Convergence and Privacy of Decentralized Nonconvex Optimization With Gradient Clipping and Communication Compression

Boyue Li and Yuejie Chi, *Fellow, IEEE*

***Abstract*—Achieving communication efficiency in decentralized machine learning has been attracting significant attention, with communication compression recognized as an effective technique in algorithm design. This paper takes a first step to understand the role of gradient clipping, a popular strategy in practice, in decentralized nonconvex optimization with communication compression. We propose `PORTER`, which considers two variants of gradient clipping added before or after taking a mini-batch of stochastic gradients, where the former variant `PORTER-DP` allows local differential privacy analysis with additional Gaussian perturbation, and the latter variant `PORTER-GC` helps to stabilize training. We develop a novel analysis framework that establishes their convergence guarantees without assuming the stringent bounded gradient assumption. To the best of our knowledge, our work provides the first convergence analysis for decentralized nonconvex optimization with gradient clipping and communication compression, highlighting the trade-offs between convergence rate, compression ratio, network connectivity, and privacy.**

***Index Terms*—Communication compression, convergence rate, gradient clipping, local differential privacy.**

I. Introduction

DECENTRALIZED machine learning has been attracting significant attention in recent years, which can be often modeled as a nonconvex finite-sum optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}),\ \ f_i(\boldsymbol{x}) = \frac{1}{m}\sum_{\boldsymbol{z}\in\mathcal{Z}_i} \ell(\boldsymbol{x};\boldsymbol{z}), \quad (1)$$

where $\boldsymbol{x}\in\mathbb{R}^d$ and $\boldsymbol{z}$ denote the optimization parameter and one data sample, $\ell(\boldsymbol{x};\boldsymbol{z})$ denotes the sample loss function that is nonconvex in $\boldsymbol{x}$, and $f_i(\boldsymbol{x})$ and $f(\boldsymbol{x})$ denote the local objective function at agent i and the global objective function. In addition, $\mathcal{Z}_i$ denotes the dataset at agent i, $m = |\mathcal{Z}_i|$ denotes the local sample size, and n denotes the number of agents. An undirected communication graph $\mathcal{G}$ is used to model the connectivity between any two agents, where there is an edge between agent i and j only if they can communicate. The goal is to efficiently optimize the global objective function $f(\boldsymbol{x})$ in a decentralized manner, subject to the network connectivity constraints specified by $\mathcal{G}$.

Communication efficiency is critical to decentralized optimization algorithms, as communication can quickly become bottleneck of the system as the number of agents and the size of the model increase. This has led to the development of communication compression (or quantization) techniques, which can significantly reduce the communication burden per round by transferring compressed information, especially when the communication bandwidth is limited. Therefore, a number of recent works have focused on designing decentralized nonconvex optimization algorithms with communication compression, including but not limited to [1], [2], [3], [4], [5], [6].

Built upon this line of work, the paper aims to understand the role of gradient clipping in decentralized nonconvex optimization algorithms with communication compression. On the one hand, gradient clipping has been used widely in privacy-preserving algorithms [7] to enable (local) differential privacy guarantees [8]. On the other hand, gradient clipping is also observed to be beneficial in stabilizing neural network training [9]. However, since gradient clipping necessarily introduces bias, the characterization of the convergence becomes much more challenging compared to their unclipped counterpart. As a result, most of the existing theoretical analyses for stochastic gradient algorithms with clipping—in the context of centralized and server/client settings—make strong assumptions such as the bounded gradient assumption [7], [10], [11] and the uniformly bounded gradient assumption [9], [12], [13]. To the best of our knowledge, the convergence of stochastic gradient algorithms with clipping in the decentralized setting has not been investigated before.[1]

Received 9 June 2024; revised 27 October 2024; accepted 29 December 2024. Date of publication 3 January 2025; date of current version 28 February 2025. The work of Boyue Li was supported by Wei Shen and Xuehong Zhang Presidential Fellowship at Carnegie Mellon University. This work was supported in part by ONR under Grant N00014-19-1-2404, in part by AFRL under Grant FA8750-20-2-0504, and in part by NSF under Grant CCF-1901199, Grant CCF-2007911, and Grant CNS-2148212. The guest editor coordinating the review of this article and approving it for publication was Prof. Jiebo Luo. *(Corresponding author: Yuejie Chi.)*

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: boyuel@andrew.cmu.edu; yuejiechi@cmu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSTSP.2025.3526081, provided by the authors.

Digital Object Identifier 10.1109/JSTSP.2025.3526081

[1] A preliminary version of this paper was posted on Arxiv in May 2023 and reported in the dissertation of the first author.

TABLE I
COMPARISON OF FINAL UTILITY UPPER BOUNDS AND COMMUNICATION COMPLEXITIES OF DIFFERENT STOCHASTIC ALGORITHMS THAT ACHIEVE (ϵ, δ)-DP/LDP

Algorithm	Privacy	Compression operator	Bounded gradient	Utility	Communication rounds
DP-SGD [7]	DP	-	✓	ϕ_m	-
DDP-SRM [10]	DP	-	✓	$\frac{1}{n}\phi_m$	$n^2 d\phi_m^{-1}$
Soteria-SGD [1] [11]	LDP	Unbiased	✓	$(1+\theta^{1/2})\big(\frac{1+\omega}{n}\big)^{1/2}\phi_m$	$(1+\theta^{1/2})\big(\frac{n}{1+\omega}\big)^{2/3} d\phi_m^{-1}$
PORTER-DP (Algorithm 1)	LDP	General	✓	$\frac{1}{(1-\alpha)^{8/3}\rho^{4/3}}\phi_m$	ϕ_m^{-2}
PORTER-DP (Algorithm 1)	LDP	General	✗	$\frac{1}{(1-\alpha)^{16/3}\rho^{8/3}}\phi_m$	ϕ_m^{-2}

The Big-O notation (defined in Section I-C) is omitted for simplicity. DP-SGD is a single-server optimization algorithm that serves as a baseline, to show the overhead brought in by the distributed setting. DDP-SRM and Soteria-SGD are server/client distributed algorithms, but DDP-SRM doesn't use communication compression. [1] $\theta = (1-\omega)^{-3/2} n^{-1/2}$, where ω is the parameter for unbiased compression.

A. Our Contributions

This paper proposes PORTER (cf. Algorithm 1),[2] which is a communication-efficient decentralized algorithm for nonconvex finite-sum optimization with gradient clipping and communication compression. PORTER is built on BEER [3]—a fast decentralized algorithm with communication compression proposed recently—by introducing gradient clipping to the local stochastic gradient computation at agents, while inheriting the desirable designs such as error feedback and stochastic gradient tracking that are crucial in enabling the fast convergence of BEER. PORTER considers two variants of gradient clipping, corresponding to adding it *before* or *after* taking a mini-batch of stochastic gradients. In particular, the former variant PORTER-DP allows local differential privacy (LDP) analysis with additional Gaussian perturbation, and the latter variant PORTER-GC helps to stabilize training. Assuming a smooth clipping operator (Definition 2) and general compression operators (Definition 3), the highlights of our contributions are as follows.

