t}$. □

Remark 10. Similar to Thm. 19, there are no requirements on the gradient used (∇f) being stochastic. Hence the same result will hold for batch (∇F_S) and mini-batch gradients (∇F_B).

Proof of Lemma 2. Since we assumed $z_{i_t} = z'_{i_t}$ in the runs of CompSGD, we use the shorthand $\nabla f(\mathbf{w}_t)$ and $\nabla f(\mathbf{w}'_t)$ for $\nabla f(\mathbf{w}_t, z_{i_t})$ and $\nabla f(\mathbf{w}'_t, z_{i_t})$ as respectively. Denote $\mathbf{w}_t - \mathbf{w}'_t$ by Δ_t. From Lemma 1 we have

$$(1-\beta_t)\|\Delta_{t+1}\|^2 \leq \|\Delta_t\|^2 - 2\eta_t\langle\nabla f(\mathbf{w}_t)-\nabla f(\mathbf{w}'_t),\Delta_t\rangle + \eta_t^2\|\nabla f(\mathbf{w}_t)-\nabla f(\mathbf{w}'_t)\|^2.$$

By Part 2 of Lemma 17 (co-coercivity), the second term on the r.h.s. is further bounded as

$$\langle\nabla f(\mathbf{w}_t)-\nabla f(\mathbf{w}'_t),\Delta_t\rangle \geq \frac{1}{\mu}\|\nabla f(\mathbf{w}_t)-\nabla f(\mathbf{w}'_t)\|^2. \qquad (77)$$

Hence, we have

$$(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 \leq \|\mathbf{w}_t-\mathbf{w}'_t\|^2 - \left(\frac{2\eta_t}{\mu}-\eta_t^2\right)\|\nabla f(\mathbf{w}_t)-\nabla f(\mathbf{w}'_t)\|^2. \qquad (78)$$

Setting $\eta_t \leq 2/\mu$ eliminates the last term in (78), and the result follows. □

Proof of Lemma 3. By Theorem 19, we have that

$$(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2 \leq \|\mathbf{w}_t-\eta_t\nabla f(\mathbf{w}_t,z)-\mathbf{w}_t\|^2.$$

Hence, $\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2 \leq \frac{\eta_t^2}{1-\beta_t}\|\nabla f(\mathbf{w}_t,z)\|^2 \leq \frac{\eta_t^2 L^2}{1-\beta_t}$, where the last inequality is a consequence of the L-Lipschitz assumption on f. □

A.2 Proofs for Compressed Gradient Descent

In this section, we present the proofs for our result in section 4.1. We start with the optimisation of CompGD in the smooth case, where we are able to obtain a faster convergence compared to CompSGD with a larger step size parameter.

Proof of Theorem 4. By Thm. 19 with $\mathbf{w}=\mathbf{w}_S^*$ we have

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}_S^*\|^2 &\leq \|\mathbf{w}_t-\eta_t\nabla F_S(\mathbf{w}_t)-\mathbf{w}_S^*\|^2\\ &= \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+2\eta_t\langle\nabla F_S(\mathbf{w}_t),\mathbf{w}_S^*-\mathbf{w}_t\rangle+\eta_t^2\|\nabla F_S(\mathbf{w}_t)\|^2\\ &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+2\eta_t(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+\eta_t^2\|\nabla F_S(\mathbf{w}_t)\|^2, \quad (79)\end{aligned}$$

where the last line follows from the convexity of f. From the smoothness we also have

$$\begin{aligned}F_S(\mathbf{w}_t)-F_S(\mathbf{w}_S^*) &\geq \langle\mathbf{w}_t-\mathbf{w}_S^*,\nabla F_S(\mathbf{w}_t)\rangle+\frac{1}{2\mu}\|\nabla F_S(\mathbf{w}_t)-\nabla F_S(\mathbf{w}_S^*)\|^2\\ &\geq \frac{1}{2\mu}\|\nabla F_S(\mathbf{w}_t)-\nabla F_S(\mathbf{w}_S^*)\|^2,\end{aligned}$$

where we have used $\langle\mathbf{w}_t-\mathbf{w}_S^*,\nabla F_S(\mathbf{w}_t)\rangle \geq 0$ by convexity and $\mathbf{w}_S^*$ is a minimiser of $F_S(\mathbf{w})$. Applying this property we have

$$\|\nabla F_S(\mathbf{w}_t)\|^2 = \|\nabla F_S(\mathbf{w}_t)-\nabla F_S(\mathbf{w}_S^*)\|^2 \leq 2\mu(F_S(\mathbf{w}_t)-F_S(\mathbf{w}_S^*)). \quad (80)$$

