Private Federated Learning with Provable Convergence via Smoothed Normalization

Egor Shulgin [1] **Sarit Khirirat** [1] **Peter Richtárik** [1]

Abstract

Federated learning enables training machine learning models while preserving the privacy of participants. Surprisingly, there is no differentially private (DP) distributed method for smooth, nonconvex optimization problems. The reason is that standard privacy techniques require bounding the participants' contributions, usually enforced via *clipping* of the updates. Existing literature typically ignores the effect of clipping by assuming the boundedness of gradient norms or analyzes distributed algorithms with clipping, but ignores DP constraints. In this work, we study an alternative approach via *smoothed normalization* of the updates, motivated by its favorable performance in the single-node setting. By integrating smoothed normalization with an Error Compensation mechanism, we design a new distributed algorithm α-NormEC. We prove that our method achieves a superior convergence rate over prior works. By extending α-NormEC to the DP setting, we obtain the first differentially private distributed optimization algorithm with provable convergence guarantees. Finally, our empirical results from neural network training indicate robust convergence of α-NormEC across different parameter settings.

1. Introduction

Federated Learning (FL) has become a viable approach for distributed collaborative training of machine learning models [37; 47; 48]. This growing interest has spurred the development of novel distributed optimization methods tailored for FL, focusing on ensuring high *communication efficiency* [30]. Although FL optimization methods ensure that private data is never directly transmitted, Boenisch et al. [6] demonstrated that the global models produced through FL can still enable the reconstruction of participants' data. Therefore, it is essential to study distributed optimization methods for *differentially private* training [17; 48; 69].

To mitigate emerging privacy risks in FL, differential privacy (DP) [17] has become the standard for providing theoretical privacy guarantees in machine learning. DP is often enforced by a clipping operator. It bounds gradient sensitivity, allowing the addition of DP noise to the updates before communication. While gradient clipping enables DP as in Differentially Private Stochastic Gradient Descent (DP-SGD) [1], it also introduces a bias that can impede convergence [11; 36]. Often, distributed DP gradient methods with clipping have been studied under assumptions that are unrealistic for heterogeneous FL environments, such as bounded gradient norms [42; 72; 45; 80], which effectively ignore the impact of clipping bias. To our knowledge, convergence guarantees for distributed DP methods remain elusive unless the impact of clipping bias is explicitly considered.

Error Feedback (EF), also known as Error Compensation (EC), such as EF21 [63] has been employed to alleviate the clipping bias and achieve strong convergence for nonprivate distributed methods with gradient clipping, as shown by [35; 77]. However, extending these methods to the private setting remains an open problem. Furthermore, optimizing the convergence of distributed DP clipping methods is challenging because the clipping threshold significantly influences both the convergence speed and DP noise variance. Extensive grid search for the optimal clipping threshold is computationally expensive [4] and leads to additional privacy loss [57]. Two major approaches have emerged to address the need to manually tune the clipping threshold. The first is to use adaptive clipping techniques, such as adaptive quantile clipping, initially proposed by [4] and further analyzed by [49; 66]. The second, which is the focus of this paper, is to replace clipping with a normalization operator.

Smoothed normalization, introduced by [9; 76], is an alternative operator to clipping. Unlike clipping, smoothed normalization eliminates the need to tune the clipping threshold. By ensuring that the Euclidean norm of the normalized

[1]King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. Correspondence to: Egor Shulgin <egor.shulgin@kaust.edu.sa>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025.

gradient is bounded above by one, smoothed normalization guarantees robust performance of DP-SGD in both convergence and privacy. However, very limited literature characterizes properties of smoothed normalization and a rigorous convergence analysis for DP-SGD using this operator, especially in the distributed setting. While the method has been studied in the single-node setting by [9] and [76], the convergence results rely on unrealistic and/or restrictive assumptions, such as symmetric gradient noise [9] and almost sure bounds on the gradient noise variance [76].

Contributions. We propose α-NormEC, the distributed gradient method that uses smoothed normalization and error compensation. Our method provides the first provable convergence guarantees in the DP setting without bounded gradient norm assumptions typically imposed in prior works. Our contributions are summarized below:

• **Favorable properties of smoothed normalization.** We present the novel properties of smoothed normalization. We show that smoothed normalization enjoys a "contractive" property similar to biased compression operators [5] widely used for reducing communication in distributed learning. This property essentially allows for analyzing α-NormEC without ignoring the impact of smoothed normalization.

• **Convergence for non-convex, smooth problems without bounded gradient norm assumptions.** We prove that α-NormEC achieves the optimal convergence rate [10] for minimizing non-convex, smooth functions without imposing additional restrictive assumptions, such as bounded gradient norms or bounded heterogeneity. Furthermore, α-NormEC achieves the faster rate than Clip21 [35], where its step size needs to know the inaccessible value of $f(x^0) - f^{\inf}$.

• **The first provable convergence in the private setting under standard assumptions.** Next, we extend α-NormEC to the differential privacy (DP) setting. Specifically, α-NormEC achieves the first convergence guarantees for DP, non-convex, smooth problems *without* ignoring the bias introduced by smoothed normalization. This is the first provably efficient distributed method in the DP setting under standard assumptions, thus addressing the gap left by prior work [35; 77], which did not adapt distributed gradient clipping methods for private training.

• **Robust empirical performance of α-NormEC.** Finally, we verify the theoretical benefits of α-NormEC in both non-private and private training via experiments on the image classification task with the CIFAR-10 dataset using the ResNet20 model. Our algorithm demonstrates robust empirical convergence across different parameter values and benefits from error compensation that enables superior performance over vanilla distributed gradient normalization methods (such as DP-SGD). In the private training, server normalization enhances the robustness of DP-α-NormEC across tuning parameters. Finally, DP-α-NormEC without server normalization outperforms DP-Clip21.

2. Preliminaries

Notations. We use $[a, b]$ to denote the set $\{a, a+1, a+2, \ldots, b\}$ for integers a, b such that $a \leq b$, and $\mathrm{E}[u]$ to represent the expectation of a random variable u. For vectors $x, y \in \mathbb{R}^d$, $\langle x, y\rangle$ denotes their inner product, and $\|x\| := \sqrt{\langle x, x\rangle}$ denotes the Euclidean norm of x. Finally, for functions $f, g : \mathbb{R}^d \to \mathbb{R}$, we write $f(x) = \mathcal{O}(g(x))$ if $f(x) \leq M \cdot g(x)$ for some $M > 0$. **Problem Formulation.** We consider the finite-sum optimization problem:

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}, \tag{1}$$

where $x \in \mathbb{R}^d$ is the vector of model parameters of dimension d, and $f_i : \mathbb{R}^d \to \mathbb{R}$ is either a loss function on client $i \in [1, n]$ (distributed setting) or data point i (single-node setting). Moreover, we impose the following standard assumption on objective functions for analyzing the convergence of first-order optimization algorithms [52].

Assumption 1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be bounded from below by a finite constant $f^{\inf}$, i.e. $f(x) \geq f^{\inf} > -\infty$ for all $x \in \mathbb{R}^d$, and be L-smooth, i.e. $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ for all $x, y \in \mathbb{R}^d$. Also, let f_i be L_i-smooth, i.e. $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|$ for all $x, y \in \mathbb{R}^d$.*

2.1. DP-SGD

To solve Problem (1), the most common approach that ensures the approximate (ϵ, δ)-differential privacy [16] is via the DP-SGD method [1]

$$x^{k+1} = x^k - \gamma \Big(\frac{1}{B} \sum_{i \in \mathcal{B}^k} \Psi(\nabla f_i(x^k)) + z^k \Big), \tag{2}$$

where $\gamma > 0$ is the step size, $\mathcal{B}^k$ is a subset of $\{1, 2, \ldots, n\}$ with cardinality $|\mathcal{B}^k| = B$, $z^k \in \mathbb{R}^d$ is the DP noise, and $\Psi : \mathbb{R}^d \to \mathbb{R}^d$ is an operator with bounded norm, i.e. $\|\Psi(g)\| \leq \Phi$ for some $\Phi > 0$ and any $g \in \mathbb{R}^d$. The method (2) achieves (ϵ, δ)-DP [1] if z^k is zero-mean Gaussian noise with variance

$$\sigma_{\mathrm{DP}}^2 \geq \Phi^2 \cdot \frac{cB^2}{n^2} \frac{K \log(1/\delta)}{\epsilon^2}, \tag{3}$$

where $c > 0$ is a constant, and $K > 0$ is the total number of iterations. To obtain reasonable DP guarantees, we usually choose $\epsilon \leq 10$ and $\delta \ll 1/n$, where n is the number of data points [59]. Notice that the DP Gaussian noise variance (3) is scaled with the *sensitivity* Φ.

The method (2) has been analyzed, e.g. by [80; 81; 51], under the bounded gradient norm assumption

$$\|\nabla f_i(x)\| \leq \Phi \quad \text{for all } i \text{ and } x \in \mathbb{R}^d. \tag{4}$$

However, this assumption has several limitations. Firstly, the sensitivity Φ is typically infeasible to compute for many loss functions used in training machine learning models. Even when it can be estimated, the resulting upper bound is often overly pessimistic, leading to excessively large DP noise and thus significantly degrading the algorithmic convergence performance. Secondly, this assumption restricts the class of models and loss functions f, as it excludes simple quadratic functions over unbounded domains. Thirdly, this assumption is "pathological" in the distributed setting because it restricts the heterogeneity between different clients and can result in vacuous bounds [33].

To enforce bounded sensitivity without imposing the bounded gradient norm Φ, it is common to apply clipping [1] with threshold $\tau > 0$, as defined by

$$\mathrm{Clip}_\tau(g) := \min(1, \tau/\|g\|)\, g. \tag{5}$$

Here, clipping ensures that $\Phi = \tau$, as $\|\Psi(g)\| = \|\mathrm{Clip}_\tau(g)\| \leq \tau = \Phi$. In fact, the method (2) that uses clipping (5) is generally referred to as DP-SGD in the literature. However, it is challenging to analyze the convergence of DP-SGD without additional restrictive assumptions such as the symmetric noise assumption [11; 60]. Even without DP noise, DP-SGD fails to converge due to the clipping bias [36]. Furthermore, since smaller values of τ imply stronger privacy but larger bias, jointly optimizing convergence and privacy of DP-SGD by carefully tuning τ and γ in the DP setting is a difficult task [39; 9].

Smoothed normalization. To eliminate the need to tune the clipping threshold τ, smoothed normalization is an alternative operator [9; 76] defined by

$$\mathrm{Norm}_\alpha(g) := \frac{1}{\alpha + \|g\|}\, g, \tag{6}$$

for some $\alpha \geq 0$ and satisfies the following property.

Lemma 1. *For any $\alpha \geq 0$, $\beta > 0$, and $g \in \mathbb{R}^d$,*

$$\|\mathrm{Norm}_\alpha(g)\| \leq 1, \tag{A}$$

$$\|g - \beta \mathrm{Norm}_\alpha(g)\|^2 = \left(1 - \frac{\beta}{\alpha + \|g\|}\right)^2 \|g\|^2. \tag{B}$$

Clearly, smoothed normalization ensures that (A) the norm of the normalized vector is bounded above by 1, and (B) the distance between the true vector and a β-multiple of the normalized vector is bounded by a function of β, α, and $\|g\|$. Although smoothed normalization with $\alpha = 0$ recovers standard normalization $g/\|g\|$ [53; 25; 40], smoothed normalization with $\alpha > 0$ improves the contraction factor, compared to standard normalization. Specifically, as $\|g\| \to 0$, the contraction factor of smoothed normalization approaches $(1 - \beta/\alpha)^2$, while standard normalization lacks this contraction property.

Although DP-SGD with smoothed normalization achieves robust empirical convergence in the DP setting [9], its theoretical convergence is limited to the single-node setting and relies on restrictive assumptions, specifically the central symmetry of stochastic gradients around the true gradient.

2.2. Limitations of DP Distributed Gradient Methods

Extending the convergence results of DP-SGD to the distributed setting poses significant challenges due to potential client heterogeneity. Existing results often address the bias introduced by the operator (clipping or normalization) by relying on restrictive assumptions, such as imposing bounded gradient norms [42; 81; 51; 73], or assuming that clipping is effectively turned off [79; 55]. Recently, Li et al. [41] extended the analysis of Koloskova et al. [36] to a distributed private setting under strong gradient dissimilarity condition. However, their method fails to converge due to the clipping bias, as discussed in the previous section. More importantly, even in the absence of the DP noise ($z^k = 0$), the inherent bias in the gradient estimator can severely impact the convergence. For instance, DP-SGD (2) diverge exponentially when $\Psi(\cdot)$ is a Top-1 compressor [5], and fail to converge when $\Psi(\cdot)$ is clipping [11; 35]. Similarly, smoothed normalization (6) with $\alpha = 0$ also cannot address this convergence issue, as demonstrated in the next example.