1) We establish that PORTER-DP (cf. Algorithm 1) achieves (ϵ, δ)-LDP under appropriate Gaussian perturbation. Under the bounded gradient assumption (when gradient clipping can be ignored), PORTER-DP converges in average squared ℓ_2 gradient norm as $\frac{1}{T}\sum_{t\in[T]} \mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2^2 \lesssim \rho^{-\frac{4}{3}}(1-\alpha)^{-\frac{8}{3}}\phi_m$ in $T = \phi_m^{-2}$ iterations, where $\bar{\boldsymbol{x}}^{(t)}$ is the average parameter, $\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$ is the baseline utility[3] for a centralized stochastic algorithm to achieve (ϵ, δ)-DP with m data samples [7], $\rho \in (0, 1]$ is the compression ratio, and $\alpha \in [0, 1)$ is the mixing rate of the topology.
2) However, the bounded gradient assumption might be too stringent to hold in practice. Instead we further establish that under the local variance and bounded dissimilarity assumptions, PORTER-DP converges in $T = \phi_m^{-2}$ iterations in minimum ℓ_2 gradient norm as $\min_{t\in[T]} \mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 \lesssim \rho^{-\frac{4}{3}}(1-\alpha)^{-\frac{8}{3}}\phi_m^{1/2}$.
3) We establish that under the local variance and bounded dissimilarity assumptions, by properly choosing the mini-batch size, PORTER-GC converges in minimum ℓ_2 gradient norm as $\min_{t\in[T]} \mathbb{E}\big\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\big\|_2 \lesssim \rho^{-\frac{2}{3}}(1-\alpha)^{-\frac{4}{3}}T^{-1/2}$, which matches the convergence rate of classical centralized stochastic algorithms.

Our work develops a novel analysis framework that establishes their convergence guarantees without assuming the stringent bounded gradient assumption, highlighting comprehensive trade-offs between convergence rate, compression ratio, network connectivity, and privacy. To the best of our knowledge, our work provides the first private decentralized optimization algorithm with communication compression, and a systematic investigation of gradient clipping in the fully decentralized setting. Table I provides a detailed comparison of PORTER-DP with prior art on private server-client algorithms, where the bounded gradient assumption is all in effect except ours.

B. Related Works

Decentralized optimization algorithms have been extensively studied to solve large-scale optimization problems. We review most closely related works in this section, and refer readers to more comprehensive reviews in [14], [15].

Decentralized stochastic nonconvex optimization: Decentralized stochastic algorithms have been a actively researched area in recent years. Various algorithms have been proposed by directly adapting existing centralized algorithms, e.g., [16], [17], [18], [19], [20], [21], [22]. However, the simple adaptations usually fail to achieve better convergence rates. Gradient tracking [23], originally proposed by the control theory community, can be used to track the global gradient at each agent, which leads to a simple systematic framework for extending existing centralized algorithms to the decentralized setting. Gradient tracking can be

[2]The name is coined for two reasons: 1) PORTER has strong connection to the prior algorithm BEER (porter is a kind of dark beer), and 2) the authors developed this algorithm in Porter Hall at Carnegie Mellon University.

[3]Here, the utility is defined as the average of gradient norms $\frac{1}{T}\sum_t \|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2$.

used for both deterministic optimization algorithms, e.g., [24], [25], [26], [27], and stochastic algorithms, e.g., [28], [29], [30], [31], [32], [33].

Communication compression: In [34], [35], gradient compression was adopted to create a server/client distributed SGD algorithm, however, the large variance of compressed gradients leads to a sub-optimal convergence rate. [36] first proposed the use of error feedback to compensate for the variance induced by compression. [37], [38], [39], [40], [41], [42] all equipped similar mechanism to improve convergence for server/client distributed optimization algorithms, and [43], [44] formalized the error feedback mechanism and reaches an $O(1/T)$ convergence rate for smooth nonconvex objective functions. [1], [2], [3], [6], [45], [46], [47], [48] further extended communication compression schemes to the decentralized setting, [49] to vertical federated learning, and [50] considered second-order convergence.

Private optimization algorithms: The concern of leaking sensitive data has been increasing with the rapid development of large-scale machine learning systems. To address this concern, the concept of differential privacy is widely adopted [8], [51], where a popular approach to protect privacy is adding noise to the model or gradients. This approach is first adopted in the single server setting to design differentially private optimization algorithms [7], [52], [53], [54], [55], [56], while [11], [57], [58], [59], [60], [61], [62] considered differential privacy for the server/client distributed setting.

Gradient clipping: Understanding gradient clipping has gained significant attention in recent years. Earlier works, e.g. [63], [64], [65], [66], [67], used gradient clipping as a pure heuristic to solve gradient vanishing/exploding problems without theoretical understandings. Then, [13], [68], [69], [70] introduced theoretical analyses to understand its impact on the convergence rate and training performance. This question is also investigated in [55], [62], [71], [72], which applies gradient clipping to limit the magnitude of each of the sample gradients, so that the variance of privacy perturbation can be decided without the bounded gradient assumption. While finishing up this paper, we became aware of [73], which also develops convergence guarantees on the minimum ℓ_2 gradient norm of clipped stochastic gradient algorithms in the centralized setting with a piece-wise linear clipping operator. In contrast, our focus is on the decentralized setting with a smooth clipping operator, where extra care is taken to deal with the discrepancy between the local and global objective functions.

C. Paper Organization and Notation

Section II introduces preliminary concepts, Section III describes the algorithm development, Section IV shows the theoretical performance guarantees for PORTER, Section V provides numerical evidence to support the analysis, and Section VI concludes the paper. Proofs are postponed to the supplemental material and can be found at [74].

Throughout this paper, we use uppercase and lowercase boldface letters to represent matrices and vectors, respectively. We use $\|\cdot\|_{\mathsf{op}}$ for matrix operator norm, $\|\cdot\|_{\mathsf{F}}$ for Frobenius norm, $\boldsymbol{I}_n$ for the n-dimensional identity matrix, $\mathbf{1}_n$ for the n-dimensional all-one vector and $\mathbf{0}_{d\times n}$ for the $(d\times n)$-dimensional zero matrix. For two real functions $f(\cdot)$ and $g(\cdot)$ defined on $\mathbb{R}^+$, we say $f(x)=O(g(x))$ or $f(x)\lesssim g(x)$ if there exists some universal constant $M>0$ such that $f(x)\le Mg(x)$. The notation $f(x)=\Omega(g(x))$ or $f(x)\gtrsim g(x)$ means $g(x)=O(f(x))$.

II. Preliminaries

We introduce a few important preliminary concepts in this section.

A. Mixing

The information mixing between agents is conducted by updating the local information via a weighted sum of information from neighbors, which is characterized by a mixing (gossiping) matrix. Concerning this matrix is an important quantity called the mixing rate, defined in Definition 1.

Definition 1 (Mixing matrix and mixing rate): The *mixing matrix* is a matrix $\boldsymbol{W}=[w_{ij}]\in\mathbb{R}^{n\times n}$, such that $w_{ij}=0$ if agent i and j are not connected according to the communication graph $\mathcal{G}$. Furthermore, $\boldsymbol{W}\mathbf{1}_n=\mathbf{1}_n$ and $\boldsymbol{W}^\top\mathbf{1}_n=\mathbf{1}_n$. The *mixing rate* of a mixing matrix $\boldsymbol{W}$ is defined as

$$\alpha:=\left\|\boldsymbol{W}-\tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right\|_{\mathsf{op}}. \tag{2}$$

The mixing rate describes the connectivity of a communication graph and the speed of information sharing. Generally, a better connected graph leads to a smaller mixing rate, for example, $\boldsymbol{W}$ can be the averaging matrix for a fully connected communication network, which results in $\alpha=0$. A comprehensive list of bounds on $1-\alpha$ is provided by [75, Proposition 5]. Our analysis does not require the mixing matrix to be doubly stochastic, while allows us to use a non-symmetric matrix with negative values as the mixing matrix, such as the FDLA matrix [17], which has a smaller mixing rate under the same connectivity pattern.