Substituting equation (80) into (79) we have

$$\begin{aligned}(79) &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+2\eta_t(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+\eta_t^2\|\nabla F_S(\mathbf{w}_t)-\nabla F_S(\mathbf{w}_S^*)\|^2\\ &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+2\eta_t(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t))+2\eta_t^2\mu(F_S(\mathbf{w}_t)-F_S(\mathbf{w}_S^*))\\ &\leq \|\mathbf{w}_t-\mathbf{w}_S^*\|^2+\frac{\eta_t}{2}(F_S(\mathbf{w}_S^*)-F_S(\mathbf{w}_t)) \text{ (assuming } \eta_t \leq 1/(2\mu)). \quad (81)\end{aligned}$$

Rearranging we have:

$$\frac{\eta_t}{2}(F_S(\mathbf{w}_t) - F_S(\mathbf{w}_S^*)) \leq \|\mathbf{w}_t - \mathbf{w}_S^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|^2 + \beta_t\|\mathbf{w}_{t+1} - \mathbf{w}_S^*\|^2.$$

Taking expectation and summing over T iterates and choosing $\beta_t = 1/(t+1)$ we have:

$$\begin{aligned}\sum_{t=1}^{T}\frac{\eta_t}{2}\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}_S^*)] &\leq \|\mathbf{w}_1-\mathbf{w}_S^*\|^2+\sum_{t=1}^{T}\beta_t\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}_S^*\|^2]\\ &= \mathcal{O}\left(\|\mathcal{C}\|^2 + \|\mathcal{C}\|^2\sum_{t=1}^{T}\beta_t\right)\\ &= \mathcal{O}\left(\|\mathcal{C}\|^2+\|\mathcal{C}\|^2\log(T)\right), \qquad (82)\end{aligned}$$

where we have used $\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}_S^*\|]^2 \leq \|\mathcal{C}\|^2$. Finally, for $\eta_t = \eta$ being an absolute constant we have

$$\left(\sum_{t=1}^{T}\eta_t\right)^{-1}\sum_{t=1}^{T}\eta_t\mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}_S^*)] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2\log(T)}{T}\right). \qquad (83)$$

The proof is completed. □

CompGD differs from CompSGD in the way that the gradient update used in CompGD is non-stochastic, which implies that if we have two similar but different training sample set S, S', then the gradient update rule G, G' is guaranteed to be different at every iteration. Hence our analysis here is a little different CompSGD, where we overcome this problem by noting that the difference in the gradient is small from neighbouring sample sets. We now formally prove Thm.5.

Proof of Theorem 5. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. WLOG we assume that the sample where S, S' differs is at index j: we denote z_j, z_j' for the sample in S and S' respectively. Fix a sample z, by the Lipschitz condition we get that

$$\mathbb{E}[|f(\mathbf{w}_T, z) - f(\mathbf{w}_T', z)|] \leq L\mathbb{E}[\delta_T], \qquad (84)$$

where $\delta_T = \|\mathbf{w}_T - \mathbf{w}_T'\|$. At iteration t, we have from Lemma 1

$$\sqrt{1-\beta_t}\|\mathbf{w}_{t+1} - \mathbf{w}_{t+1}'\| \leq \|(\mathbf{w}_t - \mathbf{w}_t') - \eta_t(\nabla F_S(\mathbf{w}_t) - \nabla F_S(\mathbf{w}_t', S'))\|. \qquad (*)$$

Note that since S, S' only differ on the j-th point, we have $S \cup \{z_j'\} = S' \cup \{z_j\}$.

$$
\begin{aligned}
(*) &= \left\| \mathbf{w}_t - \mathbf{w}'_t - \frac{\eta_t}{n} \sum_{z \in S \cup \{z'_j\}} (\nabla f(\mathbf{w}_t, z) - \nabla f(\mathbf{w}'_t, z)) + \frac{\eta_t}{n} (\nabla f(\mathbf{w}_t, z'_j) - \nabla f(\mathbf{w}'_t, z_j)) \right\| \\
&\leq \|\mathbf{w}_t - \mathbf{w}'_t\| + \frac{\eta_t}{n} \left\| \nabla f(\mathbf{w}_t, z_j) - \nabla f(\mathbf{w}'_t, z'_j) \right\| \leq \|\mathbf{w}_t - \mathbf{w}'_t\| + \frac{2L\eta_t}{n}, \qquad (85)
\end{aligned}
$$

where we have used the non-expansitivity of the gradient update rule (Lemma 5) and the sub-additivity of the norm on the second step. The last inequality is by applying the L-Lipschitz condition of f. Therefore we have the recursion

$$
\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\| \leq \frac{\|\mathbf{w}_t - \mathbf{w}'_t\|}{\sqrt{1-\beta_t}} + \frac{2L\eta_t}{n\sqrt{1-\beta_t}}. \qquad (86)
$$