Example 1. *Consider Problem (1) with $n = 2, d = 1$, $f_1(x) = \frac{1}{2}(x-3)^2$ and $f_2(x) = \frac{1}{2}(x+3)^2$. Then, $f(x) = \frac{1}{2}(f_1(x) + f_2(x))$ satisfies Assumption 1 and is minimized at $x^\star = 0$. The iterates $\{x^k\}$ generated by (2) (for $B = 2$) with $z^k = 0$ and $\alpha = 0$ do not progress when $x^0 = 2$, as the gradient estimator $\mathrm{Norm}_\alpha\left(\nabla f_1(x^k)\right) + \mathrm{Norm}_\alpha\left(\nabla f_2(x^k)\right)$ results in*

$$\frac{\nabla f_1(x^0)}{\|\nabla f_1(x^0)\|} + \frac{\nabla f_2(x^0)}{\|\nabla f_2(x^0)\|} = -1/1 + 5/5 = 0.$$

Naively applying normalization to the gradients in DP-SGD leads to the method that does not converge in the distributed setting without extra assumptions. Also, this fundamental limitation affects federated averaging algorithms that apply normalization on the client updates [15].

2.3. EF21 Mechanism

One approach to resolve the convergence issues of distributed gradient methods with biased operators is to use EF21, an error feedback mechanism developed by Richtárik et al. [63]. Instead of directly applying the biased gradient estimator Ψ to the gradient, EF21 applies Ψ to the *difference* between the true gradient and the current error feedback (memory) vector. At iteration $k = 0, \ldots, K$, each client i receives the current iterate x^k from the central server, and

Operator	Property
Contractive compressor $\mathcal{C}:\mathbb{R}^d\to\mathbb{R}^d$	$\|\mathcal{C}(g)-g\|^2\le(1-\eta)^2\|g\|^2$
Clipping $\mathrm{Clip}_\tau(g):=\min\left(1,\frac{\tau}{\|g\|}\right)g$	$\|\mathrm{Clip}_\tau(g)-g\|^2\le\max(0,\|g\|-\tau)^2$
Smoothed normalization $\mathrm{Norm}_\alpha(g):=\frac{1}{\alpha+\|g\|}g$	$\|\mathrm{Norm}_\alpha(g)-g\|^2\le\left(1-\frac{1}{\alpha+\|g\|}\right)^2\|g\|^2$

Table 1: Smoothed normalization, unlike clipping, satisfies the contractive property similar to compressors.

computes its local update g_i^{k+1} via

$$g_i^{k+1}=g_i^k+\beta\Psi(\nabla f_i(x^k)-g_i^k), \tag{7}$$

where $\beta>0$. Next, the central server receives an average of local error feedback vectors that are communicated by all clients $\frac{1}{n}\sum_{i=1}^n\Psi(\nabla f_i(x^k)-g_i^k)$, computes the global gradient estimator $g^k:=\frac{1}{n}\sum_{i=1}^n g_i^k$ as

$$g^{k+1}=g^k+\frac{\beta}{n}\sum_{i=1}^n\Psi(\nabla f_i(x^k)-g_i^k), \tag{8}$$

and updates the next iterate x^{k+1} via

$$x^{k+1}=x^k-\gamma g^{k+1}. \tag{9}$$

This method generalized EF21 by replacing a contractive compressor[1] with other biased estimator, such as clipping in Clip21 proposed by Khirirat et al. [35].

Despite achieving the $\mathcal{O}(1/K)$ rate in the non-private setting, Clip21 faces difficulty in establishing the convergence in the presence of DP noise for two primary reasons. Firstly, its convergence analysis relies on separate descent inequalities when clipping turns on and off, as the operator does not satisfy the contractive compressor property required by EF21 (see Table 1). Secondly, the clipping threshold τ intricately influences both privacy and convergence. A sufficiently large τ is required to achieve the descent inequality, but this condition requires adding large Gaussian noise, which can prevent the convergence when it is accumulated. Due to these properties of clipping, analyzing the convergence of Clip21 in the DP setting is challenging.

3. Algorithm and Analysis

To address the convergence challenges of Clip21, we propose α-NormEC, the first distributed method to provide provable convergence guarantees in the DP setting. α-NormEC implements the update rules defined by (7), (8), and (9), where $\Psi(\cdot)$ is smoothed normalization (6) that offers key advantages over clipping. In the update rule in (9), we rather use server normalization $x^{k+1}=x^k-\gamma g^{k+1}/\left\|g^{k+1}\right\|$ and adopt notation $0/0=0$. See Algorithm 1 for the detailed description of α-NormEC.

[1] A contractive compressor [68; 5] is defined by $\|g-\mathcal{C}(g)\|^2\le(1-\eta)^2\|g\|^2$, for some $\eta\in(0,1]$ and for all $g\in\mathbb{R}^d$.

Algorithm 1 (DP-)α-NormEC

Input: Step size $\gamma>0$; $\beta>0$; normalization parameter $\alpha>0$; starting points $x^0,g_i^0\in\mathbb{R}^d$ for $i\in[1,n]$ and $\hat{g}^0=\frac{1}{n}\sum_{i=1}^n g_i^0$; $z_i^k\in\mathbb{R}^d$ are sampled from Gaussian distribution with zero mean and σ_{DP}^2-variance.
for each iteration $k=0,1,\ldots,K$ **do**
  for each client $i=1,2,\ldots,n$ in parallel **do**
    Compute local gradient $\nabla f_i(x^k)$
    Compute $\Delta_i^k=\mathrm{Norm}_\alpha\left(\nabla f_i(x^k)-g_i^k\right)$
    Update $g_i^{k+1}=g_i^k+\beta\Delta_i^k$
    Non-private setting: Transmit $\hat{\Delta}_i^k=\Delta_i^k$
    Private setting: Transmit $\hat{\Delta}_i^k=\Delta_i^k+z_i^k$
  end for
  Server computes $\hat{g}^{k+1}=\hat{g}^k+\frac{\beta}{n}\sum_{i=1}^n\hat{\Delta}_i^k$
  Server updates $x^{k+1}=x^k-\gamma\hat{g}^{k+1}/\left\|\hat{g}^{k+1}\right\|$
end for
Output: x^{K+1}

α-NormEC achieves better convergence guarantees than Clip21 in the non-private setting and the first convergence guarantees in the DP setting. These theoretical benefits of α-NormEC stem from favorable properties of smoothed normalization. Specifically, smoothed normalization, unlike clipping, behaves similarly to a contractive compressor (see Table 1), which simplifies the convergence analysis of α-NormEC compared to Clip21.

Next, we present the convergence theorem of α-NormEC.

Theorem 1. *Consider* DP-α-NormEC *(Algorithm 1) for solving Problem (1), where Assumption 1 holds. Let $\beta,\alpha,\gamma>0$ be chosen such that*

$$\frac{\beta}{\alpha+R}<1,\quad\text{and}\quad\gamma\le\frac{\beta R}{\alpha+R}\frac{1}{L_{\max}},$$

where $R=\max_{i\in[1,n]}\left\|\nabla f_i(x^0)-g_i^0\right\|$, and $L_{\max}=\max_{i\in[1,n]}L_i$. Then,

$$\min_{k\in[0,K]}\mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right]\le\frac{f(x^0)-f^{\inf}}{\gamma(K+1)}+2R+\frac{L}{2}\gamma+2\sqrt{\beta^2(K+1)\sigma_{\mathrm{DP}}^2/n}.$$

From Theorem 1, α-NormEC converges sublinearly up to

additive constants due to the bias of smoothed normalization $2R + \frac{L}{2}\gamma$ and the DP noise $2\sqrt{\beta^2(K+1)\sigma_{\mathrm{DP}}^2/n}$.

Non-private setting. We begin by discussing the convergence of α-NormEC from Theorem 1 in the non-private setting, when $\sigma_{\mathrm{DP}} = 0$. The next corollary shows that α-NormEC attains the $\mathcal{O}(1/\sqrt{K})$ rate when we properly choose initialized memory vectors g_i^0 and the step size γ.

Corollary 1 (Non-private setting)**.** *Consider α-NormEC (Algorithm 1) for solving Problem (1) under the same setting as Theorem 1 with $\sigma_{\mathrm{DP}} = 0$. If we choose $g_i^0 \in \mathbb{R}^d$ such that $\max_{i\in[1,n]} \left\|\nabla f_i(x^0) - g_i^0\right\| = D(K+1)^{-1/2}$ with any $D > 0$, $\gamma \leq \frac{\beta}{L_{\max}} \frac{D}{\alpha+D} \frac{1}{(K+1)^{1/2}}$, and $\alpha > \beta$, then*

$$\min_{k\in[0,K]} \left\|\nabla f(x^k)\right\| \leq \frac{C}{(K+1)^{1/2}},$$

where $C = \frac{L_{\max}(\alpha+D)}{\beta D}(f(x^0) - f^{\inf}) + 2D + \frac{L}{2}\frac{\beta D}{L_{\max}(\alpha+D)}$.

From Corollary 1, α-NormEC enjoys the $\mathcal{O}(1/\sqrt{K})$ rate in the gradient norm when we choose g_i^0 such that $R = \mathcal{O}(1/\sqrt{K})$, and $\gamma = \mathcal{O}(\beta/\sqrt{K})$. By further choosing $\alpha > 1$, and proper choice of β, the associated convergence bound from Corollary 1 becomes

$$\min_{k\in[0,K]} \left\|\nabla f(x^k)\right\| \leq \frac{\sqrt{2L(f(x^0) - f^{\inf})} + 2D}{(K+1)^{1/2}}. \quad (10)$$

This bound comprises two terms. The first term $\sqrt{2L(f(x^0) - f^{\inf})}(K+1)^{-1/2}$ is the convergence bound by classical gradient descent [10], while the second term $2D(K+1)^{-1/2}$ due to the initialized memory vectors g_i^0 for running the error feedback mechanism. We can initialize $x^0, g_i^0 \in \mathbb{R}^d$ to ensure that $\max_{i\in[1,n]} \left\|\nabla f_i(x^0) - g_i^0\right\| = D(K+1)^{-1/2}$ with any $D > 0$. For example, this condition holds when we choose $g_i^0 = \nabla f_i(x^0) + e$ for any $x^0 \in \mathbb{R}^d$ and $e = (D/\sqrt{K+1}, 0, \ldots, 0) \in \mathbb{R}^d$ with any $D > 0$ and any total iteration count K.

Comparison between α-NormEC and Clip21. In the non-private setting, α-NormEC provides stronger convergence guarantees than Clip21. In particular, the convergence bound of α-NormEC in (10) exhibits a smaller convergence factor than that of Clip21, as explained in Appendix E.3. Specifically, by choosing $g_i^0 \in \mathbb{R}^d$ such that D is sufficiently small, the convergence bound of α-NormEC in (10) approaches that of classical gradient descent [10]. Furthermore, the hyperparameters of α-NormEC ($\beta, \alpha, \gamma > 0$), as defined in Theorem 2, are easier to implement than Clip21. The step-size γ for α-NormEC does not need to know the function sub-optimal gap $f(x^0) - f(x^\star)$ that is inaccessible in practice, in contrast to Clip21, whose step-size in (Theorem 5.6 of [34]) depends on not only the function sub-optimality gap but also $C_1 = \max_{i\in[1,n]} \left\|\nabla f_i(x^0)\right\|$.

Private setting. Next, we discuss the convergence of α-NormEC in the DP setting. From Theorem 1, we show that α-NormEC achieves (ϵ, δ)-DP and the corresponding utility bounds, in contrast to Clip21 [35] where its convergence is limited to the non-private setting. We show this by setting the standard deviation of the DP noise according to Theorem 1 of [1], i.e., $\sigma_{\mathrm{DP}} = \mathcal{O}(\sqrt{(K+1)\log(1/\delta)}\epsilon^{-1})$, which yields the utility bound $\mathcal{O}\left(\Delta \sqrt[4]{d\log(1/\delta)/(n\epsilon^2)}\right)$ with constant $\Delta > 0$ defined in Corollary 2. Unlike Clip21, α-NormEC provides the first utility bound in the DP distributed setting that accounts for the effect of bounding sensitivity, a factor often neglected in the existing literature. Our obtained utility bound applies for smooth problems without the bounded gradient norm assumption, the limitation present in prior works for analyzing DP-SGD such as [42; 72; 45; 80].