B. Gradient Clipping

In practice, gradient clipping is frequently adopted to ensure the gradients are within a predetermined region, so that the variance of privacy perturbation can be decided accordingly. The clipping operator we adopt is a smooth clipping operator [12] defined in Definition 2, which scales a vector into a ball of radius τ centered at the origin.

Definition 2 (Smooth clipping operator): For $\boldsymbol{x}\in\mathbb{R}^d$, the clipping operator is defined as

$$\mathsf{Clip}_\tau(\boldsymbol{x})=\frac{\tau}{\tau+\|\boldsymbol{x}\|_2}\boldsymbol{x}.$$

For $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]\in\mathbb{R}^{d\times n}$, the distributed clipping operator is defined as

$$\mathsf{Clip}_\tau(\boldsymbol{X})=[\mathsf{Clip}_\tau(\boldsymbol{x}_1),\ldots,\mathsf{Clip}_\tau(\boldsymbol{x}_n)].$$

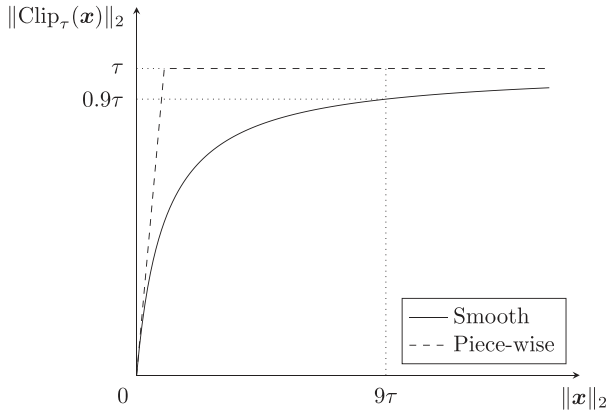


Fig. 1. Illustration of input norm and clipped norm for the smooth clipping operator (Definition 2) and piece-wise linear clipping operator, where τ is the clipping parameter.

Remark 1: Another widely used clipping operator is the piece-wise linear clipping operator, which scales inputs whenever its gradient norm is larger than τ and does nothing otherwise, defined by

$$\mathsf{Clip}_\tau(\boldsymbol{x}) = \boldsymbol{x} \min\left\{1, \tau/\|\boldsymbol{x}\|_2\right\}.$$

Fig. 1 plots the norm of a vector before and after clipping for these two clipping operators, which show that they behave quite similarly.

C. Compression Operators

Following [43], [44], Definition 3 defines a randomized general compression operator that only guarantees the expected compression error $\mathbb{E}\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2$ is less than the magnitude of original message $\|\boldsymbol{x}\|_2^2$.

Definition 3 (General compression operator): A randomized map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a ρ-compression operator if $\forall \boldsymbol{x} \in \mathbb{R}^d$ and some $\rho \in [0, 1]$, the following inequality holds:

$$\mathbb{E}\left\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\right\|_2^2 \leq (1-\rho)\|\boldsymbol{x}\|_2^2.$$

Many widely used compression schemes can be modeled as special cases, for example, random sparsificiation and top-k compression.

Example 1 (Random sparsification): Random sparsification keeps an element from a d-dimensional vector with probability $\frac{k}{d}$. Let $\boldsymbol{u} \in \mathbb{R}^d$ where $u_i \sim B(\frac{k}{d})$, then random sparsification is defined as $\text{random}_k(\boldsymbol{x}) = \boldsymbol{u} \odot \boldsymbol{x}$, which satisfies Definition 3 with $\rho = \frac{k}{d}$.

Example 2 (top$_k$): top$_k$ [37], [38] keeps k elements that have the largest absolute values and sets other elements to 0, which is defined as $\text{top}_k(\boldsymbol{x}) := \boldsymbol{x} \odot \boldsymbol{u}(\boldsymbol{x})$, where $[\boldsymbol{u}(\boldsymbol{x})]_i = 1$ if the absolute value of the i-th element is one of the k-largest absolute values, otherwise $[\boldsymbol{u}(\boldsymbol{x})]_i = 0$. It follows that top$_k$ satisfies Definition 3 with $\rho = k/d$.

D. Local Differential Privacy

In decentralized learning systems, all agents share information with their neighbors that are potentially sensitive. If some agents are exploited by adversaries, the system will face a risk of privacy leakage even when the system-level privacy is protected. Therefore, we introduce local differential privacy (LDP)—defined in Definition 4—following [76], [77], [78], which protects each agent's privacy from leaking to other agents.

Definition 4 (Local differential privacy (LDP)): A randomized mechanism $\mathcal{M} : \mathcal{Z} \to \mathcal{R}$ with domain $\mathcal{Z}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-local differential privacy for client i, if for any two neighboring dataset $\boldsymbol{Z}_i, \boldsymbol{Z}_i' \in \mathcal{Z}$ at client i and for any subset of outputs $\boldsymbol{R} \subseteq \mathcal{R}$, it holds that

$$\mathbb{P}\left(\mathcal{M}(\boldsymbol{Z}_i) \in \boldsymbol{R}\right) \leq e^\epsilon \mathbb{P}\left(\mathcal{M}(\boldsymbol{Z}_i') \in \boldsymbol{R}\right) + \delta. \tag{3}$$

The two datasets $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_i'$ are called neighboring if they are only different by one data point at client i.

Definition 4 is a stricter privacy guarantee because it can imply general differential privacy (DP). Consequently, LDP requires a larger perturbation variance than general DP. To identify the impact of the decentralized LDP setting compared to centralized DP setting, we define the baseline utility

$$\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}, \tag{4}$$

which can be understood as the final utility of a centralized system with m data samples that guarantees (ϵ, δ)-DP. For typical private problems, the local sample size m has to be large enough for the privacy perturbation to deliver meaning guarantees, we impose a mild assumption that $\phi_m < 1$ to simplify presentation. For example, the problem defined in (1) has in total mn data samples, running an (ϵ, δ)-DP algorithm on one server that can access all data will achieve $\frac{1}{n}\phi_m$ utility in $n\phi_m^{-1}$ iterations.

III. Proposed Algorithm

We propose PORTER, a novel stochastic decentralized optimization algorithm for finding first-order stationary points of nonconvex finite-sum problems with gradient clipping and communication compression; the details are described in Algorithm 1. On a high level, PORTER is composed of local stochastic gradient updates and neighboring information sharing, following a similar structure as BEER [3], in terms of the use of error feedback [43], which accelerates the convergence with biased compression operators, and stochastic gradient tracking to track the global gradient locally at each agent. A key difference, however, is the use of gradient clipping. Motivated by efficient training and privacy preserving, we consider two variants, corresponding to clipping *before* the mini-batch with privacy perturbation (PORTER-DP), and *after* taking the mini-batch (PORTER-GC), respectively.