By the same argument as in proof of Theorem 2 starting with equation (19), we have:

$$
\mathbb{E}[\delta_T] \leq \frac{2L}{n} \sum_{t=1}^{T-1} \eta_t \prod_{j=t}^{T-1} (1-\beta_j)^{-1/2}. \qquad (87)
$$

By letting $\beta_t = 1/(t+1)$, $\eta_t = \eta$ for some absolute constant η and $T \asymp \sqrt{n}$, we have

$$
\mathbb{E}[\delta_T] \leq \frac{2L\eta\sqrt{T}}{n} \sum_{t=1}^{T-1} \frac{1}{\sqrt{t}} = \mathcal{O}\left(\frac{L\eta}{\sqrt{n}} \right). \qquad (88)
$$

□

Proof of Theorem 6. Let ∇F_t denote the gradient $\frac{1}{n}\sum_{z \in S} \nabla f(\mathbf{w}_t, z)$ at iteration t. By Jensen's inequality, we have

$$
\begin{aligned}
F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*) &= F_S\left(\left(\sum_{t=1}^{T} \eta_t \right)^{-1} \sum_{t=1}^{T} \eta_t \mathbf{w}_t \right) - F_S(\mathbf{w}^*) \\
&\leq \left(\sum_{t=1}^{T} \eta_t \right)^{-1} \sum_{t=1}^{T} \eta_t (F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)) \\
&\leq \left(\sum_{t=1}^{T} \eta_t \right)^{-1} \sum_{t=1}^{T} \eta_t \langle \nabla F_t, \mathbf{w}_t - \mathbf{w}^* \rangle, \qquad (89)
\end{aligned}
$$

where the last inequality is by the convexity of f. To bound the terms above, note that we have for all $t \geq 1$:

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 &\leq \|(\mathbf{w}_t-\mathbf{w}^*)-\eta_t\nabla F_t\|^2\\ &= \|\mathbf{w}_t-\mathbf{w}^*\|^2 + 2\eta_t\langle\nabla F_t, \mathbf{w}^*-\mathbf{w}_t\rangle + 4\eta_t^2L^2.\end{aligned}$$

Rearranging we have:

$$2\eta_t\langle\nabla F_t, \mathbf{w}_t-\mathbf{w}^*\rangle \leq \|\mathbf{w}_t-\mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 + \beta_t\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 + 4\eta_t^2L^2. \qquad (90)$$

Taking expectation and summing over T iterates we have:

$$\begin{aligned}2\sum_{t=1}^{T}\eta_t\mathbb{E}[\langle\nabla f_t, \mathbf{w}_t-\mathbf{w}^*\rangle] &\leq \|\mathbf{w}_1-\mathbf{w}^*\|^2 + \sum_{t=1}^{T}\beta_t\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2] + 4L^2\sum_{t=1}^{T}\eta_t^2\\ &= \mathcal{O}\left(\|\mathcal{C}\|^2\log(T) + L^2\sum_{t=1}^{T}\eta_t^2\right), \qquad (91)\end{aligned}$$

where we have used $\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|]^2 \leq \|\mathcal{C}\|^2$ and $\beta_t = \frac{1}{t+1}$ in the last line. Hence by choosing $\eta_t = \eta/\sqrt{t}$ for some absolute constant η, we have

$$\left(\sum_{t=1}^{T}\eta_t\right)^{-1}\sum_{t=1}^{T}\eta_t\mathbb{E}[\langle\nabla F_t, \mathbf{w}_t-\mathbf{w}^*\rangle] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2\log(T)}{\sqrt{T}} + \frac{L^2}{\sqrt{T}}\sum_{t=1}^{T}\frac{1}{t}\right). \qquad (92)$$

Finally, combining the above inequalities we have

$$\mathbb{E}[F_S(\bar{\mathbf{w}}_T) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{(\|\mathcal{C}\|^2+L^2)\log(T)}{\sqrt{T}}\right). \qquad (93)$$

□

Proof of Theorem 7. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. W.l.o.g. assume that they differ on the j-th point denoted z_j, z_j' for S and S', respectively. Fix a sample z, by the Lipschitz condition we get

$$\mathbb{E}[|f(\mathbf{w}_T) - f(\mathbf{w}_T')|] \leq L\mathbb{E}[\delta_T], \qquad (94)$$

where $\delta_T = \|\mathbf{w}_T - \mathbf{w}_T'\|$.