4. Experiments

We evaluate the performance of α-NormEC to solve the non-convex optimization problem of deep neural network training in both non-private and private settings. We conduct experiments on the CIFAR-10 [38] dataset using the ResNet20 [26] model for the image classification task. The compared methods are run for 300 communication rounds. The convergence plots present results for tuned step size γ. Additional experimental details and results are provided in Appendix G.

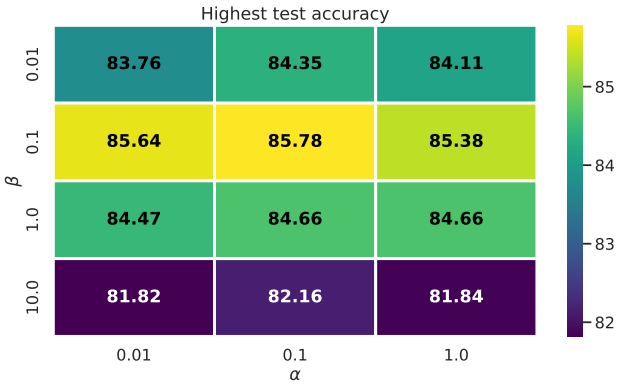

Figure 1: The highest test accuracy achieved by α-NormEC with different α and β values.

α-NormEC demonstrates stable convergence with respect to the normalization parameter α, and robustness to variations in β values. From Figure 1, we observe that convergence of α-NormEC is stable with respect to a wide range of α values and robust to variations in β. The performance of α-NormEC is primarily governed by the choice of β. From Figure 1, optimal performance (85-86% accuracy) is observed when β is around 0.1. While α-NormEC is stable with respect to α, extreme values of β lead to suboptimal performance: very large values ($\beta = 10.0$) result in lower accuracy (81-82%), while very small values

($\beta = 0.01$) achieve moderate performance (83-84%). The optimal configuration, achieving the highest 85.78% accuracy, is $\beta = 0.1$ and $\alpha = 0.1$. Further analysis of the algorithm's stability with respect to α and robustness to β, including additional metrics (along with convergence curves) is provided in Appendix G.2.1. For subsequent experiments, we set $\alpha = 0.01$, which is consistent with the prior work in the single-node setting [9].

Error compensation enables α-NormEC to outperform DP-SGD. From Figure 2, α-NormEC outperforms DP-SGD with smoothed normalization (defined by Equation (2) with $B = n$ and $z \equiv 0$). This improvement is attributed to error compensation (EC), as confirmed by running α-NormEC without server normalization (Line 11 of Algorithm 1). From Figure 2, α-NormEC achieves faster convergence than DP-SGD with smoothed normalization for most β values, with the exception of $\beta = 10$. However, such a large β is impractical for differentially private training due to the resulting increase in noise variance. Moreover, while our algorithm demonstrates robust performance across varying β values, DP-SGD with smoothed normalization exhibits greater sensitivity to this parameter, notably struggling with convergence at $\beta = 0.01$. This comparison underscores how EC not only accelerates convergence but also improves the algorithm's stability with respect to parameter selection. Appendix G.2.2 presents further details (such as accuracy convergence curves in Figure 6) and optimal parameters with corresponding final accuracies (Table 7).

An ablation study examining the impact of server normalization on α-NormEC is provided in Appendix G.2.3. Furthermore, a comparison between α-NormEC and Clip21 is presented in Appendix G.2.4.

Private Training. We analyze the performance of α-NormEC in the private setting by choosing the variance of added noise at $\beta\sqrt{K\log(1/\delta)}\epsilon^{-1}$ for $\epsilon = 8, \delta = 10^{-5}$ and vary β to simulate different privacy levels. Figure 3 shows the training loss curves for DP-α-NormEC (with and without server normalization) and DP-Clip21. Notably, compared to the non-private case, convergence in the DP setting is slower and requires a smaller β (e.g., 0.01) for best performance.

From Figure 3 we observe three key findings: **(1)** DP-α-NormEC without Server Normalization converges significantly faster than DP-Clip21 at all privacy levels ($\beta \in \{0.001, 0.01, 0.1\}$); **(2)** Server normalization (SN) provides crucial stability at high noise levels–at $\beta = 1.0$, only DP-α-NormEC with SN maintains convergence; **(3)** While SN improves robustness, it comes with a slight reduction in convergence speed at lower noise levels.

The complete analysis, including test accuracy results across different hyperparameter combinations and detailed performance comparisons, is presented in Appendix G.3. Notably, server normalization significantly reduces performance variation across different learning rates (γ), with at most 6% variation compared to 40% without normalization, demonstrating improved hyperparameter robustness.

5. Conclusion

We have proposed and analyzed α-NormEC, a novel distributed algorithm that integrates smoothed normalization with the EF21 mechanism for solving non-convex, smooth optimization problems in both non-private and private settings. Unlike Clip21, α-NormEC achieves strong convergence guarantees that almost match those of classical gra-

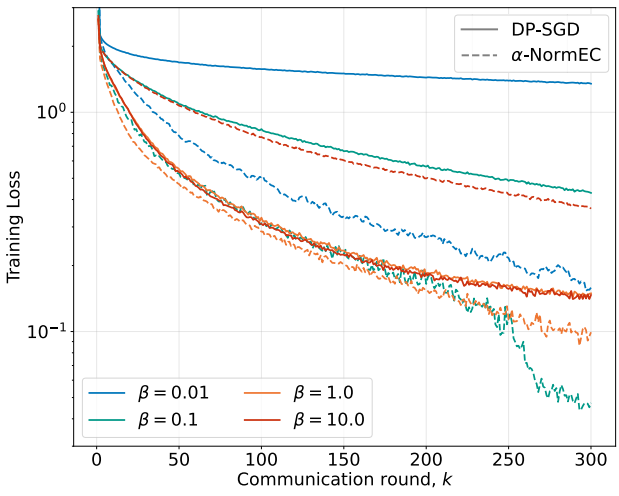

Figure 2: Comparison of DP-SGD (2) [solid] and α-NormEC (1) [dashed] without server normalization.

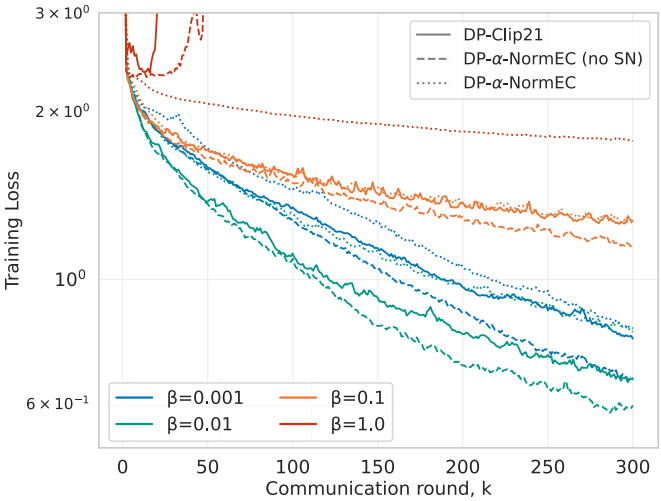

Figure 3: Comparison of methods in the Differentially Private (DP) setting across different β values.

dient descent for non-private training and provides the first utility bound for private training without relying on restrictive assumptions such as bounded gradient norms. In neural network training, α-NormEC achieves robust convergence across varying hyperparameters and significantly stronger convergence (due to error compensation) compared to DP-SGD with smoothed normalization. In the private training, DP-α-NormEC benefits from server normalization for increased robustness and outperforms DP-Clip21.

Our work implies many promising research directions. One direction is to extend α-NormEC to accommodate the partial participation case, where the central server receives the local normalized gradients from a few clients, and the stochastic case, where each client has access only to stochastic gradients. Another important direction is to modify α-NormEC to solve federated learning problems, where the clients run their local updates before the local updates are normalized and transmitted to the central server.

Acknowledgements

We would like to thank Rustem Islamov for sharing the code for experiments.

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) Center of Excellence for Generative AI, under award number 5940, iii) SDAIA-KAUST Center of Excellence in Artificial Intelligence and Data Science.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016. (Cited on pages 1, 2, 3, 5, and 13)

[2] Alber, Y. I., Iusem, A. N., and Solodov, M. V. On the projected subgradient method for nonsmooth convex optimization in a hilbert space. *Mathematical Programming*, 81:23–35, 1998. (Cited on page 13)

[3] Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018. (Cited on page 13)

[4] Andrew, G., Thakkar, O., McMahan, H. B., and Ramaswamy, S. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12191–12203, 2021. (Cited on page 1)

[5] Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24 (276):1–50, 2023. (Cited on pages 2, 3, and 4)

[6] Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A. S., Shumailov, I., and Papernot, N. When the curious abandon honesty: Federated learning is not private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 175–199. IEEE, 2023. (Cited on page 1)

[7] Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In *International conference on machine learning*, pp. 1059–1071. PMLR, 2021. (Cited on page 13)

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. (Cited on page 13)

[9] Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Automatic clipping: Differentially private deep learning made easier and stronger. *Advances in Neural Information Processing Systems*, 36, 2024. (Cited on pages 1, 2, 3, 6, and 13)

[10] Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020. (Cited on pages 2 and 5)

[11] Chen, X., Wu, S. Z., and Hong, M. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020. (Cited on pages 1, 3, and 13)

[12] Chezhegov, S., Klyukin, Y., Semenov, A., Beznosikov, A., Gasnikov, A., Horváth, S., Takáč, M., and Gorbunov, E. Gradient clipping improves AdaGrad when the noise is heavy-tailed. *arXiv preprint arXiv:2406.04443*, 2024. (Cited on page 13)

[13] Crawshaw, M., Bao, Y., and Liu, M. EPISODE: Episodic gradient clipping with periodic resampled corrections for federated learning with heterogeneous data. In *The Eleventh International Conference on Learning Representations*, 2023. (Cited on page 13)

[14] Danilova, M. and Gorbunov, E. Distributed methods with absolute compression and error compensation. In

International Conference on Mathematical Optimization Theory and Operations Research, pp. 163–177. Springer, 2022. (Cited on page 13)

[15] Das, R., Hashemi, A., sujay sanghavi, and Dhillon, I. S. Differentially private federated learning with normalized updates. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022. (Cited on pages 3 and 13)

[16] Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006. (Cited on page 2)

[17] Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. (Cited on page 1)

[18] Ermoliev, Y. Stochastic quasigradient methods. numerical techniques for stochastic optimization. *Springer Series in Computational Mathematics*, (10):141–185, 1988. (Cited on page 13)

[19] Fatkhullin, I., Tyurin, A., and Richtárik, P. Momentum provably improves error feedback! *Advances in Neural Information Processing Systems*, 36, 2024. (Cited on page 13)

[20] Gao, Y., Islamov, R., and Stich, S. U. EControl: Fast distributed optimization with compression and error control. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 13)

[21] Gorbunov, E., Danilova, M., and Gasnikov, A. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *Advances in Neural Information Processing Systems*, 33:15042–15053, 2020. (Cited on page 13)

[22] Gorbunov, E., Kovalev, D., Makarenko, D., and Richtárik, P. Linearly converging error compensated SGD. *Advances in Neural Information Processing Systems*, 33:20889–20900, 2020. (Cited on page 13)

[23] Gorbunov, E., Sadiev, A., Danilova, M., Horváth, S., Gidel, G., Dvurechensky, P., Gasnikov, A., and Richtárik, P. High-probability convergence for composite and distributed stochastic minimization and variational inequalities with heavy-tailed noise. In *Forty-first International Conference on Machine Learning*, 2024. (Cited on page 13)

[24] Gorbunov, E., Tupitsa, N., Choudhury, S., Aliev, A., Richtárik, P., Horváth, S., and Takáč, M. Methods for convex (L_0, L_1)-smooth optimization: Clipping, acceleration, and adaptivity. In *The Thirteenth International Conference on Learning Representations*, 2025. (Cited on page 13)

[25] Hazan, E., Levy, K., and Shalev-Shwartz, S. Beyond convexity: Stochastic quasi-convex optimization. *Advances in neural information processing systems*, 28, 2015. (Cited on page 3)

[26] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. (Cited on page 5)

[27] Hübler, F., Yang, J., Li, X., and He, N. Parameter-agnostic optimization under relaxed smoothness. In *International Conference on Artificial Intelligence and Statistics*, pp. 4861–4869. PMLR, 2024. (Cited on page 13)