Before proceeding, we introduce some notation convenient for describing decentralized algorithms. Let $\boldsymbol{x}_i \in \mathbb{R}^d$ be the optimization variable at agent i, we define the collection of all optimization variables as a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, and the average as $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$. The gradient estimates $\boldsymbol{V}$, stochastic gradients $\boldsymbol{G}$, perturbation noise $\boldsymbol{E}$, compressed

Algorithm 1: PORTER

1: **input:** $\bar{\boldsymbol{x}}^{(0)}$, gradient stepsize η, consensus stepsize γ, clipping threshold τ, mini-batch size b, perturbation noise σ_p, number of iterations T
2: **initialize:** $\boldsymbol{V}^{(0)} = \boldsymbol{Q}_v^{(0)} = \boldsymbol{G}_p^{(0)} = \boldsymbol{0}_{d\times n}$, $\boldsymbol{Q}_x^{(0)} = \boldsymbol{X}^{(0)} = \bar{\boldsymbol{x}}^{(0)}\boldsymbol{1}_n^\top$
3: **for** $t = 1, \ldots, T$ **do**
4: Draw the local mini-batch of size b uniformly at random $\mathcal{Z}^{(t)} = \{\mathcal{Z}_i^{(t)}\}_{i=1}^n$ at each agent i
5: **Option I: PORTER-DP (differentially-private SGD)**
6: $\boldsymbol{G}_\tau^{(t)} = \frac{1}{b}\sum_{\boldsymbol{Z}\in\mathcal{Z}^{(t)}} \mathsf{Clip}_\tau(\nabla\ell(\boldsymbol{X}^{(t-1)};\boldsymbol{Z}))$
7: $\boldsymbol{G}_p^{(t)} = \boldsymbol{G}_\tau^{(t)} + \boldsymbol{E}^{(t)}$, where $\boldsymbol{e}_i^{(t)} \sim \mathcal{N}(\boldsymbol{0}_d, \sigma_p^2\boldsymbol{I}_d)$
8: **Option II: PORTER-GC (gradient clipping SGD)**
9: $\boldsymbol{G}^{(t)} = \frac{1}{b}\sum_{\boldsymbol{Z}\in\mathcal{Z}^{(t)}} \nabla\ell(\boldsymbol{X}^{(t-1)};\boldsymbol{Z})$
10: $\boldsymbol{G}_p^{(t)} = \boldsymbol{G}_\tau^{(t)} = \mathsf{Clip}_\tau(\boldsymbol{G}^{(t)})$
11: $\boldsymbol{Q}_v^{(t)} = \boldsymbol{Q}_v^{(t-1)} + \mathcal{C}(\boldsymbol{V}^{(t-1)} - \boldsymbol{Q}_v^{(t-1)})$
12: $\boldsymbol{V}^{(t)} = \boldsymbol{V}^{(t-1)} + \gamma\boldsymbol{Q}_v^{(t)}(\boldsymbol{W}-\boldsymbol{I}_n) + \boldsymbol{G}_p^{(t)} - \boldsymbol{G}_p^{(t-1)}$
13: $\boldsymbol{Q}_x^{(t)} = \boldsymbol{Q}_x^{(t-1)} + \mathcal{C}(\boldsymbol{X}^{(t-1)} - \boldsymbol{Q}_x^{(t-1)})$
14: $\boldsymbol{X}^{(t)} = \boldsymbol{X}^{(t-1)} + \gamma\boldsymbol{Q}_x^{(t)}(\boldsymbol{W}-\boldsymbol{I}_n) - \eta\boldsymbol{V}^{(t)}$
15: **end for**
16: **output:** $\boldsymbol{x}_{out} \sim \text{Uniform}(\{\boldsymbol{x}_i^{(t)} | i\in[n], t\in[T]\})$.

surrogate $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_v$, and their corresponding agent-wise values are defined analogously. The distributed gradient is defined as $\nabla F(\boldsymbol{X}) = [\nabla f_1(\boldsymbol{x}_1), \nabla f_2(\boldsymbol{x}_2), \ldots, \nabla f_n(\boldsymbol{x}_n)] \in \mathbb{R}^{d\times n}$.

To provide more detail, PORTER initializes gradient-related variables to $\boldsymbol{0}_d$ and other variables to the same random value $\bar{\boldsymbol{x}}^{(0)}$, which improves stability in early iterations and simplifies analysis, but has no impact on the convergence rates. Within each iteration, PORTER is consisted of 3 major steps.

1) *Computing clipped stochastic gradients:* We consider two options. The first option PORTER-DP corresponds to differentially-private SGD (Lines 6-7), where Line 6 computes a batch of clipped stochastic gradient on each agent, and then Line 7 adds Gaussian noise to ensure privacy. The second option PORTER-GC corresponds to SGD with gradient clipping (Lines 9-10), where Line 9 computes a batch stochastic gradient on each agent, and Line 10 applies clipping to each batch stochastic gradient.
2) *Updating gradient estimates:* Line 11 updates the auxiliary variable $\boldsymbol{Q}_v^{(t)}$, which is a compressed surrogate of $\boldsymbol{V}^{(t-1)}$, by adding the compressed difference to itself $\mathcal{C}(\boldsymbol{V}^{(t-1)} - \boldsymbol{Q}_v^{(t-1)})$. Meanwhile, each agent i sends its compressed difference $\mathcal{C}(\boldsymbol{v}_i^{(t-1)} - \boldsymbol{q}_{v,i}^{(t-1)})$ to all of its neighbors, so that every neighbor can reconstruct the auxiliary variable $\boldsymbol{q}_{v,i}^{(t)}$ by accumulating this difference. Line 12 then adds a correction term $\gamma\boldsymbol{Q}_v^{(t)}(\boldsymbol{W}-\boldsymbol{I}_n)$, and applies stochastic gradient tracking to update gradient estimates.
3) *Updating variable estimates:* Similar to updating gradient estimates, Line 13 updates the auxiliary variable $\boldsymbol{Q}_x^{(t)}$, which is a compressed surrogate of variable estimates $\boldsymbol{X}^{(t-1)}$, and communicates with neighbors. Line 14 applies correction and updates the variable estimates by a gradient-style update.

IV. Theoretical Guarantees

This section theoretically analyzes the privacy and convergence properties of PORTER under various assumptions. Section IV-A lists all assumptions required for convergence analysis, Section IV-B shows the privacy and convergence of PORTER-DP using a specific perturbation variance, and Section IV-C shows the convergence of PORTER corresponding to clipped SGD without privacy.

A. Assumptions

We start with smoothness assumption in Assumption 1, which is standard and required for all of our analysis.

Assumption 1 (L-smoothness): For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and any datum $\boldsymbol{z}$ in dataset $\mathcal{Z}$,

$$\|\nabla\ell(\boldsymbol{x};\boldsymbol{z}) - \nabla\ell(\boldsymbol{y};\boldsymbol{z})\|_2 \le L\|\boldsymbol{x}-\boldsymbol{y}\|_2.$$

Note that the gradient clipping operator $\mathsf{Clip}_\tau(\cdot)$ is utilized to ensure gradients are bounded. In addition, the boundedness of the gradient ensures the application of differentially-private mechanisms. However, stochastic gradients at different agents lose correct scaling after clipping, which breaks the stationary point property at local minima and introduces bias. To simplify analysis, one assumption that has been adopted widely in theoretical analysis [11] is the following bounded gradient assumption.

Assumption 2 (Bounded gradient): For any $\boldsymbol{x} \in \mathbb{R}^d$ and any datum $\boldsymbol{z}$ in dataset $\mathcal{Z}$, $\|\nabla\ell(\boldsymbol{x};\boldsymbol{z})\|_2 \le \tau$.

Under Assumption 2, PORTER can skip the clipping operator, and $\boldsymbol{g}_{\tau_i}^{(t)}$ becomes an unbiased estimator of local gradient $\nabla f_i(\boldsymbol{x}_i^{(t)})$, while still allowing privacy analysis. However, Assumption 2 is rather strong and seldomly met in practice. For example, the gradient of a quadratic loss function is not bounded. In addition, it may result in an overly pessimistic clipping operation when there are possibly adversarial gradients with large norms in the samples. Going beyond the strong bounded gradient assumption, we consider a much milder assumption that bounds the local variance as follows, which is more standard in the analysis of unclipped stochastic algorithms.