Since after each update, $\mathbf{w}_t \in \mathcal{C}$ for all t, by Lemma 1 we have (here we will denote

$\nabla F_S(\mathbf{w}_t)$ by ∇F_t and $\nabla F_{S'}(\mathbf{w}'_t)$ by $\nabla F'_t$)

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 &\le \|(\mathbf{w}_t-\mathbf{w}'_t)-\eta_t(\nabla F_S(\mathbf{w}_t)-\nabla F_{S'}(\mathbf{w}'_t))\|^2 \\ &= \|\mathbf{w}_t-\mathbf{w}'_t\|^2+2\eta_t\langle \mathbf{w}'_t-\mathbf{w}_t, \nabla F_t-\nabla F'_t\rangle+4\eta_t^2L^2, \qquad (95)\end{aligned}$$

where the last line follows from the Lipschitz assumption.

Since S, S' only differs on the j-th point, we have

$$\begin{aligned}\langle \mathbf{w}'_t-\mathbf{w}_t, \nabla F_t-\nabla F'_t\rangle &= \langle \mathbf{w}'_t-\mathbf{w}_t, \nabla F_{S\cup\{z'_j\}}(\mathbf{w}_t)-\nabla F_{S'\cup\{z_j\}}(\mathbf{w}'_t)\rangle \\ &\quad+\frac{1}{n}\langle \mathbf{w}'_t-\mathbf{w}_t, \nabla f(\mathbf{w}'_t,z_j)-\nabla f(\mathbf{w}_t,z'_j)\rangle \\ &\le \frac{1}{n}\langle \mathbf{w}'_t-\mathbf{w}_t, \nabla f(\mathbf{w}'_t,z_j)-\nabla f(\mathbf{w}_t,z'_j)\rangle \\ &\le \frac{2L}{n}\|\mathbf{w}_t-\mathbf{w}'_t\|, \qquad (96)\end{aligned}$$

where the second line holds because the first term is negative by the convexity of F_S, and the last line follows from the Lipschitz condition.

We substitute the inequality (96) into equation (95) and obtain:

$$(1-\beta_t)\delta_{t+1}^2 = \delta_t^2+4L\eta_t\left(\frac{\delta_t}{n}+L\eta_t\right). \qquad (97)$$

By multiplying both sides by $\prod_{j=1}^{t-1}(1-\beta_j)$, we have

$$\left[\prod_{j=1}^{t}(1-\beta_j)\right]\delta_{t+1}^2 \le \left[\prod_{j=1}^{t-1}(1-\beta_j)\right]\delta_t^2+\frac{4\eta_tL\delta_t}{n}\prod_{j=1}^{t-1}(1-\beta_j)+4\eta_t^2L^2\left[\prod_{j=1}^{t-1}(1-\beta_j)\right]. \qquad (98)$$

By summing over T iterates we have:

$$\left[\prod_{j=1}^{T-1}(1-\beta_j)\right]\delta_T^2 \le \sum_{t=1}^{T-1}\frac{4\eta_tL\delta_t}{n}\prod_{j=1}^{t-1}(1-\beta_j)+\sum_{t=1}^{T-1}4\eta_t^2L^2\left[\prod_{j=1}^{t-1}(1-\beta_j)\right]. \qquad (99)$$

Taking $\beta_t = 1/t+1$ and rearranging we have:

$$\delta_T^2 \le \frac{4LT}{n}\sum_{t=1}^{T-1}\frac{\eta_t\delta_t}{t}+4L^2T\sum_{t=1}^{T-1}\frac{\eta_t^2}{t}. \qquad (100)$$

The rest of the proof follows from the same procedure as in the proof of Thm. 13, starting with equation (62).