[28] Hübler, F., Fatkhullin, I., and He, N. From gradient clipping to normalization for heavy tailed SGD. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. (Cited on page 13)

[29] Idelbayev, Y. Proper ResNet implementation for CIFAR10/CIFAR100 in PyTorch. https://github.com/akamaster/pytorch_resnet_cifar10. Accessed: 2024-12-31. (Cited on page 22)

[30] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083. (Cited on page 1)

[31] Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. Error feedback fixes signSGD and other gradient compression schemes. In *International Conference*

on Machine Learning, pp. 3252–3261. PMLR, 2019. (Cited on page 13)

[32] Karimireddy, S. P., He, L., and Jaggi, M. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, pp. 5311–5319. PMLR, 2021. (Cited on page 13)

[33] Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020. (Cited on page 3)

[34] Khirirat, S., Magnússon, S., and Johansson, M. Convergence bounds for compressed gradient methods with memory based error compensation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2857–2861. IEEE, 2019. (Cited on pages 5 and 13)

[35] Khirirat, S., Gorbunov, E., Horváth, S., Islamov, R., Karray, F., and Richtárik, P. Clip21: Error feedback for gradient clipping. *arXiv preprint arXiv:2305.18929*, 2023. (Cited on pages 1, 2, 3, 4, 5, 13, 14, and 16)

[36] Koloskova, A., Hendrikx, H., and Stich, S. U. Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *International Conference on Machine Learning*, pp. 17343–17363. PMLR, 2023. (Cited on pages 1, 3, and 13)

[37] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *NIPS Private Multi-Party Machine Learning Workshop*, 2016. (Cited on page 1)

[38] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, 2009. (Cited on page 5)

[39] Kurakin, A., Song, S., Chien, S., Geambasu, R., Terzis, A., and Thakurta, A. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022. (Cited on page 3)

[40] Levy, K. Y. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016. (Cited on page 3)

[41] Li, B., Jiang, X., Schmidt, M. N., Alstrøm, T. S., and Stich, S. U. An improved analysis of per-sample and per-update clipping in federated learning. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on pages 3 and 13)

[42] Li, Z., Zhao, H., Li, B., and Chi, Y. SoteriaFL: A unified framework for private federated learning with communication compression. *Advances in Neural Information Processing Systems*, 35:4285–4300, 2022. (Cited on pages 1, 3, 5, and 13)

[43] Liu, M., Zhuang, Z., Lei, Y., and Liao, C. A communication-efficient distributed gradient clipping algorithm for training deep neural networks. *Advances in Neural Information Processing Systems*, 35:26204–26217, 2022. (Cited on page 13)

[44] Lobanov, A., Gasnikov, A., Gorbunov, E., and Takác, M. Linear convergence rate in convex setup is possible! gradient descent method variants under (L_0, L_1)-smoothness. *arXiv preprint arXiv:2412.17050*, 2024. (Cited on page 13)

[45] Lowy, A., Ghafelebashi, A., and Razaviyayn, M. Private non-convex federated learning without a trusted server. In *International Conference on Artificial Intelligence and Statistics*, pp. 5749–5786. PMLR, 2023. (Cited on pages 1, 5, and 13)

[46] Malinovsky, G., Gorbunov, E., Horváth, S., and Richtárik, P. Byzantine robustness and partial participation can be achieved simultaneously: Just clip gradient differences. In *Privacy Regulation and Protection in Machine Learning*, 2023. (Cited on page 13)

[47] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017. (Cited on page 1)

[48] McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. (Cited on pages 1 and 13)

[49] Merad, I. and Gaïffas, S. Robust stochastic optimization via gradient quantile clipping. *Transactions on Machine Learning Research*, 2024. (Cited on page 1)

[50] Merity, S., Keskar, N. S., and Socher, R. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2018. (Cited on page 13)

[51] Murata, T. and Suzuki, T. Diff2: Differential private optimization via gradient differences for nonconvex distributed learning. In *International Conference on Machine Learning*, pp. 25523–25548. PMLR, 2023. (Cited on pages 2, 3, and 13)

[52] Nesterov, Y. et al. *Lectures on convex optimization*, volume 137. Springer, 2018. (Cited on page 2)

[53] Nesterov, Y. E. Minimization methods for nonsmooth convex and quasiconvex functions. *Matekon*, 29(3): 519–531, 1984. (Cited on page 3)

[54] Nguyen, T. D., Nguyen, T. H., Ene, A., and Nguyen, H. Improved convergence in high probability of clipped gradient methods with heavy tailed noise. *Advances in Neural Information Processing Systems*, 36:24191–24222, 2023. (Cited on page 13)

[55] Noble, M., Bellet, A., and Dieuleveut, A. Differentially private federated learning on heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 10110–10145. PMLR, 2022. (Cited on pages 3 and 13)

[56] Özfatura, K., Özfatura, E., Küpçü, A., and Gunduz, D. Byzantines can also learn from history: Fall of centered clipping in federated learning. *IEEE Transactions on Information Forensics and Security*, 19: 2010–2022, 2023. (Cited on page 13)

[57] Papernot, N. and Steinke, T. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022. (Cited on page 1)

[58] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1310–1318. PMLR, 2013. (Cited on page 13)

[59] Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H. B., Vassilvitskii, S., Chien, S., and Thakurta, A. G. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023. (Cited on page 2)

[60] Qian, J., Wu, Y., Zhuang, B., Wang, S., and Xiao, J. Understanding gradient clipping in incremental gradient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 1504–1512. PMLR, 2021. (Cited on page 3)

[61] Qian, X., Richtárik, P., and Zhang, T. Error compensated distributed SGD can be accelerated. *Advances in Neural Information Processing Systems*, 34:30401–30413, 2021. (Cited on page 13)

[62] Qian, X., Dong, H., Zhang, T., and Richtarik, P. Catalyst acceleration of error compensated methods leads to better communication complexity. In *International Conference on Artificial Intelligence and Statistics*, pp. 615–649. PMLR, 2023. (Cited on page 13)

[63] Richtárik, P., Sokolov, I., and Fatkhullin, I. EF21: a new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34:4384–4396, 2021. (Cited on pages 1, 3, and 13)

[64] Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, volume 2014, pp. 1058–1062. Singapore, 2014. (Cited on page 13)

[65] Shor, N. Z. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012. (Cited on page 13)

[66] Shulgin, E. and Richtárik, P. On the convergence of DP-SGD with adaptive clipping. *arXiv preprint arXiv:2412.19916*, 2024. (Cited on page 1)

[67] Stich, S. U. and Karimireddy, S. P. The error-feedback framework: SGD with delayed gradients. *Journal of Machine Learning Research*, 21(237):1–36, 2020. (Cited on page 13)

[68] Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified SGD with memory. *Advances in neural information processing systems*, 31, 2018. (Cited on pages 4 and 13)

[69] Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. (Cited on page 1)

[70] Tang, H., Yu, C., Lian, X., Zhang, T., and Liu, J. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pp. 6155–6165. PMLR, 2019. (Cited on page 13)

[71] Vankov, D., Rodomanov, A., Nedich, A., Sankar, L., and Stich, S. U. Optimizing (l_0, l_1)-smooth functions by gradient methods. In *The Thirteenth International Conference on Learning Representations*, 2025. (Cited on page 13)

[72] Wang, L., Jayaraman, B., Evans, D., and Gu, Q. Efficient privacy-preserving stochastic nonconvex optimization. In Evans, R. J. and Shpitser, I. (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 2203–2213. PMLR, 31 Jul–04 Aug 2023. (Cited on pages 1, 5, and 13)

[73] Wang, L., Zhou, X., Patel, K. K., Tang, L., and Saha, A. Efficient private federated non-convex optimization with shuffled model. In *Privacy Regulation and Protection in Machine Learning*, 2024. (Cited on pages 3 and 13)

[74] Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q., and Poor, H. V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020. (Cited on page 13)

[75] Wu, J., Huang, W., Huang, J., and Zhang, T. Error compensated quantized SGD and its applications to large-scale distributed optimization. In *International conference on machine learning*, pp. 5325–5333. PMLR, 2018. (Cited on page 13)

[76] Yang, X., Zhang, H., Chen, W., and Liu, T.-Y. Normalized/clipped SGD with perturbation for differentially private non-convex optimization. *arXiv preprint arXiv:2206.13033*, 2022. (Cited on pages 1, 2, 3, and 13)

[77] Yu, S., Jakovetic, D., and Kar, S. Smoothed gradient clipping and error feedback for distributed optimization under heavy-tailed noise. *arXiv preprint arXiv:2310.16920*, 2023. (Cited on pages 1, 2, and 13)

[78] Zhang, J., He, T., Sra, S., and Jadbabaie, A. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. (Cited on page 13)

[79] Zhang, M., Xie, Z., and Yin, L. Private and communication-efficient federated learning based on differentially private sketches. *arXiv preprint arXiv:2410.05733*, 2024. (Cited on pages 3 and 13)

[80] Zhang, X., Fang, M., Liu, J., and Zhu, Z. Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed SGD approach. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 261–270, 2020. (Cited on pages 1, 2, 5, and 13)

[81] Zhang, X., Chen, X., Hong, M., Wu, Z. S., and Yi, J. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning, ICML 2022*, 2022. (Cited on pages 2, 3, and 13)

Contents

A. Related Work

In this section, we review prior work that is closely related to our paper.

Clipping and normalization. Clipping and normalization address many key challenges in machine learning. They mitigate the problem of exploding gradients in recurrent neural networks [58], enhance neural network training for tasks in natural language processing [50; 8] and computer vision [7], enable differentially private machine learning [1; 48], and provide robustness in the presence of misbehaving or adversarial workers [32; 56; 46]. In this paper, we consider smoothed normalization, introduced by Yang et al. [76]; Bu et al. [9], as an alternative to clipping, given its robust empirical performance and hyperparameter tuning benefits in the DP setting.

Private optimization methods. DP-SGD [1] is the standard distributed first-order method that achieves the DP guarantee by clipping the gradient before adding noise scaled with the clipped gradient's sensitivity. However, existing DP-SGD convergence analyses often neglect the clipping bias. Specifically, convergence results for smooth functions under differential privacy often require either the assumption of bounded gradient norms [80; 42; 81; 72; 45; 51; 73] or conditions where clipping is effectively inactive [79; 55]. Thus, the convergence behavior of DP-SGD in the presence of clipping bias remains poorly understood.

Single-node non-private methods with clipping. The impact of clipping bias has been extensively studied in single-node gradient methods for non-private optimization. Numerous works have shown strong convergence guarantees of clipped gradient methods under various conditions, including nonsmooth, rapidly growing convex functions [65; 18; 2], generalized smoothness [78; 36; 24; 71; 44; 27], and heavy-tailed noise [21; 54; 23; 28; 12].

Distributed non-private methods with clipping. Applying gradient clipping in the distributed setting is challenging. Existing convergence analyses often rely on bounded heterogeneity assumptions, which often do not hold in cases of arbitrary diverse clients. For example, federated optimization methods with clipping have been analyzed under the bounded difference between the local and global gradients [74; 43; 13; 41]. However, even in the non-private setting, these distributed clipping methods do not converge for simple problems [11; 35] for arbitrary clipping threshold. To address the convergence issue, one approach is to use error feedback mechanisms, such as EF21 [63], that are employed by [35; 77] to compute local gradient estimators and alleviate clipping bias. However, these distributed clipping methods using error feedback are limited to the non-private setting, and extending them to the DP setting is still an open problem. In this paper, we propose a distributed method that replaces clipping with smoothed normalization in the EF21 mechanism. Our method provides the first provable convergence in the DP setting and empirically outperforms the distributed version of DP-SGD with smoothed normalization [9; 76], a special case of [15].

Error feedback. Error feedback, or error compensation, has been applied to improve the convergence of distributed methods with gradient compression for communication-efficient learning. First introduced by [64], EF14 was extensively analyzed for first-order methods in both single-node [68; 31; 67; 34] and distributed settings [75; 3; 22; 61; 70; 14; 62]. Another error feedback variant is EF21 proposed by [63] that ensures strong convergence under any contractive compression operator for non-convex, smooth problems. Recent variants, e.g. EF21-SGD2M [19] and EControl [20], have been developed to obtain the lower iteration and communication complexities than EF21 for stochastic optimization.