Assumption 3 (Bounded local variance): For any $\boldsymbol{x} \in \mathbb{R}^d$ and $i \in [n]$,

$$\mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}_i}\|\nabla\ell(\boldsymbol{x};\boldsymbol{z}) - \nabla f_i(\boldsymbol{x})\|_2^2 \le \sigma_g^2.$$

An additional challenge is associated with dealing with the decentralized environment, where the local objective functions can be rather distinct from the global one. Our analysis identifies the following assumption, called bounded gradient dissimilarity, which says that the difference between the local gradient and the global gradient should be small relative to the global one.

Assumption 4 (Bounded gradient dissimilarity): For any $\boldsymbol{x} \in \mathbb{R}^d$ and $i \in [n]$,

$$\left\|\nabla f(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\right\|_2 \leq \frac{1}{12}\left\|\nabla f(\boldsymbol{x})\right\|_2.$$

B. Privacy and Convergence Guarantees of PORTER-DP

We start by analyzing the privacy and convergence guarantees of PORTER-DP, assuming the batch size $b = 1$.

Privacy guarantee: Theorem 1 proves that PORTER-DP is (ϵ, δ)-LDP when setting the variance of Gaussian perturbation properly.

Theorem 1 (Local differential privacy): Let $\phi_m = \frac{\sqrt{d\log(1/\delta)}}{m\epsilon}$ and $b = 1$. For any $\epsilon \leq T/m^2$ and $\delta \in (0,1)$, PORTER-DP is (ϵ, δ)-LDP after T iterations if we set

$$\sigma_p^2 = \frac{T\tau^2\log(1/\delta)}{m^2\epsilon^2} = T\tau^2\phi_m^2/d. \tag{5}$$

Theorem 1 shows that PORTER-DP can achieve (ϵ, δ)-LDP regardless of whether the bounded gradient assumption presents, because using the clipping operator $\mathsf{Clip}_\tau(\cdot)$ can guarantee all the stochastic gradients' ℓ_2 norms are bounded by τ, so that the perturbation variance can be set accordingly.

Convergence with bounded gradient assumption: We start by analyzing the convergence when the gradients are bounded under Assumption 2, in which case PORTER-DP can omit the clipping operator. Theorem 2 presents the convergence result of PORTER-DP using general compression operators (Definition 3).

Theorem 2 (Convergence of PORTER-DP with bounded gradient assumption): Assume Assumptions 1 and 2 hold, and use general compression operators (Definition 3). Let $\Delta = \mathbb{E}[f(\bar{\boldsymbol{x}}^{(0)})] - f^\star$. Set $\gamma = O((1-\alpha)\rho)$, $\eta = O(\gamma^{4/3}\rho^{4/3}\phi_m)$, $T = \phi_m^{-2}$, $b = 1$ and $\sigma_p^2 = T\tau^2\phi_m^2/d$. PORTER-DP converges in average squared ℓ_2 gradient norm as

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\|_2^2 \lesssim \frac{\phi_m}{\rho^{\frac{4}{3}}(1-\alpha)^{\frac{8}{3}}} \cdot \max\left\{\tau^2, L\Delta\right\}. \tag{6}$$

Theorem 2 shows the convergence error of the squared ℓ_2 gradient norm with explicit dependency on the compression ratio ρ and mixing rate α. When we fix ρ and α, Theorem 2 reaches an $O(\phi_m)$ final average squared ℓ_2 gradient norm, which matches the result of SoteriaFL-SGD [11], the state-of-the-art stochastic algorithm with local differential privacy guarantees and *unbiased* communication compression in the server-client setting. However, due to extra complexities induced by the decentralized setting and *biased* compression, PORTER-DP takes $O(\phi_m^{-2})$ iterations to converge while SoteriaFL-SGD only takes $O(\phi_m^{-1})$ iterations; in addition, PORTER-DP has a slightly worse dependency on the compression ratio ρ.

Convergence without bounded gradient assumption: A more interesting and challenging scenario is when Assumption 2 does not hold, PORTER-DP applies gradient clipping to ensure gradients are bounded to suit the privacy constraints. Fortunately, Theorem 3 describes the convergence behavior of Algorithm 1 in this case, under the much weaker bounded local variance and bounded dissimilarity assumptions.

Theorem 3 (Convergence of PORTER-DP without bounded gradient assumption): Assume Assumptions 1, 3 and 4 hold, and use general compression operators (Definition 3). Let $\Delta = \mathbb{E}[f(\bar{\boldsymbol{x}}^{(0)})] - f^\star$. Set $\tau = \max\{365\rho^{-\frac{4}{3}}(1-\alpha)^{-\frac{8}{3}}\phi_m^{1/2}, 24\sigma_g\}$, $\gamma = O((1-\alpha)\rho)$, $\eta = O(L^{-1})$, $b = 1$, $T = \phi_m^{-2}$ and $\sigma_p^2 = T\tau^2\phi_m^2/d$. Algorithm 1 converges in minimum ℓ_2 gradient norm as

$$\min_{t\in[T]}\mathbb{E}\left\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\right\|_2 \lesssim \max\left\{\rho^{-\frac{4}{3}}(1-\alpha)^{-\frac{8}{3}}(L\Delta\phi_m)^{1/2}, \sigma_g\right\}. \tag{7}$$

Theorem 3 shows PORTER-DP converges in minimum ℓ_2 gradient norm with explicit dependency on compression ratio ρ, mixing rate α and gradient variance σ_g, under much weaker assumptions. To compare with Theorem 2, we can take the square root of (6), which translates to minimum ℓ_2 gradient norm convergence on the order of $O(\rho^{-2/3}(1-\alpha)^{-4/3}\phi_m^{1/2}\cdot\max\{\tau, (L\Delta)^{1/2}\})$. In comparison, although Theorem 3 has worse dependency on compression ratio ρ and mixing rate α, it matches the dependency on the baseline privacy loss ϕ_m.

C. Convergence Guarantees of PORTER-GC

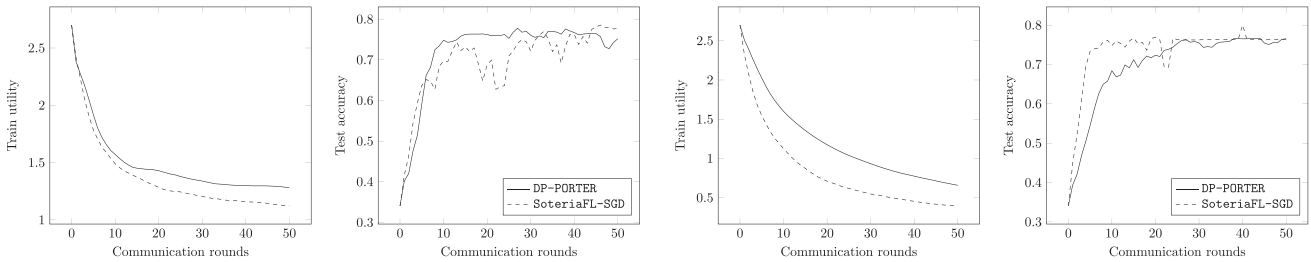
Theorem 4 further establishes the convergence of PORTER-GC without the bounded gradient assumption, which applies the clipping operator to mini-batch stochastic gradients without privacy perturbation.