□

A.3 Proofs for Compressed Mini-batch SGD

Proof of Theorem 8. By Thm.19 with $\mathbf{w} = \mathbf{w}^*$, we have for all $t \geq 1$:

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 &\leq \|(\mathbf{w}_t - \mathbf{w}^*) - \frac{\eta_t}{b}\sum_{z\in B_t}\nabla f(\mathbf{w}_t, z_{i_t})\|^2 \\ &= \|\mathbf{w}_t - \mathbf{w}^*\|^2 + 2\eta_t\Big\langle\frac{1}{b}\sum_{z\in B_t}\nabla f(\mathbf{w}_t, z), \mathbf{w}^* - \mathbf{w}_t\Big\rangle + \eta_t^2\|\frac{1}{b}\sum_{z\in B_t}\nabla f(\mathbf{w}_t, z)\|^2 \\ &\leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 + \frac{2\eta_t}{b}\sum_{z\in B_t}(f(\mathbf{w}^*, z) - f(\mathbf{w}_t, z)) + \eta_t^2 L^2.\end{aligned}$$

Rearranging the above inequality gives

$$\frac{2\eta_t}{b}\sum_{z\in B_t}(f(\mathbf{w}_t) - f(\mathbf{w}^*)) \leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 + \beta_t\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 + \eta_t^2 L^2.$$

Since each mini-batch B_t is drawn uniformly from the sample, we note that $\mathbb{E}_A[F_{B_t}(\mathbf{w})] = \mathbb{E}_A[F_S(\mathbf{w})]$. Hence, setting $\beta_t = 1/(t+1)$, taking expectation and summing over T iterates give

$$\begin{aligned}2\sum_{t=1}^{T}\eta_t\mathbb{E}_{S,A}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)] &\leq \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=1}^{T}\beta_t\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] + L^2\sum_{t=1}^{T}\eta_t^2 \\ &= \mathcal{O}\left(\|\mathcal{C}\|^2 + \|\mathcal{C}\|^2\sum_{t=1}^{T}\frac{1}{t+1} + L^2\sum_{t=1}^{T}\eta_t^2\right) \\ &= \mathcal{O}\left(\|\mathcal{C}\|^2\log(T) + L^2\sum_{t=1}^{T}\eta_t^2\right),\end{aligned}$$

where we have used $\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}^*\|]^2 \leq \|\mathcal{C}\|^2$. Finally, choosing $\eta_t = \frac{\eta}{\sqrt{t}}$ we have

$$\left(\sum_{t=1}^{T}\eta_t\right)^{-1}\sum_{t=1}^{T}\eta_t\mathbb{E}[F_S(\mathbf{w}_t) - F_S(\mathbf{w}^*)] = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2\log(T)}{\sqrt{T}} + \frac{L^2\log(T)}{\sqrt{T}}\right). \tag{101}$$

The proof is completed. □

Proof of Theorem 9. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. Denote the gradient updates by $G_1, \ldots, G_T$ and $G'_1, \ldots, G'_T$ induced by running the CompSGD on S and S', respectively. Let $\delta_T = \|\mathbf{w}_T - \mathbf{w}'_T\|$.

Observe that at step t, with probability $1 - b/n$, the mini-batch B_t, B'_t selected is the same in both S and S'. In this case we have $G_t = G'_t$ and we use the expansivity of the update G_t by a similar proof as Lemma 2. With probability b/n the selected mini-batch B_t is different in which case assume they differ by the j-th point and we have from Lemma 1

$$\sqrt{1-\beta_t}\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\| \le \|(\mathbf{w}_t-\mathbf{w}'_t)-\eta_t(\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B'_t}(\mathbf{w}'_t))\|. \qquad (*)$$

Note that since B_t, B'_t only differ on the j-th point, we have $B_t \cup \{z'_j\} = B'_t \cup \{z_j\}$.

$$\begin{aligned}(*) &= \left\|(\mathbf{w}_t-\mathbf{w}'_t) - \frac{\eta_t}{b}\sum_{z\in B_t\cup\{z'_j\}}(\nabla f(\mathbf{w}_t,z)+\nabla f(\mathbf{w}'_t,z)) - \frac{\eta_t}{b}(\nabla f(\mathbf{w}_t,z'_j)-\nabla f(\mathbf{w}'_t,z_j))\right\| \\ &\le \|\mathbf{w}_t-\mathbf{w}'_t\| + \frac{\eta_t}{b}\left\|\nabla f(\mathbf{w}_t,z_j)-\nabla f(\mathbf{w}'_t,z'_j)\right\| \le \|\mathbf{w}_t-\mathbf{w}'_t\| + \frac{2L\eta_t}{b}, \qquad (102)\end{aligned}$$

where we have used the non-expansitivity of gradient update (Lemma 5) and the subadditivity of the norm on the second step. The last inequality is by applying the L-Lipschitz condition of f. Hence, combining the two cases and by the linearity of expectation we have the following:

$$\mathbb{E}[\delta_{t+1}] \le \left(\frac{1-b/n}{\sqrt{1-\beta_t}}\right)\mathbb{E}[\delta_t] + \frac{b}{n\sqrt{1-\beta_t}}\left(\mathbb{E}[\delta_t]+\frac{2L\eta_t}{b}\right). \qquad (103)$$

The rest of the proof for stability then follows by the same argument as in the proof of Theorem 2 starting with equation (19).