B. Basic Facts

For $n \in \mathbb{N}$ and $x_1, \ldots, x_n, x, y \in \mathbb{R}^d$,

$$\langle x, y \rangle \leq \|x\| \|y\|, \tag{11}$$

$$\|x + y\| \leq \|x\| + \|y\|, \quad \text{and} \tag{12}$$

$$\left\| \frac{1}{n} \sum_{i=1}^{n} x_i \right\| \leq \frac{1}{n} \sum_{i=1}^{n} \|x_i\|. \tag{13}$$

C. Proof of Lemma 1

We prove the first statement by taking the Euclidean norm. Next, we prove the second statement. From the definition of the Euclidean norm,

$$\begin{aligned}\|g-\beta\mathrm{Norm}_\alpha(g)\|^2 &\overset{(6)}{=} \|g\|^2+\frac{\beta^2}{(\alpha+\|g\|)^2}\|g\|^2-2\beta\frac{\|g\|^2}{\alpha+\|g\|}\\ &= \left(1-\frac{\beta}{\alpha+\|g\|}\right)^2\|g\|^2.\end{aligned}$$

D. Comparison of EF21 with Clipping and Smoothed Normalization

We compare the modified EF21 mechanism, where a contractive compressor is replaced with clipping in Clip21 and with smoothed normalization in α-NormEC. To compare these modified updates, given the optimal vector $g^\star\in\mathbb{R}^d$, we consider the single-node EF21 mechanism, which computes the memory vector $g^k\in\mathbb{R}^d$ according to

$$g^{k+1}=g^k+\Psi(g^\star-g^k), \tag{14}$$

where $\Psi:\mathbb{R}^d\to\mathbb{R}^d$ is the biased gradient estimator, and $g^0\in\mathbb{R}^d$ is the initial memory vector.

If $\Psi(g)=\mathrm{Clip}_\tau(g)$, then from Theorem 4.3 of [35]

$$\left\|g^k-g^\star\right\|\le\max(0,\left\|g^0-g^\star\right\|-k\tau).$$

If $\Psi(g)=\mathrm{Norm}_\alpha(g)$, then from Lemma 1

$$\begin{aligned}\left\|g^\star-g^k\right\|^2 &= \left\|g^\star-g^{k-1}-\beta\mathrm{Norm}_\alpha\left(g^\star-g^{k-1}\right)\right\|^2\\ &= \left(1-\frac{\beta}{\alpha+\|g^\star-g^{k-1}\|}\right)^2\left\|g^\star-g^{k-1}\right\|^2\\ &\vdots\\ &= \left\|g^\star-g^0\right\|^2\cdot\prod_{l=1}^k\left(1-\frac{\beta}{\alpha+\|g^\star-g^{l-1}\|}\right)^2.\end{aligned}$$

In conclusion, while the EF21 mechanism with clipping ensures that the memory g^k will reach $g^\star$ within a finite number of iterations k (when $k\ge\left\|g^0-g^\star\right\|/\tau$), the EF21 mechanism with smoothed normalization guarantees that g^k will eventually reach $g^\star$ (provided that$\beta/\alpha<1$).

E. Non-private Results

Theorem 2 (Non-private setting)**.** *Consider α-NormEC (Algorithm 1) for solving Problem (1), where Assumption 1 holds. Let $\beta,\alpha,\gamma>0$ be chosen such that*

$$\frac{\beta}{\alpha+R}<1,\quad \textit{and}\quad \gamma\le\frac{\beta R}{\alpha+R}\frac{1}{L_{\max}},$$

where $R=\max_{i\in[1,n]}\left\|\nabla f_i(x^0)-g_i^0\right\|$ and $L_{\max}=\max_{i\in[1,n]}L_i$. Then,

$$\min_{k\in[0,K]}\left\|\nabla f(x^k)\right\|\le\frac{f(x^0)-f^{\inf}}{\gamma(K+1)}+2R+\frac{L}{2}\gamma.$$

E.1. Proof of Theorem 2

Proof outline. By the L-smoothness of the objective function f, and by the update for x^{k+1} in α-NormEC, we obtain

$$V^{k+1}\le V^k-\gamma\left\|\nabla f(x^k)\right\|+\frac{L\gamma^2}{2}+2\gamma W^k,$$

where $V^k := f(x^k) - f^{\inf}$, and $W^k := \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$. The key step to establish the convergence is to bound $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$. Using Lemma 2 and appropriate choices of the tuning parameters β, α, and γ, we get

$$\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| \leq \max_{i\in[1,n]} \left\|\nabla f_i(x^0) - g_i^0\right\|, \quad \forall k \geq 0.$$

Finally, substituting this bound into the previous descent inequality yields the convergence bound in $\min_{k\in[0,K]} \left\|\nabla f(x^k)\right\|$. Deriving the bound on $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$ for α-NormEC by induction is similar to but simpler than Clip21. This simplified proof is possible because smoothed normalization possesses a contractive property similar to the contractive compressor used in EF21.

We prove Theorem 2 by Lemma 2, which states $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$ for some positive scalars R, given that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$, and hyperparameters α, β, γ are properly tuned.

Lemma 2 (Non-private setting). *Consider α-NormEC (Algorithm 1) for solving Problem (1), where Assumption 1 holds. If $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$, $\frac{\beta}{\alpha+R} < 1$, and $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$ with $L_{\max} = \max_{i\in[1,n]} L_i$, then $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$.*

Proof. From the definition of the Euclidean norm,

$$\begin{aligned}
\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| &\overset{(12)}{\leq} \left\|\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right\| + \left\|\nabla f_i(x^k) - g_i^{k+1}\right\| \\
&\overset{g_i^{k+1}}{=} \left\|\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right\| \\
&\quad + \left\|\nabla f_i(x^k) - g_i^k - \beta \mathrm{Norm}_\alpha\left(\nabla f_i(x^k) - g_i^k\right)\right\| \\
&\overset{\text{Lemma 1}}{\leq} \left\|\nabla f_i(x^{k+1}) - \nabla f_i(x^k)\right\| \\
&\quad + \left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right| \left\|\nabla f_i(x^k) - g_i^k\right\| \\
&\overset{\text{Assumption 1, and } x^{k+1}}{\leq} L_{\max}\gamma + \left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right| \left\|\nabla f_i(x^k) - g_i^k\right\|.
\end{aligned}$$

If $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$ and $\frac{\beta}{\alpha+R} < 1$, then $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$ when

$$\gamma \leq \frac{\beta R}{\alpha + R}\frac{1}{L_{\max}}.$$

□

Now, we are ready to prove the result in Theorem 2 in four steps.

Step 1) Prove by induction that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$ for $R = \max_{i\in[1,n]} \left\|\nabla f_i(x^0) - g_i^0\right\|$. For $k = 0$, this is obvious. Next, let $\left\|\nabla f_i(x^l) - g_i^l\right\| \leq R$ for $R = \max_{i\in[1,n]} \left\|\nabla f_i(x^0) - g_i^0\right\|$ for $l = 0, 1, \ldots, k$. Then, if $\beta/(\alpha + R) < 1$, and $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$, then from Lemma 2 $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$.

Step 2) Bound $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$. From the definition of the Euclidean norm,

$$\begin{aligned}
\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| &\overset{g_i^{k+1}}{=} \left\|\nabla f_i(x^k) - g_i^k - \beta \mathrm{Norm}_\alpha\left(\nabla f_i(x^k) - g_i^k\right)\right\| \\
&\overset{\text{Lemma 1}}{\leq} \left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right| \left\|\nabla f_i(x^k) - g_i^k\right\| \\
&\overset{(*)}{\leq} \left(1 - \frac{\beta}{\alpha + R}\right) R \leq R,
\end{aligned}$$

where we reach $(*)$ by the fact that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$, $\frac{\beta}{\alpha+R} < 1$, and $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$.

Step 3) Derive the descent inequality. By the L-smoothness of f, by the definition of x^{k+1}, and by the fact that $\hat{g}^{k+1} = g^{k+1}$,

$$\begin{aligned} f(x^{k+1}) - f^{\inf} &\leq f(x^k) - f^{\inf} - \frac{\gamma}{\|g^{k+1}\|}\left\langle \nabla f(x^k), g^{k+1}\right\rangle + \frac{L\gamma^2}{2} \\ &= f(x^k) - f^{\inf} - \gamma\left\|g^{k+1}\right\| + \frac{\gamma}{\|g^{k+1}\|}\left\langle \nabla f(x^k) - g^{k+1}, g^{k+1}\right\rangle + \frac{L\gamma^2}{2} \\ &\overset{(11)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|g^{k+1}\right\| + \gamma\left\|\nabla f(x^k) - g^{k+1}\right\| + \frac{L\gamma^2}{2} \\ &\overset{(12)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\left\|\nabla f(x^k) - g^{k+1}\right\| + \frac{L\gamma^2}{2} \\ &\overset{(13)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\frac{1}{n}\sum_{i=1}^n\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| + \frac{L\gamma^2}{2}. \end{aligned}$$

Next, since $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| \leq R$ with $R = \max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\|$, we have

$$f(x^{k+1}) - f^{\inf} \leq f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\| + \frac{L\gamma^2}{2}.$$

Step 4) Finalize the convergence rate. Finally, by re-arranging the terms of the inequality,

$$\begin{aligned} \min_{k\in[0,K]}\left\|\nabla f(x^k)\right\| &\leq \frac{1}{K+1}\sum_{k=0}^K\left\|\nabla f(x^k)\right\| \\ &\leq \frac{[f(x^0) - f^{\inf}] - [f(x^{K+1}) - f^{\inf}]}{\gamma(K+1)} + 2\max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\| + \frac{L}{2}\gamma \\ &\overset{(\dagger)}{\leq} \frac{f(x^0) - f^{\inf}}{\gamma(K+1)} + 2\max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\| + \frac{L}{2}\gamma, \end{aligned}$$

where we reach $(\dagger)$ by the fact that $f^{\inf} \geq f(x^{K+1})$.

E.2. Proof of Corollary 1

If $g_i^0 \in \mathbb{R}^d$ is chosen such that $\max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\| = \frac{D}{(K+1)^{1/2}}$ with any $D > 0$, $\gamma \leq \frac{\beta}{L_{\max}}\frac{D}{\alpha+D}\frac{1}{(K+1)^{1/2}}$, and $\beta < \alpha$, then from Theorem 2, we obtain $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$ with $R = \max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\|$, and

$$\min_{k\in[0,K]}\left\|\nabla f(x^k)\right\| \leq \frac{L_{\max}(\alpha+D)}{\beta D}\frac{f(x^0) - f^{\inf}}{(K+1)^{1/2}} + 2\frac{D}{(K+1)^{1/2}} + \frac{L}{2}\frac{\beta D}{L_{\max}(\alpha+D)}\frac{1}{(K+1)^{1/2}}.$$

E.3. α-NormEC and Clip21 Comparison

We show that the convergence bound of α-NormEC (10) has a smaller factor than that of Clip21 from Theorem 5.6. of [35].

To show this, let $\hat{x}^K$ be selected uniformly at random from a set $\{x^0, x^1, \ldots, x^K\}$. Then, from Theorem 5.6. of [35], Clip21 achieves the following convergence bound:

$$\begin{aligned} \min_{k\in[0,K]}\left\|\nabla f(x^k)\right\| &\leq \mathrm{E}\left[\left\|\nabla f(\hat{x}^K)\right\|\right] \\ &\leq \sqrt{\mathrm{E}\left[\left\|\nabla f(\hat{x}^K)\right\|^2\right]} \\ &\leq \frac{L_{\max}(f(x^0) - f^{\inf})}{\tau(K+1)^{1/2}} + \frac{\sqrt{(1 + C_1/\tau)C_2}}{(K+1)^{1/2}}, \end{aligned}$$

where $\tau > 0$ is a clipping threshold, $C_1 = \max_{i\in[1,n]}\left\|\nabla f_i(x^0)\right\|$, and $C_2 = \max(\max(L, L_{\max})(f(x^0) - f^{\inf})), C_1^2)$.