Theorem 4 (Convergence of PORTER-GC without bounded gradient assumption): Assume Assumptions 1, 3 and 4 hold, and use general compression operators (Definition 3). Let $\Delta = \mathbb{E}[f(\bar{\boldsymbol{x}}^{(0)})] - f^\star$. Set $\tau = O(\rho^{-\frac{2}{3}}(1-\alpha)^{-\frac{4}{3}}T^{-1/2})$, $\gamma = O((1-\alpha)\rho)$, $\eta = O(L^{-1})$ and $b = O(\sigma_g^2\rho^{-\frac{4}{3}}(1-\alpha)^{-\frac{8}{3}}T^{-1})$. PORTER-GC converges in minimum ℓ_2 gradient norm as

$$\min_{t\in[T]}\mathbb{E}\left\|\nabla f(\bar{\boldsymbol{x}}^{(t)})\right\|_2 \lesssim \frac{1}{\rho^{\frac{2}{3}}(1-\alpha)^{\frac{4}{3}}} \cdot \frac{1}{T^{1/2}}.$$

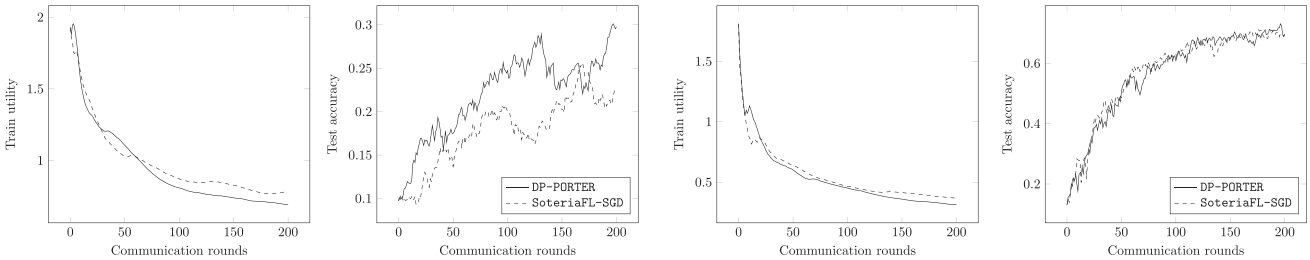
Theorem 4 suggests that by picking the clipping threshold τ and batch size b properly, PORTER-GC converges at an $O(1/T^{1/2})$ rate. In comparison, assuming gradients are bounded by $O(1/T^{1/2})$, the gradient clipping threshold of Theorem 4, standard centralized SGD converges in average squared ℓ_2 gradient norm at an $O(1/T)$ rate, which also translates to a minimum ℓ_2 gradient norm convergence in the form of $\min_{t\in[T]}\mathbb{E}\left\|\nabla f(\boldsymbol{x}^{(t)})\right\|_2 \lesssim 1/T^{1/2}$. Therefore, roughly speaking, using gradient clipping for decentralized SGD does not affect the convergence rate, providing proper hyper parameter choices.

When the gradients are bounded, we can omit the clipping operator in PORTER-GC, which become the same as BEER [3]. Recall that BEER guarantees a minimum ℓ_2 gradient norm convergence at the rate $\frac{1}{\rho^{1/2}(1-\alpha)^{3/2}}\cdot\frac{1}{T^{1/2}}$. In comparison,



(a) $(10^{-2}, 10^{-3})$-LDP (b) $(10^{-1}, 10^{-3})$-LDP

Fig. 2. The train utility and test accuracy vs. communication rounds for logistic regression with nonconvex regularization on the a9a dataset when guaranteeing $(10^{-2}, 10^{-3})$-LDP and $(10^{-1}, 10^{-3})$-LDP, respectively. Both PORTER-DP and SoteriaFL-SGD employ random$_{162}$ compression (cf. Example 1).



(a) $(10^{-2}, 10^{-3})$-LDP (b) $(10^{-1}, 10^{-3})$-LDP.

Fig. 3. The train utility and test accuracy vs. communication rounds for training a one-hidden-layer neural network on the MNIST dataset when guaranteeing $(10^{-2}, 10^{-3})$-LDP and $(10^{-1}, 10^{-3})$-LDP, respectively. Both PORTER-DP and SoteriaFL-SGD employ random$_{2583}$ compression (cf. Example 1).

Theorem 4 has a better dependency on the mixing rate α, but has a slightly worse dependency on the compression ratio ρ, which again emphasizes that gradient clipping does not harm convergence.

V. Numerical Experiments

This section presents numerical experiments to examine the performance of PORTER-DP, with comparison to the state-of-the-art server/client private stochastic algorithm SoteriaFL-SGD, which also utilizes communication compression and guarantees local differential privacy. More specifically, we evaluate the convergence of utility and accuracy in terms of communication rounds and communication bits, to analyze the privacy-utility-communication trade-offs of different algorithms.

For all experiments, we split shuffled datasets evenly to 10 agents that are connected by an Erdős-Rényi random graph with connecting probability $p = 0.8$. We use the FDLA matrix [17] as the mixing matrix to perform weighted information aggregation to accelerate convergence. We use biased random sparsification (cf. Example 1) for all algorithms where $k = \lfloor \frac{d}{20} \rfloor$, i.e., the compressor randomly selects 5% elements from each vector. We also apply gradient clipping with $\tau = 1$ to all algorithms for simplicity. For each experiment, all algorithms are initialized to the same starting point, and use best-tuned learning rates, batch size 1 and $\sigma_p = \frac{\tau\sqrt{T\log(1/\delta)}}{m\epsilon}$.

A. Logistic Regression With Nonconvex Regularization

We run experiments on logistic regression with nonconvex regularization on the a9a dataset [79]. Following [80], the objective function is defined as

$$\ell(\boldsymbol{x}; \{\boldsymbol{f}, l\}) = \log\left(1 + l\exp(-\boldsymbol{x}^\top \boldsymbol{f})\right) + \lambda \sum_{i=1}^{d} \frac{x_i^2}{1 + x_i^2},$$

where $\{\boldsymbol{f}, l\}$ represents a training tuple, $\boldsymbol{f} \in \mathbb{R}^d$ is the feature vector and $l \in \{0, 1\}$ is the label, and λ is the regularization parameter which is set to $\lambda = 0.2$.

Fig. 2 shows the convergence results of PORTER-DP and SoteriaFL-SGD for logistic regression with nonconvex regularization on the a9a dataset to reach $(10^{-2}, 10^{-3})$-LDP and $(10^{-1}, 10^{-3})$-LDP, respectively. Under $(10^{-2}, 10^{-3})$-LDP, which is a stricter privacy setting, PORTER-DP converges faster than SoteriaFL-SGD in test accuracy, while converges slightly slower in train utility. Under $(10^{-1}, 10^{-3})$-LDP, PORTER-DP performs slightly worse than SoteriaFL-SGD. Given that PORTER-DP operates under the decentralized topology with much weaker information exchange, the results highlight PORTER-DP's communication efficiency by showing it can achieve similar performance as its server/client counterpart, i.e. SoteriaFL-SGD, especially under strict privacy constraints.