□

Proof of Theorem 10. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. Let $G(\mathbf{w}_t) = \mathbf{w}_{t+1}$ denote the gradient update and let $G_1, \ldots, G_T$ and $G'_1, \ldots, G'_T$ be the updates induced by running the CompSGD on S and S' for T iterates, respectively. Let $\delta_T = \|\mathbf{w}_T - \mathbf{w}'_T\|$, by the Lipschitz condition,

$$\mathbb{E}[|f(\mathbf{w}_T,z)-f(\mathbf{w}'_T,z)|] \le L\mathbb{E}[\delta_T]. \qquad (104)$$

If at iteration t, the mini-batch B_t, B'_t we selected is the same, i.e. $G_t = G'_t$, then from Lemma 1 we have the following

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 &\le \|(\mathbf{w}_t-\mathbf{w}'_t)-\eta_t(\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B_t}(\mathbf{w}'_t))\|^2 \\ &= \|\mathbf{w}_t-\mathbf{w}'_t\|^2 - 2\eta_t\langle\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B_t}(\mathbf{w}'_t), \mathbf{w}_t-\mathbf{w}'_t\rangle \\ &\quad + \eta_t^2\|\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B_t}(\mathbf{w}'_t)\|^2.\end{aligned}$$

From the convexity of f we have that $\langle \nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B_t}(\mathbf{w}'_t), \mathbf{w}_t - \mathbf{w}'_t\rangle \geq 0$ and from Lipschitzness of f we also have $\|\nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B_t}(\mathbf{w}'_t)\| \leq 2L$. Hence we obtain the following bound

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 \leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + 4L^2\eta_t^2. \tag{105}$$

For the case where $G_t \neq G'_t$, note that B_t and B'_t differ by a single sample, hence we have

$$\begin{aligned}(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 &\leq \|(\mathbf{w}_t - \mathbf{w}'_t) - \eta_t(\nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B'_t}(\mathbf{w}'_t))\|^2 \\ &\leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 - 2\eta_t\langle \nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B'_t}(\mathbf{w}'_t), \mathbf{w}_t - \mathbf{w}'_t\rangle \\ &\qquad + \eta_t^2\|\nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B'_t}(\mathbf{w}'_t)\|^2 \\ &\leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + \frac{4L\eta_t}{b}\|\mathbf{w}_t - \mathbf{w}'_t\| + 4L^2\eta_t^2, \end{aligned} \tag{106}$$

where the last inequality (106) follows by the same derivation as for equation (96), replacing S, S' with B_t, B'_t respectively. Combining the two cases we have

$$\begin{aligned}(1-\beta_t)\mathbb{E}[\delta_{t+1}^2] &\leq \left(1 - \frac{b}{n}\right)(\mathbb{E}[\delta_t^2] + 4L^2\eta_t^2) + \frac{b}{n}\left(\mathbb{E}[\delta_t^2] + \frac{4L\eta_t\mathbb{E}[\delta_t]}{b} + 4L^2\eta_t^2\right) \\ &= \mathbb{E}[\delta_t^2] + 4L\eta_t\left(\frac{\mathbb{E}[\delta_t]}{n} + L\eta_t\right). \end{aligned} \tag{107}$$

The rest of the proof then follows by the same procedure as the proof for Theorem 13 starting from equation (97). □

B Proofs for Differentially Private SGD

B.1 Privacy guarantees

We will use the composition theorem to guarantee DP over a series of steps that requires to query the same sample, which is the setting for iterative gradient methods.

Theorem 20 (Strong composition [Dwork and Roth, 2014]). *Let $\epsilon, \delta, \delta' > 0$ and $\epsilon \leq 1$. A mechanism that permits T adaptive interactions with mechanisms that preserves (ϵ, δ)-differential privacy ensures $(\epsilon\sqrt{2T\log(1/\delta')} + 2k\epsilon^2, T\delta + \delta')$-differential privacy.*

The privacy guarantee of DP-CompGD now follows from the strong composition theorem.