If $\tau = \frac{L_{\max}}{\sqrt{2L}}\sqrt{f(x^0) - f^{\inf}}$, then

$$\begin{aligned}\min_{k\in[0,K]} \left\|\nabla f(x^k)\right\| &\leq \sqrt{\frac{2L(f(x^0) - f^{\inf})}{K+1}} + \frac{\sqrt{\left(1 + \frac{C_1\sqrt{2L}}{L_{\max}\sqrt{f(x^0)-f^{\inf}}}\right)C_2}}{(K+1)^{1/2}} \\ &\leq \sqrt{\frac{2L(f(x^0) - f^{\inf})}{K+1}} \\ &\quad + \frac{\sqrt{C_2} + \mathcal{O}\left(\max(\sqrt{C_1}\sqrt[4]{f(x^0) - f^{\inf}}, C_1^3/\sqrt{f(x^0) - f^{\inf}})\right)}{(K+1)^{1/2}}.\end{aligned}$$

The first term in the convergence bound of Clip21 matches that of α-NormEC as given in (10). However, the second term in the convergence bound of α-NormEC is $D/\sqrt{K+1}$, where $D > 0$ can be made arbitrarily small. In contrast, the corresponding term for Clip21 is $C/\sqrt{K+1}$, where $C > 0$ may become significantly larger than D if $x^0 \in \mathbb{R}^d$ is far from the stationary point, leading to a large value of $C_1 = \max_{i\in[1,n]} \left\|\nabla f_i(x^0)\right\|$.

F. Private Results

F.1. Proof of Theorem 1

We prove Theorem 1 by two useful lemmas:

1. Lemma 2, which states $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$ for some positive scalars R, given that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$ and the hyperparameters γ, β, α are properly tuned, and
2. Lemma 3, which bounds the difference in expectation between the memory vectors maintained by the central server and clients.

Lemma 3 (DP setting)**.** *Consider* DP-α-NormEC *(Algorithm 1) for solving Problem* (1)*, where Assumption 1 holds. If* $\hat{g}^0 = \frac{1}{n}\sum_{i=1}^n g_i^0$*, then*

$$\mathrm{E}\left[\left\|\hat{g}^{k+1} - \frac{1}{n}\sum_{i=1}^n g_i^{k+1}\right\|\right] \leq \sqrt{\frac{\beta^2(K+1)\sigma_{\mathrm{DP}}^2}{n}}.$$

Proof. From the definition of g_i^k and $\hat{g}^k$,

$$e^{k+1} = e^k + \beta z^{k+1},$$

where $e^k = \hat{g}^k - \frac{1}{n}\sum_{i=1}^n g_i^k$, and $z^k = \frac{1}{n}\sum_{i=1}^n z_i^k$. By applying the equation recursively,

$$e^{k+1} = e^0 + \beta\sum_{l=1}^{k+1} z^l.$$

Therefore, by the triangle inequality,

$$\left\|e^{k+1}\right\| \leq \left\|e^0\right\| + \left\|\beta\sum_{l=1}^{k+1} z^l\right\|.$$

If $\hat{g}^0 = \frac{1}{n}\sum_{i=1}^n g_i^0$, then $e^0 = 0$ and therefore

$$\left\|e^{k+1}\right\| \leq \left\|\beta\sum_{l=1}^{k+1} z^l\right\|.$$

By taking the expectation,

$$\begin{aligned}\mathrm{E}\left[\left\|e^{k+1}\right\|\right] &\leq \mathrm{E}\left[\left\|\beta\sum_{l=1}^{k+1} z^l\right\|\right] \\ &= \mathrm{E}\left[\sqrt{\left\|\beta\sum_{l=1}^{k+1} z^l\right\|^2}\right] \\ &\leq \sqrt{\mathrm{E}\left[\left\|\beta\sum_{l=1}^{k+1} z^l\right\|^2\right]},\end{aligned}$$

where we reach the last inequality by Jensen's inequality. Next, by expanding the terms,

$$\begin{aligned}\mathrm{E}\left[\left\|e^{k+1}\right\|\right] &\leq \sqrt{\beta^2\sum_{l=1}^{k+1}\mathrm{E}\left[\left\|z^l\right\|^2\right] + \beta^2\sum_{j\neq i}\mathrm{E}\left[\langle z^i, z^j\rangle\right]} \\ &\overset{(*)}{=} \sqrt{\beta^2\sum_{l=1}^{k+1}\mathrm{E}\left[\left\|z^l\right\|^2\right]} \\ &\overset{(\ddagger)}{\leq} \sqrt{\frac{\beta^2}{n}\sum_{l=1}^{k+1}\sigma_{\mathrm{DP}}^2},\end{aligned}$$

where we reach $(*)$ by the fact that $\mathrm{E}\left[\langle z^j, z^i\rangle\right] = 0$ for $i \neq j$, and $(\ddagger)$ by the fact that $\mathrm{E}\left[\left\|z^k\right\|^2\right] = \sigma_{\mathrm{DP}}^2/n$ (as z_i^k is independent of z_j^k for $i \neq j$). Therefore,

$$\begin{aligned}\mathrm{E}\left[\left\|e^{k+1}\right\|\right] &\leq \sqrt{\frac{\beta^2(k+1)\sigma_{\mathrm{DP}}^2}{n}} \\ &\overset{k\leq K}{\leq} \sqrt{\frac{\beta^2(K+1)\sigma_{\mathrm{DP}}^2}{n}}.\end{aligned}$$

□

Now, we prove Theorem 1 in three steps.

Step 1) Prove by induction that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$ for $R = \max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\|$. For $k = 0$, this is obvious. Next, let $\left\|\nabla f_i(x^l) - g_i^l\right\| \leq R$ for $R = \max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\|$ for $l = 0, 1, \ldots, k$. Then, if $\beta/(\alpha + R) < 1$, and $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$, then from Lemma 2 $\left\|\nabla f_i(x^{k+1}) - g_i^{k+1}\right\| \leq R$.

Step 2) Bound $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$. From the definition of the Euclidean norm,

$$\begin{aligned}\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| &\overset{g_i^{k+1}}{=} \left\|\nabla f_i(x^k) - g_i^k - \beta\mathrm{Norm}_\alpha\left(\nabla f_i(x^k) - g_i^k\right)\right\| \\ &\overset{\text{Lemma 2}}{\leq} \left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right| \left\|\nabla f_i(x^k) - g_i^k\right\|.\end{aligned}$$

Step 3) Derive the descent inequality in $\mathrm{E}\left[f(x^k) - f^{\inf}\right]$**.** Denote $g^k = \frac{1}{n}\sum_{i=1}^n g_i^k$. By the L-smoothness of f, and by the definition of x^{k+1},

$$\begin{aligned}
f(x^{k+1}) - f^{\inf} &\leq f(x^k) - f^{\inf} - \frac{\gamma}{\|\hat{g}^{k+1}\|}\left\langle \nabla f(x^k), \hat{g}^{k+1}\right\rangle + \frac{L\gamma^2}{2} \\
&= f(x^k) - f^{\inf} - \gamma\left\|\hat{g}^{k+1}\right\| + \frac{\gamma}{\|\hat{g}^{k+1}\|}\left\langle \nabla f(x^k) - \hat{g}^{k+1}, \hat{g}^{k+1}\right\rangle + \frac{L\gamma^2}{2} \\
&\overset{(11)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|\hat{g}^{k+1}\right\| + \gamma\left\|\nabla f(x^k) - \hat{g}^{k+1}\right\| + \frac{L\gamma^2}{2} \\
&\overset{(12)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\left\|\nabla f(x^k) - \hat{g}^{k+1}\right\| + \frac{L\gamma^2}{2} \\
&\overset{(13)}{\leq} f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\frac{1}{n}\sum_{i=1}^n\left\|\nabla f_i(x^k) - g_i^{k+1}\right\| \\
&\quad + 2\gamma\left\|\hat{g}^{k+1} - g^{k+1}\right\| + \frac{L\gamma^2}{2}.
\end{aligned}$$

Next, let $\mathcal{F}^k$ be the history up to iteration k, i.e. $\mathcal{F}^k := \{x^0, z_1^0, \ldots, z_n^0, \ldots, x^k, z_1^k, \ldots, z_n^k\}$. Then,

$$\begin{aligned}
\mathrm{E}\left[f(x^{k+1}) - f^{\inf}\middle|\mathcal{F}^k\right] &\leq f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma\frac{1}{n}\sum_{i=1}^n\mathrm{E}\left[\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|\middle|\mathcal{F}^k\right] \\
&\quad + 2\gamma\mathrm{E}\left[\left\|\hat{g}^{k+1} - g^{k+1}\right\|\middle|\mathcal{F}^k\right] + \frac{L\gamma^2}{2}.
\end{aligned}$$

Next, by the upper-bound for $\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|$,

$$\begin{aligned}
\mathrm{E}\left[\left\|\nabla f_i(x^k) - g_i^{k+1}\right\|\middle|\mathcal{F}^k\right] &\leq \mathrm{E}\left[\left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right|\left\|\nabla f_i(x^k) - g_i^k\right\|\middle|\mathcal{F}^k\right] \\
&= \left|1 - \frac{\beta}{\alpha + \left\|\nabla f_i(x^k) - g_i^k\right\|}\right|\left\|\nabla f_i(x^k) - g_i^k\right\| \\
&\leq \left(1 - \frac{\beta}{\alpha + R}\right)R \leq R,
\end{aligned}$$

where we reach the second inequality by the fact that $\left\|\nabla f_i(x^k) - g_i^k\right\| \leq R$, $\frac{\beta}{\alpha+R} < 1$, and $\gamma \leq \frac{\beta R}{\alpha+R}\frac{1}{L_{\max}}$. Thus,

$$\begin{aligned}
\mathrm{E}\left[f(x^{k+1}) - f^{\inf}\middle|\mathcal{F}^k\right] &\leq f(x^k) - f^{\inf} - \gamma\left\|\nabla f(x^k)\right\| + 2\gamma R \\
&\quad + 2\gamma\mathrm{E}\left[\left\|\hat{g}^{k+1} - g^{k+1}\right\|\middle|\mathcal{F}^k\right] + \frac{L\gamma^2}{2}.
\end{aligned}$$

By taking the expectation, and by the tower property $\mathrm{E}\left[\mathrm{E}\left[X\middle|Y\right]\right] = \mathrm{E}\left[X\right]$,

$$\begin{aligned}
\mathrm{E}\left[f(x^{k+1}) - f^{\inf}\right] &= \mathrm{E}\left[\mathrm{E}\left[f(x^{k+1}) - f^{\inf}\middle|\mathcal{F}^k\right]\right] \\
&\leq \mathrm{E}\left[f(x^k) - f^{\inf}\right] - \gamma\mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] + 2\gamma R \\
&\quad + 2\gamma\mathrm{E}\left[\left\|\hat{g}^{k+1} - g^{k+1}\right\|\right] + \frac{L\gamma^2}{2}.
\end{aligned}$$

Next, by using Lemma 3,

$$\begin{aligned}
\mathrm{E}\left[f(x^{k+1}) - f^{\inf}\right] &\leq \mathrm{E}\left[f(x^k) - f^{\inf}\right] - \gamma\mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] + 2\gamma R \\
&\quad + 2\gamma\sqrt{\frac{\beta^2(K+1)\sigma_{\mathrm{DP}}^2}{n}} + \frac{L\gamma^2}{2}.
\end{aligned}$$

Therefore,

$$\min_{k\in[0,K]} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] \leq \frac{1}{K+1}\sum_{k=0}^{K} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right]$$
$$\leq \frac{\mathrm{E}\left[f(x^0)-f^{\inf}\right]-\mathrm{E}\left[f(x^{K+1})-f^{\inf}\right]}{\gamma(K+1)} + 2R + 2\sqrt{\frac{\beta^2(K+1)\sigma_{\mathrm{DP}}^2}{n}} + \frac{L}{2}\gamma$$
$$\leq \frac{f(x^0)-f^{\inf}}{\gamma(K+1)} + 2R + 2\sqrt{\frac{\beta^2(K+1)\sigma_{\mathrm{DP}}^2}{n}} + \frac{L}{2}\gamma,$$

where we reach the last inequality by the fact that $f^{\inf} \geq f(x^{K+1})$.

F.2. Discussion on Theorem 1

By choosing g_i^0 such that $R = \frac{D}{(K+1)^{1/6}}$ with any $D > 0$, $\beta = \frac{\beta_0}{(K+1)^{2/3}}$ with $\beta_0 \in (0,1]$, $\alpha > 1$, and $\gamma \leq \frac{A}{(K+1)^{5/6}}$ with $A = \frac{\beta_0 D}{L_{\max}(\alpha+D)}$, then the conditions for β, α, γ in Theorem 1 are satisfied, and from Theorem 1 DP-α-NormEC attains the $\mathcal{O}(1/K^{1/6})$ convergence rate in the gradient norm:

$$\min_{k\in[0,K]} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] \leq \frac{C}{(K+1)^{1/6}} + \frac{LA}{2(K+1)^{5/6}},$$

where $C_1 = \frac{f(x^0)-f^{\inf}}{A} + 2D + 2\beta_0\sigma_{\mathrm{DP}}$.