B. One-Hidden-Layer Neural Network Training

We evaluate PORTER-DP by training a one-hidden layer neural network on the MNIST dataset [81]. The network uses 64 hidden neurons, sigmoid activation functions and cross-entropy loss, where the loss function over a training pair is defined as

$$\ell(\boldsymbol{x};(\boldsymbol{f},l)) = \mathsf{CrossEntropy}(\mathsf{softmax}(\boldsymbol{W}_2\,\mathsf{sigmoid}(\boldsymbol{W}_1\boldsymbol{f}+\boldsymbol{c}_1)+\boldsymbol{c}_2),l).$$

Here, the model parameter is defined by $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{W}_1,\boldsymbol{c}_1,\boldsymbol{W}_2,\boldsymbol{c}_2)$, where the dimensions of the network parameters $\boldsymbol{W}_1$, $\boldsymbol{c}_1$, $\boldsymbol{W}_2$, $\boldsymbol{c}_2$ are 64×784, 64×1, 10×64, and 10×1, respectively. In addition, the training pair $\{\boldsymbol{f},l\}$ are specified with $\boldsymbol{f}\in\mathbb{R}^{768}$ and $l\in\{0,\ldots,9\}$.

Fig. 3 shows the convergence results of PORTER-DP and SoteriaFL-SGD for training a one-hidden-layer neural network on the MNIST dataset to reach $(10^{-2},10^{-3})$-LDP and $(10^{-1},10^{-3})$-LDP, respectively. Under both privacy settings, PORTER-DP converges at a similar rate as SoteriaFL-SGD in train utility. However, in terms of convergence in test accuracy, PORTER-DP outperforms SoteriaFL-SGD under the stricter $(10^{-2},10^{-3})$-LDP, while the two algorithms have similar performance under the other setting. This experiment again emphasizes PORTER-DP's communication efficiency in comparison to the server/client algorithm SoteriaFL-SGD.

VI. Conclusion

In this paper, we propose an algorithmic framework called PORTER, which incorporates practically-relevant gradient clipping and communication compression simultaneously in the design of decentralized nonconvex optimization algorithms. We propose two variants: PORTER-DP and PORTER-GC. While they share a similar structure that makes use of gradient tracking, communication compression, and error feedback, their focuses are on different perspectives motivated by applications in privacy preserving and neural network training, respectively. PORTER-DP adds privacy perturbation to clipped gradients to protect the local differential privacy of each agent, with explicit utility and communication complexities. PORTER-GC applies gradient clipping to mini-batch stochastic gradients, which converges in minimum ℓ_2 gradient norm at similar rate as centralized SGD without clipping under proper choices of hyperparameters. The development of PORTER offers a simple analysis framework to understand gradient clipping in decentralized nonconvex optimization without bounded gradient assumptions, highlighting the potential of achieving both communication efficiency and privacy preserving in the decentralized framework.

References

[1] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. 36th Int. Conf. Mach. Learn.*, Jun. 2019, pp. 3478–3487.

[2] N. Singh, D. Data, J. George, and S. Diggavi, "SQuARM-SGD: Communication-efficient momentum SGD for decentralized optimization," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 3, pp. 954–969, Sep. 2021.

[3] H. Zhao, B. Li, Z. Li, P. Richtárik, and Y. Chi, "BEER: Fast O (1/T) rate for decentralized nonconvex optimization with communication compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, Art. no. 2295.

[4] H. Tang et al., "Deepsqueeze: Decentralization meets error-compensated compression," 2019, *arXiv:1907.07346*.

[5] K. Huang and S. Pu, "CEDAS: A compressed decentralized stochastic gradient method with improved convergence," *IEEE Trans. Autom. Control*, in press, 2024.

[6] Y. Yan, J. Chen, P.-Y. Chen, X. Cui, S. Lu, and Y. Xu, "Compressed decentralized proximal stochastic gradient method for nonconvex composite problems with heterogeneous data," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 39035–39061.

[7] M. Abadi et al., "Deep learning with differential privacy," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.

[8] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, Berlin, Germany, 2008, pp. 1–19.

[9] J. Zhang, T. He, S. Sra, and A. Jadbabaie, "Why gradient clipping accelerates training: A theoretical justification for adaptivity," 2019, *arXiv:1905.11881*.

[10] L. Wang, B. Jayaraman, D. Evans, and Q. Gu, "Efficient privacy-preserving stochastic nonconvex optimization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2023, pp. 2203–2213.

[11] Z. Li, H. Zhao, B. Li, and Y. Chi, "SoteriaFL: A unified framework for private federated learning with communication compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 4285–4300.

[12] X. Yang, H. Zhang, W. Chen, and T.-Y. Liu, "Normalized/clipped SGD with perturbation for differentially private non-convex optimization," 2022, *arXiv:2206.13033*.

[13] B. Zhang, J. Jin, C. Fang, and L. Wang, "Improved analysis of clipping algorithms for non-convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15511–15521.

[14] M. Nokleby, H. Raja, and W. U. Bajwa, "Scaling-up distributed processing of data streams for machine learning," in *Proc. IEEE*, vol. 108, no. 11, pp. 1984–2012, Nov. 2020.

[15] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," in *Proc. IEEE*, vol. 108, no. 11, pp. 1869–1889, Nov. 2020.

[16] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annu. IEEE Symp. Found. Comput. Sci.*, Oct. 2003, pp. 482–491.

[17] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[18] D. Shah, "Gossip algorithms," *Found. Trends Netw.*, vol. 3, no. 1, pp. 1–125, 2007.

[19] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 391–405, Feb. 2013.

[20] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5330–5340.

[21] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 344–353.

[22] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 213:9709–213:9758, Jan. 2021.

[23] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[24] P. Di Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[25] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[26] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[27] B. Li, S. Cen, Y. Chen, and Y. Chi, "Communication-efficient distributed optimization in networks with gradient tracking and variance reduction," *J. Mach. Learn. Res.*, vol. 21, no. 180, pp. 1–51, 2020.

[28] H. Sun, S. Lu, and M. Hong, "Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 9217–9228.