Theorem 21. *The output of DP-CompGD in Alg. 4 satisfies (ϵ, δ)-differential privacy.*

Proof. The proof follows a similar procedure as in [Bassily et al., 2014] (Thm. 2.1) with a standard application of the Gaussian mechanism. Note that the norm of the projected gradient $\Phi_t \nabla F_S(\mathbf{w}_t)$ is normalized by the normalization factor s_t (line 7 of Alg. 4). Hence the norm $\|s_t \Phi_t \nabla F_S(\mathbf{w}_t)\|$ is upper bounded by the Lipschitz constant L which implies a global sensitivity of $2L$. The privacy guarantee then follows directly by applying the Gaussian mechanism with the strong composition theorem (Thm. 20) over T iterations of SGD. □

In the case where we only use a random subset of the whole sample set in each iterate, the sensitivity will increase due to a smaller sample set, however we can apply the following result to strengthen our privacy guarantee:

Theorem 22 (Amplification by subsampling [Balle et al., 2018]). *Let $\mathcal{X}$ be a data domain and $M : \mathcal{X}^n \to \mathcal{X}^b$ be a procedure such that $M(S)$ returns a random subset of b records sampled uniformly without replacement from S. Let A be an (ϵ, δ)-DP algorithm. Then $A \circ S$ satisfies $(\epsilon', (b/n)\delta)$-DP with $\epsilon' = \log(1 + (b/n)(e^\epsilon - 1))$.*

Theorem 23. *The output of DP-CompMiniBatch in Alg. 5 satisfies (ϵ, δ)-differential privacy.*

Proof. The proof follows the same idea as for DP-CompGD. Note that since we only use a random subset of S for each iteration (of size b), we can apply Thm. 22 to obtain a stronger privacy guarantee relative to the size of the subsample at each iteration. The privacy guarantee then follows similarly by the strong composition over T iterations. □

B.2 Proof of DP-CompSGD with MiniBatch

Proof of Theorem 14. We start with the same procedure as in the derivation for the optimisation convergence rate of DP-CompGD in equation (46). Note that we can replace the batch gradient ∇F_S with the mini-batch gradient ∇F_{B_t} without affecting the derivation of the inequality

$$(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}\|^2 \leq \|\mathbf{w}_t - \mathbf{w}\|^2 + 2C_{m_t}\eta_t(F_{B_t}(\mathbf{w}) - F_{B_t}(\mathbf{w}_t)) + \eta_t^2(L^2 + m_t\sigma^2).$$

Rearranging the inequality and let $\mathbf{w} = \mathbf{w}^*$ we have:

$$2C_{m_t}\eta_t(F_{B_t}(\mathbf{w}_t) - F_{B_t}(\mathbf{w}^*)) \leq \|\mathbf{w}_t - \mathbf{w}^*\|^2 - (1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 + \eta_t^2(L^2 + m_t\sigma^2). \quad (108)$$

Since B_t is a random subset drawn uniformly from S, we have $\mathbb{E}_A[F_{B_t}(\mathbf{w})] = F_S(\mathbf{w})$. Hence by taking expectations and summing over T iterations, we have

$$2C_{m_t}\sum_{t=1}^{T}\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)] \leq \|\mathbf{w}_1-\mathbf{w}^*\|^2+\sum_{t=1}^{T}\beta_t\mathbb{E}[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2]+\sum_{t=1}^{T}\eta_t^2(L^2+m_t\sigma^2).$$

Choosing $\beta_t = 1/(t+1)$ we obtain that

$$\frac{\sum_{t=1}^{T}\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)]}{\sum_{t=1}^{T}\eta_t} = \mathcal{O}\left(\frac{\|\mathcal{C}\|^2+\log T\|\mathcal{C}\|^2+\sum_{t=1}^{T}\eta_t^2(L^2+m_t\sigma^2)}{\sum_{t=1}^{T}\eta_t}\right). \tag{109}$$

Finally, note that $m_T = \max_{t\in[T]} m_t$.