F.3. Utility Guarantee of DP-α-NormEC

In this section, we present the utility guarantee of DP-α-NormEC.

Corollary 2 (Utility guarantee in DP setting)**.** *Consider DP-α-NormEC (Algorithm 1) for solving Problem (1) under the same setting as Theorem 1. If $\sigma_{\mathrm{DP}} = \mathcal{O}(\sqrt{(K+1)\log(1/\delta)}\epsilon^{-1})$, $\beta = \frac{\beta_0}{K+1}$ with $\beta_0 = \mathcal{O}\left(\sqrt{\frac{\Delta}{A}}\right)$ and $\alpha = R = \mathcal{O}\left(\sqrt[4]{d}\sqrt{\Delta A}\right)$ for $\Delta = \sqrt{L_{\max}(f(x^0)-f^{\inf})}$ and $A = \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}\epsilon}$, then Algorithm 1 satisfies (ϵ,δ)-DP and attains the bound*

$$\min_{k\in[0,K]} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] \leq \mathcal{O}\left(\Delta\sqrt[4]{d\frac{\log(1/\delta)}{n\epsilon^2}}\right).$$

Proof: Let $\sigma_{\mathrm{DP}} = \mathcal{O}\left(\frac{\sqrt{(K+1)\log(1/\delta)}}{\epsilon}\right)$, and $\beta = \frac{\beta_0}{K+1}$ with $0 < \beta_0 < \alpha + R$. Then, from Theorem 1, we get $\gamma \leq \frac{\beta_0 R}{\alpha+R}\frac{1}{L_{\max}}\frac{1}{K+1}$ with $R = \max_{i\in[1,n]}\left\|\nabla f_i(x^0) - g_i^0\right\|$, and

$$\min_{k\in[0,K]} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] \leq \frac{L_{\max}(\alpha+R)(f(x^0)-f^{\inf})}{\beta_0 R} + 2R + 2\frac{\beta_0\sqrt{\log(1/\delta)}}{\sqrt{n}\epsilon} + \frac{L\beta_0 R}{2(\alpha+R)L_{\max}}\frac{1}{K+1}.$$

If $\beta_0 = \mathcal{O}\left(\sqrt{\frac{L_{\max}(f(x^0)-f^{\inf})}{A}}\right)$ and $\alpha = R = \mathcal{O}\left(\sqrt[4]{d}\sqrt{L_{\max}(f(x^0)-f^{\inf})A}\right)$ for $A = \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}\epsilon}$, then

$$\min_{k\in[0,K]} \mathrm{E}\left[\left\|\nabla f(x^k)\right\|\right] \leq \mathcal{O}\left(\sqrt[4]{d}\sqrt{L_{\max}(f(x^0)-f^{\inf})A}\right).$$

F.4. Private initialization of the memory vectors

The server's initial memory vector $\hat{g}^0$ in Algorithm 1 is set as the average of the initial client memory vectors: $\hat{g}^0 = \frac{1}{n}\sum_{i=1}^n g_i^0$. Notably, Lemma 3 in our analysis allows for more general initializations, including an additive error term e: $\hat{g}^0 = \frac{1}{n}\sum_{i=1}^n g_i^0 + e$. This initial error can be kept small by privately estimating the mean of the g_i^0 vectors, incurring a privacy cost only once. Furthermore, secure aggregation techniques can completely remove this error. For instance, if clients share a random seed, they can add and subtract identical cryptographic noise (h) to their respective local memory vectors (e.g., $g_1^0 + h$ and $g_2^0 - h$). This protects the individual vectors from the server while ensuring the average remains accurate: $\frac{1}{2}(g_1^0 + h) + \frac{1}{2}(g_2^0 - h) = \frac{1}{2}(g_1^0 + g_2^0)$.

G. Experimental Details and Additional Results

We include details on experimental setups and additional results in the non-private and private training for the ResNet20 model on the CIFAR-10 dataset.

G.1. Additional Experimental Details

The dataset was split into train (90%) and test (10%) sets. The train samples were randomly shuffled and distributed across 10 workers. Every worker computed gradients with batch size 32. The training was performed for 300 communication rounds. The random seed was fixed to 42 for reproducibility.

All the methods were run with a constant step size (learning rate) without other techniques, such as schedulers, warm-up, or weight decay. They were evaluated across the following hyperparameter combinations:

- step size γ: $\{0.001, 0.01, 0.1, 1.0\}$,
- Sensitivity/clip threshold β: $\{0.01, 0.1, 1.0, 10.0\}$,
- Smoothed normalization value α: $\{0.01, 0.1, 1.0\}$.

Our implementation is based on the public GitHub repository [29]. Experiments were performed on a machine with a single GPU: NVIDIA GeForce RTX 3090.

G.2. Non-private Training

G.2.1. Sensitivity of α-NormEC to Parameters β, α

Figure 5 supplements the results in Figure 1 with other metrics. Figure 4 displays convergence curves across different combinations of α, β parameters with optimally selected step sizes γ.

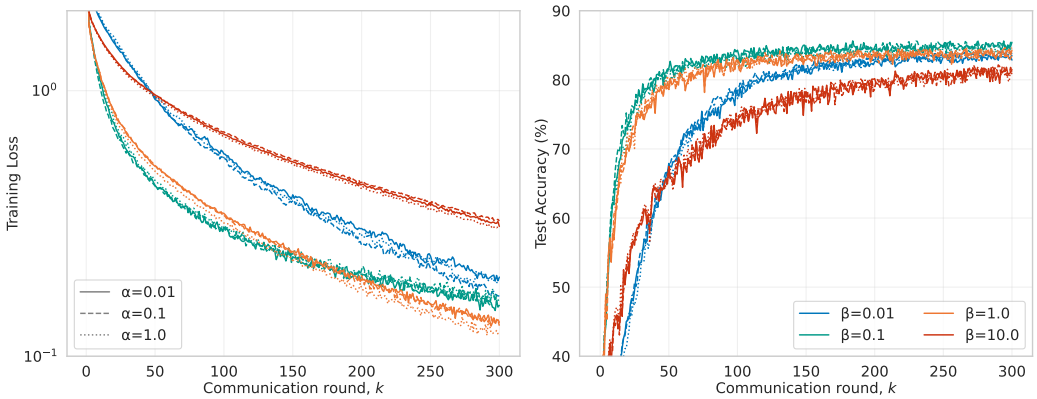

Figure 4: Training loss and test accuracy of non-private α-NormEC with $\alpha = 0.01$ [solid], 0.1 [dashed], and 1.0 [dotted], and $\beta = 0.01$ [blue], 0.1 [green], 1.0 [orange], and 10.0 [red].

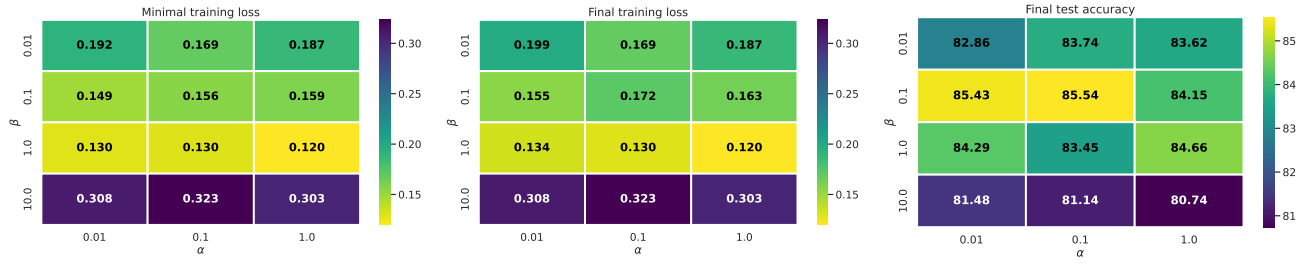

Figure 5: **Minimal** train loss (left), **final** train loss (middle), and **final** test accuracy (right) achieved by non-private α-NormEC, after 300 communication rounds using a fine-tuned constant step size γ.

G.2.2. Benefits of Error Compensation

Leveraging error compensation (EC), α-NormEC without server normalization achieves superior performance compared to DP-SGD with direct smoothed normalization across a range of β and γ hyperparameter settings (where $\alpha = 0.01$), in terms of the final test accuracy reported in Figure 6 and Table 7. From Table 7, α-NormEC without server normalization consistently outperforms DP-SGD across most combinations. This trend is particularly evident for small β values ($\beta = 0.01$), where DP-SGD achieves only 51.10% accuracy while α-NormEC reaches 84.04%. The only exception is $\beta = 10.0$, where DP-SGD outperforms α-NormEC. However, this combination is less practical in the private setting, as too high β values imply high private noise, thus leading to slow algorithmic convergence.

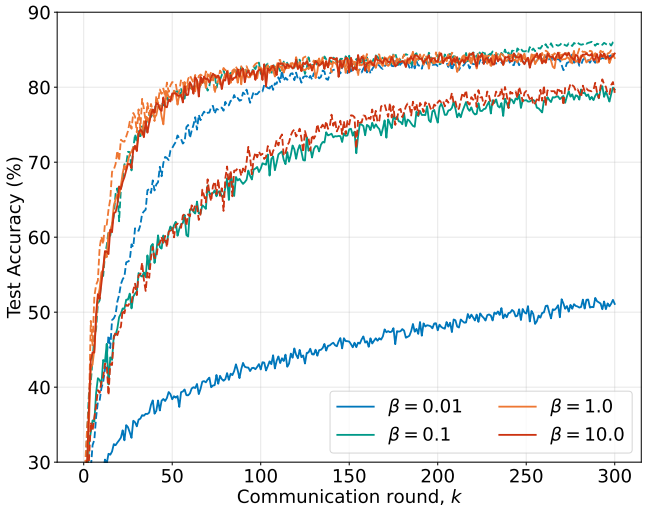

Figure 6: Comparison of DP-SGD (2) [solid] and α-NormEC (1) [dashed] without server normalization.

Method	β	γ	Final Accuracy
α-NormEC	0.01	0.1	84.04%
	0.1	0.1	**86.09%**
	1.0	0.1	84.80%
	10.0	0.01	79.25%
DP-SGD (2)	0.01	1.0	51.10%
	0.1	1.0	79.68%
	1.0	1.0	83.89%
	10.0	0.1	84.50%

Figure 7: Best configurations and final test accuracies.

G.2.3. Effect of Server Normalization

We investigate the impact of server-side normalization (Line 11 in Algorithm 1) on the convergence performance of α-NormEC. We reported training loss and test accuracy of α-NormEC without and with server normalization in Figure 8 while summarizing their final test accuracy in Table 2.

α-NormEC without server normalization generally achieves faster convergence in training loss and higher test accuracy than α-NormEC with server normalization across varying β values. Notably, at $\beta = 0.1$, α-NormEC without server normalization achieves the highest test accuracy of **86.09%**. Only at the large value of $\beta = 10.0$ does server normalization improve the test accuracy of α-NormEC without server normalization by approximately 2.2%.

Method: α-NormEC	β	γ	Final Accuracy
With server normalization	0.01	0.01	82.86%
	0.1	0.1	85.43%
	1.0	0.1	84.29%
	10.0	0.1	81.48%
Without server normalization	0.01	0.1	84.04%
	0.1	0.1	**86.09%**
	1.0	0.1	84.80%
	10.0	0.01	79.25%

Table 2: Best configurations and final test accuracies of α-NormEC with and without server normalization.

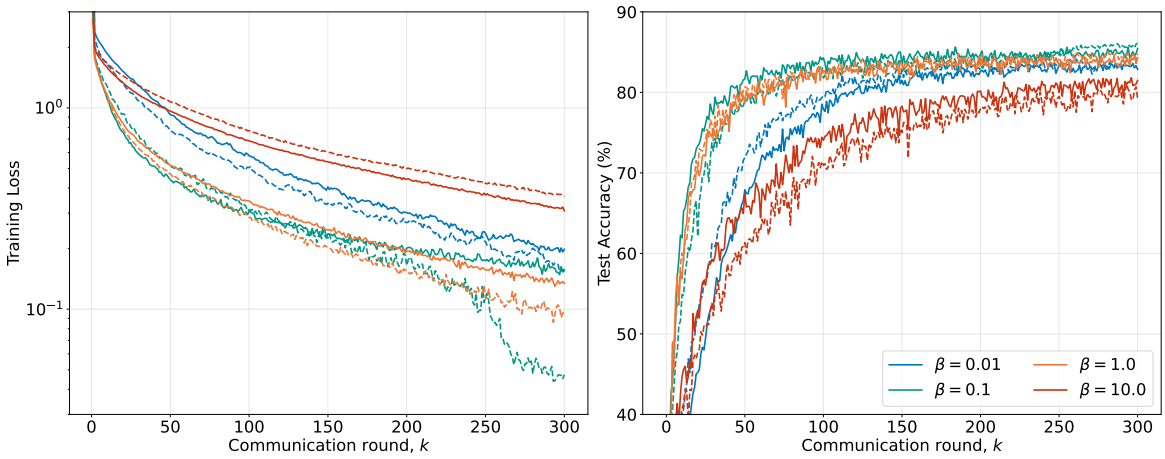

Figure 8: Training loss and test accuracy of α-NormEC with [solid] and without [dashed] server normalization.