[29] R. Xin, U. A. Khan, and S. Kar, "A fast randomized incremental gradient method for decentralized nonconvex optimization," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5150–5165, Oct. 2022.
[30] R. Xin, U. A. Khan, and S. Kar, "Fast decentralized nonconvex finite-sum optimization with recursive variance reduction," *SIAM J. Optim.*, vol. 32, no. 1, pp. 1–28, Mar. 2022.
[31] B. Li, Z. Li, and Y. Chi, "DESTRESS: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization," *SIAM J. Math. Data Sci.*, vol. 4, no. 3, pp. 1031–1051, Sep. 2022.
[32] Y. Huang, Y. Sun, Z. Zhu, C. Yan, and J. Xu, "Tackling data heterogeneity: A new unified framework for decentralized SGD with sample-induced topology," in *Proc. 39th Int. Conf. Mach. Learn.*, Jun. 2022, pp. 9310–9345.
[33] L. Luo and H. Ye, "An optimal stochastic algorithm for decentralized nonconvex finite-sum optimization," 2022, *arXiv:2210.13931*.
[34] C. M. De Sa, C. Zhang, K. Olukotun, and C. Ré, "Taming the wild: A unified analysis of HOG WILD-style algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2674–2682.
[35] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.
[36] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Sep. 2014, pp. 1058–1062.
[37] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4452–4463.
[38] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5977–5987.
[39] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik, "Distributed learning with compressed gradient differences," *Optim. Methods Softw.*, pp. 1–16, 2024.
[40] Z. Li, D. Kovalev, X. Qian, and P. Richtarik, "Acceleration for compressed gradient descent in distributed and federated optimization," in *Proc. 37th Int. Conf. Mach. Learn.*, Nov. 2020, pp. 5895–5904.
[41] E. Gorbunov, K. P. Burlachenko, Z. Li, and P. Richtarik, "MARINA: Faster non-convex distributed learning with compression," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 3788–3798.
[42] Z. Li and P. Richtarik, "CANITA: Faster rates for distributed convex optimization with communication compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 13770–13781.
[43] P. Richtarik, I. Sokolov, and I. Fatkhullin, "EF21: A new, simpler, theoretically better, and practically faster error feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 4384–4396.
[44] I. Fatkhullin, I. Sokolov, E. Gorbunov, Z. Li, and P. Richtárik, "EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback," 2021, *arXiv:2110.03294*.
[45] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7652–7662.
[46] H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized decentralized stochastic learning over directed graphs," in *Proc. 37th Int. Conf. Mach. Learn.*, Nov. 2020, pp. 9324–9333.
[47] Y. Liao, Z. Li, and S. Pu, "A linearly convergent robust compressed push-pull method for decentralized optimization," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, 2023, pp. 4156–4161.
[48] H. Zhao, K. Burlachenko, Z. Li, and P. Richtárik, "Faster rates for compressed federated learning with client-variance reduction," *SIAM J. Math. Data Sci.*, vol. 6, no. 1, pp. 154–175, 2024.
[49] P. Valdeira, J. Xavier, C. Soares, and Y. Chi, "Communication-efficient vertical federated learning via compressed error feedback," 2024, *arXiv:2406.14420*.
[50] S. Chen, Z. Li, and Y. Chi, "Escaping saddle points in heterogeneous federated learning via distributed SGD with communication compression," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2024, pp. 2701–2709.
[51] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.*, Berlin, Germany, 2006, pp. 265–284.
[52] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2719–2728.
[53] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang, "Towards practical differentially private convex optimization," in *Proc. 2019 IEEE Symp. Secur. Privacy*, May 2019, pp. 299–316.
[54] V. Feldman, T. Koren, and K. Talwar, "Private stochastic convex optimization: Optimal rates in linear time," in *Proc. 52nd Annu. ACM SIGACT Symp. Theory Comput.*, New York, NY, USA, Jun. 2020, pp. 439–449.
[55] X. Chen, S. Z. Wu, and M. Hong, "Understanding gradient clipping in private SGD: A geometric perspective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 13773–13782.
[56] D. Wang, H. Xiao, S. Devadas, and J. Xu, "On differentially private stochastic convex optimization with heavy-tailed data," in *Proc. 37th Int. Conf. Mach. Learn.*, Nov. 2020, pp. 10081–10091.
[57] X. Huang, Y. Ding, Z. L. Jiang, S. Qi, X. Wang, and Q. Liao, "DP-FL: A novel differentially private federated learning framework for the unbalanced data," *World Wide Web*, vol. 23, no. 4, pp. 2529–2545, Jul. 2020.
[58] S. Asoodeh, W.-N. Chen, F. P. Calmon, and A. Özgür, "Differentially private federated learning: An information-theoretic perspective," in *Proc. 2021 IEEE Int. Symp. Inf. Theory*, Jul. 2021, pp. 344–349.
[59] M. Noble, A. Bellet, and A. Dieuleveut, "Differentially private federated learning on heterogeneous data," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, May 2022, pp. 10110–10145.
[60] J. Du, S. Li, X. Chen, S. Chen, and M. Hong, "Dynamic differential-privacy preserving SGD," 2022, *arXiv:2111.00173*.
[61] T. Murata and T. Suzuki, "DIFF2: Differential private optimization via gradient differences for nonconvex distributed learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 25523–25548.
[62] X. Zhang, X. Chen, M. Hong, S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *Proc. 39th Int. Conf. Mach. Learn.*, Jun. 2022, pp. 26048–26067.
[63] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. 30th Int. Conf. Mach. Learn.*, May 2013, pp. 1310–1318.
[64] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proc. 2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2013, pp. 8624–8628.
[65] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.
[66] S. Kanai, Y. Fujiwara, and S. Iwamura, "Preventing gradient explosions in gated recurrent units," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 435–444.
[67] Y. You, I. Gitman, and B. Ginsburg, "Scaling SGD batch size to 32k for ImageNet training," Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2017-156, Sep. 2017. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-156.html
[68] J. Zhang, T. He, S. Sra, and A. Jadbabaie, "Why gradient clipping accelerates training: A theoretical justification for adaptivity," in *Proc. Int. Conf. Learn. Representations*, Mar. 2020.
[69] J. Zhang et al., "Why are adaptive methods good for attention models?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15383–15393.
[70] A. Reisizadeh, H. Li, S. Das, and A. Jadbabaie, "Variance-reduced clipping for non-convex optimization," 2023, *arXiv:2303.00883*.
[71] R. Das, S. Kale, Z. Xu, T. Zhang, and S. Sanghavi, "Beyond uniform lipschitz condition in differentially private optimization," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 7066–7101.
[72] H. Fang, X. Li, C. Fan, and P. Li, "Improved convergence of differential private SGD with gradient clipping," in *Proc. 11th Int. Conf. Learn. Representations*, 2023.
[73] A. Koloskova, H. Hendrikx, and S. U. Stich, "Revisiting gradient clipping: Stochastic bias and tight convergence guarantees," in *Proc. 40th Int. Conf. Mach. Learn.*, Jul. 2023, Art. no. 714.
[74] B. Li and Y. Chi, "Convergence and privacy of decentralized nonconvex optimization with gradient clipping and communication compression," 2023, *arXiv:2305.09896*.
[75] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," in *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.
[76] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, Oct. 2013, pp. 429–438.
[77] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Proc. Int. Sump. Privacy Enhancing Technol. Symp.*, Berlin, Germany, 2013, pp. 82–102.
[78] H. Xiao, Y. Ye, and S. Devadas, "Local differential privacy in decentralized optimization," 2019, *arXiv:1902.06101*.

[79] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011, doi: 10.1145/1961189.1961199.
[80] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, "SpiderBoost and momentum: Faster variance reduction algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2406–2416.
[81] Y. LeCun et al., "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural Netw., Statist. Mechan. Perspective*, vol. 261, no. 276, p. 2, 1995.

Boyue Li received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2016, and the M.S. and Ph.D. degrees from Carnegie Mellon University, Pittsburgh, PA, USA, in 2018 and 2023, respectively, advised by Prof. Yuejie Chi. His research interests include distributed optimization and differential privacy, focusing on developing methodologies that enhance efficiency of large-scale distributed machine learning systems and protection of sensitive data.

Yuejie Chi (Fellow, IEEE) received the B.E. (Hons.) degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, USA, in 2009 and 2012, respectively. She is currently the Sense of Wonder Group Endowed Professor of electrical and computer engineering in AI systems with Carnegie Mellon University, Pittsburgh, PA, USA, with courtesy appointments with the Machine Learning Department and CyLab. Her research interests include the theoretical and algorithmic foundations of data science, signal processing, machine learning and inverse problems, with applications in sensing, imaging, decision making, and AI systems. She is an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE, *Information and Inference: A Journal of the IMA*, and *SIAM Journal on Mathematics of Data Science*. Among others, she was the recipient of Presidential Early Career Award for Scientists and Engineers (PECASE), SIAM Activity Group on Imaging Science Best Paper Prize, IEEE Signal Processing Society Young Author Best Paper Award, and inaugural IEEE Signal Processing Society Early Career Technical Achievement Award for contributions to high-dimensional structured signal processing. She was named a Goldsmith Lecturer by IEEE Information Theory Society, Distinguished Lecturer by IEEE Signal Processing Society, and Distinguished Speaker by ACM.