Let $\eta_t = \|\mathcal{C}\|/\sqrt{t(L^2+m_T\sigma^2)}$ and $\sigma^2 = \mathcal{O}(TL^2\log(1/\delta)\log(4TB/(\delta n)/(\epsilon^2n^2))$ we have

$$\begin{aligned}\frac{\sum_{t=1}^{T}\eta_t\mathbb{E}[F_S(\mathbf{w}_t)-F_S(\mathbf{w}^*)]}{\sum_{t=1}^{T}\eta_t} &= \mathcal{O}\left(\frac{\log T\|\mathcal{C}\|\sqrt{L^2+m_T\sigma^2}}{\sqrt{T}}\right)\\ &\leq \mathcal{O}\left(\frac{\log T\|\mathcal{C}\|L}{\sqrt{T}}+\frac{\log T\|\mathcal{C}\|L\sqrt{m_TT\log(1/\delta)\log(4TB/(\delta n)}}{n\epsilon\sqrt{T}}\right)\\ &= \mathcal{O}\left(\frac{\log T\|\mathcal{C}\|L}{\sqrt{T}}+\frac{\log T\|\mathcal{C}\|L\sqrt{m_T\log(1/\delta)\log(4TB/(\delta n)}}{n\epsilon}\right).\end{aligned}$$

The proof is completed. □

Proof of Theorem 15. Let S and S' be two neighbouring sample sets of size n that differ in one single sample. Let $G(\mathbf{w}_t) = \mathbf{w}_{t+1}$ denote the gradient update and let $G_1, \dots, G_T$ and $G'_1, \dots, G'_T$ be the updates induced by running the CompSGD on S and S' for T iterates, respectively. Let $\delta_T = \|\mathbf{w}_T - \mathbf{w}'_T\|$, by the Lipschitz condition we have

$$\mathbb{E}[|f(\mathbf{w}_T, z) - f(\mathbf{w}'_T, z)|] \leq L\mathbb{E}[\delta_T]. \tag{110}$$

If at iteration t, the mini-batch B_t, B'_t we selected is the same, i.e. $G_t = G'_t$, then by the same derivation for equation (58) we have the following bound

$$(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 \leq \|\mathbf{w}_t-\mathbf{w}'_t\|^2 - 2\eta_t\langle\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B_t}(\mathbf{w}'_t), \mathbf{w}_t-\mathbf{w}'_t\rangle + 4\eta_t^2L^2.$$

From the convexity of f we have that $\langle\nabla F_{B_t}(\mathbf{w}_t)-\nabla F_{B_t}(\mathbf{w}'_t), \mathbf{w}_t-\mathbf{w}'_t\rangle \geq 0$. Hence we obtain the following bound

$$(1-\beta_t)\|\mathbf{w}_{t+1}-\mathbf{w}'_{t+1}\|^2 \leq \|\mathbf{w}_t-\mathbf{w}'_t\|^2 + 4L^2\eta_t^2. \tag{111}$$

For the case where $G_t \neq G'_t$, we use the fact that B_t and B'_t differ by a single sample. Hence we have

$$
\begin{aligned}
&(1-\beta_t)\|\mathbf{w}_{t+1} - \mathbf{w}'_{t+1}\|^2 \\
&\qquad \leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 - 2\eta_t\langle \nabla F_{B_t}(\mathbf{w}_t) - \nabla F_{B'_t}(\mathbf{w}'_t), \mathbf{w}_t - \mathbf{w}'_t\rangle + 4\eta_t^2 L^2 \\
&\qquad \leq \|\mathbf{w}_t - \mathbf{w}'_t\|^2 + \frac{4L\eta_t}{b}\|\mathbf{w}_t - \mathbf{w}'_t\| + 4L^2\eta_t^2, \qquad (112)
\end{aligned}
$$

where the last inequality (112) follows by the same derivation as for equation (96), replacing S, S' with B_t, B'_t respectively. Combining the two cases we have

$$
\begin{aligned}
(1-\beta_t)\mathbb{E}[\delta_{t+1}^2] &\leq \left(1 - \frac{b}{n}\right)(\mathbb{E}[\delta_t^2] + 4L^2\eta_t^2) + \frac{b}{n}\left(\mathbb{E}[\delta_t^2] + \frac{4L\eta_t\mathbb{E}[\delta_t]}{b} + 4L^2\eta_t^2\right) \\
&= \mathbb{E}[\delta_t^2] + 4L\eta_t\left(\frac{\mathbb{E}[\delta_t]}{n} + L\eta_t\right). \qquad (113)
\end{aligned}
$$

The rest of the stability proof then follows by the same procedure as the proof for Thm. 13 starting from equation (97). The generalization result also follows directly by combining with Thm. 14 and Thm. 1 using the strategy discussed in section 2.2. □