G.2.4. Comparison of Clip21 and α-NormEC

Figure 10 and Table 9 show that α-NormEC without server normalization[2] achieves comparable convergence performance to Clip21 for most β values. At small β values $(0.01, 0.1)$, α-NormEC without server normalization attains slightly lower final test accuracy. However, at high $\beta = 10.0$, Clip21 maintains the higher test accuracy, as the large clipping threshold effectively disables clipping. Furthermore, in most cases, both methods achieve their best performance with $\gamma = 0.1$, except for α-NormEC at $\beta = 10.0$, where a smaller learning rate ($\gamma = 0.01$) was optimal.

Method	β	γ	Final Accuracy
Clip21	0.01	0.1	83.00%
	0.1	0.1	85.91%
	1.0	0.1	84.78%
	10.0	0.1	83.19%
α-NormEC	0.01	0.1	84.04%
	0.1	0.1	**86.09**%
	1.0	0.1	84.80%
	10.0	0.01	79.25%

Figure 9: Best configurations and final test accuracies.

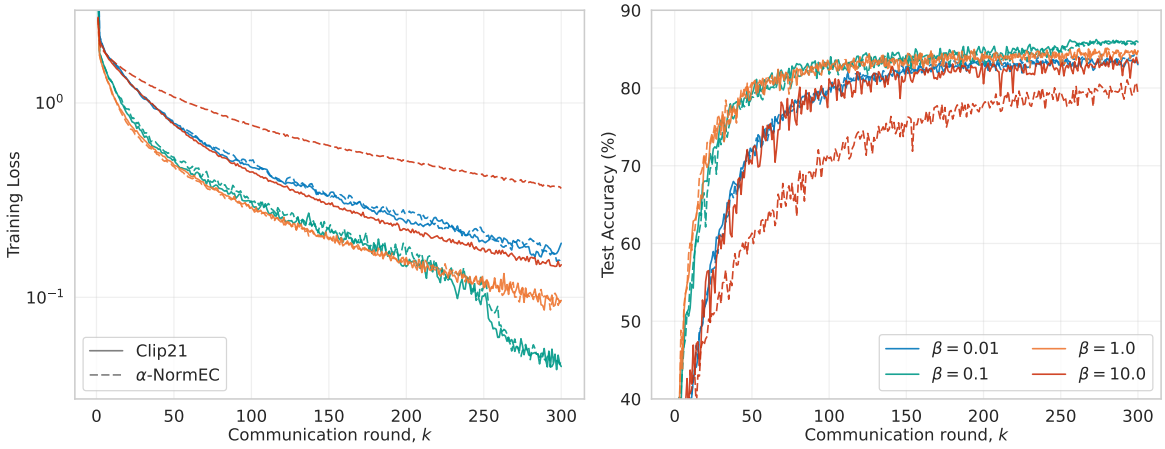

Figure 10: Training loss and test accuracy of Clip21 [solid] and α-NormEC [dashed] without server normalization in the non-private training.

[2]We ran α-NormEC without server normalization because it showed better performance according to Appendix G.2.3.

G.3. Private Training

We complement the results in Section 4 with test accuracy convergence curves in Figure 12 (right). Additionally, Figure 13 presents a comprehensive heatmap analysis of the highest test accuracy achieved by DP-α-NormEC with and without server normalization (SN) across different privacy levels (β) and learning rates (γ). The heatmaps reveal that without server normalization, performance is highly sensitive to hyperparameter selection, with accuracy ranging from 10% to 77.56% depending on the specific β-γ combination. With server normalization, this sensitivity is significantly reduced, with performance varying more gradually across the parameter space. The rightmost heatmap quantifies this difference, showing that server normalization provides substantial benefits (up to +53.49%) at high privacy levels ($\beta = 1.0$) and higher learning rates, while the non-normalized version performs better (up to -37.92%) at lower privacy levels with specific learning rates.

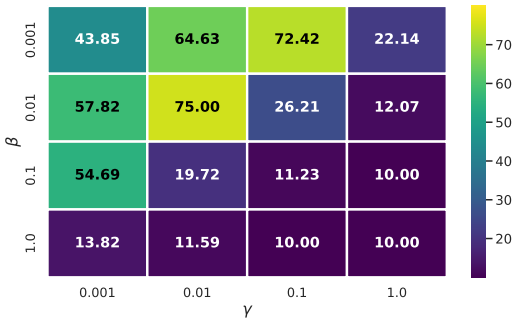

Figure 11: The highest test accuracy of DP-Clip21.

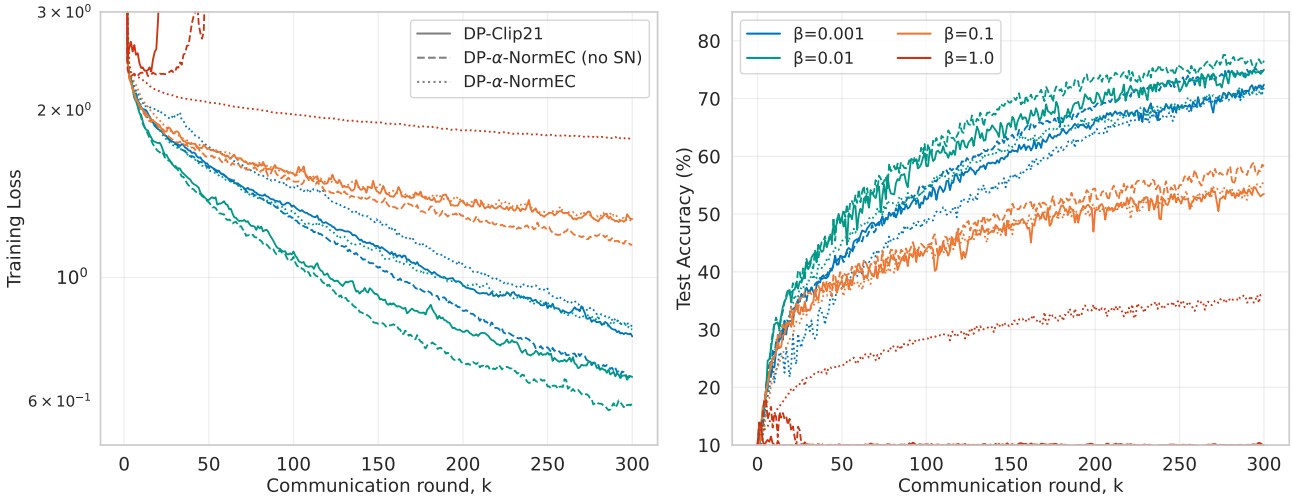

Figure 12: Training loss and test accuracy of DP-Clip21 [solid], and DP-α-NormEC with [dotted] and without [dashed] server normalization (SN) across different β values (with fine-tuned step sizes).

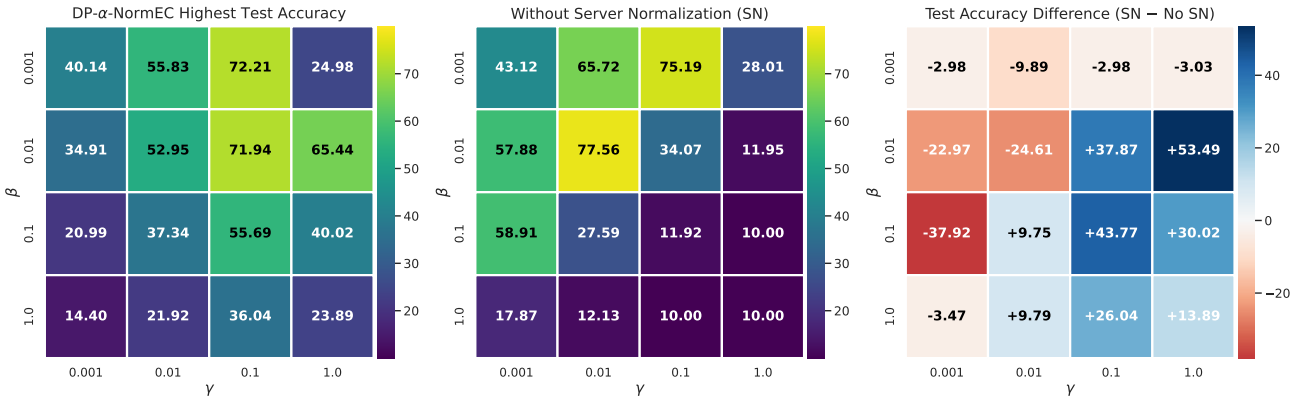

Figure 13: The highest test accuracy of DP-α-NormEC with [left] and without [center] Server Normalization (SN), and their difference [right].

G.3.1. Shorter Training

We present additional results in Figures 15, 14 by running DP-α-NormEC for **150 communication rounds**. The step size γ is tuned for every parameter β. In the non-private setting, (reasonably) longer training is basically always beneficial. However, in the private scenario, it may not hold due to increased noise variance as it scales with a number of iterations. Interestingly, we observe that for $\beta = 1$, the highest achieved accuracy after 150 iterations is almost the same as after a doubled communication budget of 300 iterations.

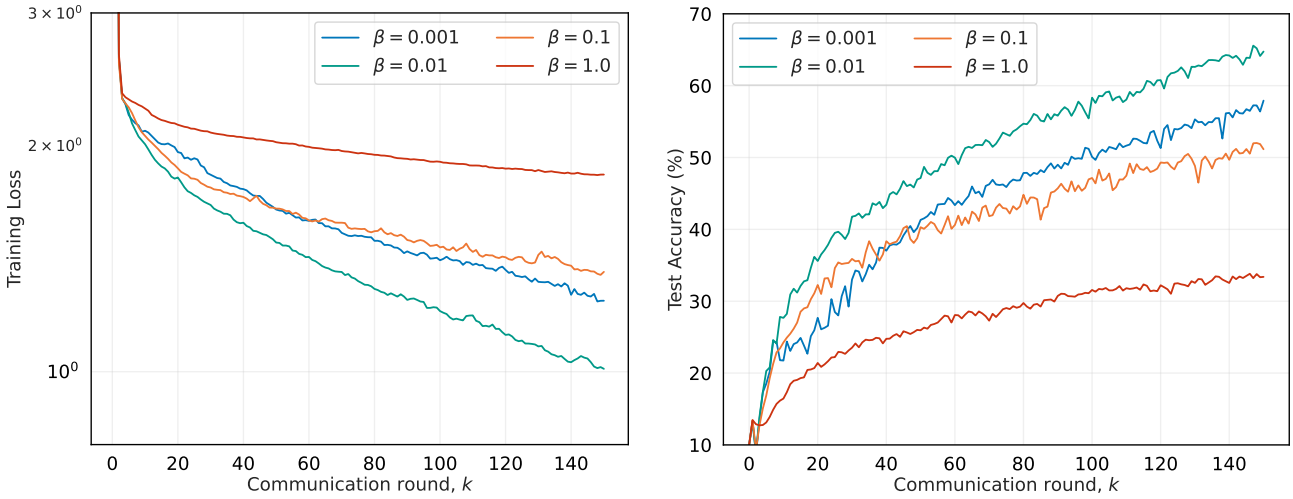

Figure 14: Training loss and test accuracy of DP-α-NormEC across different β values.

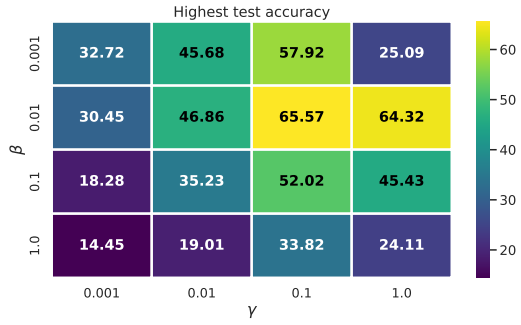

Figure 15: Best test accuracy of DP-α-NormEC across different β, γ values.