DECENTRALIZED NONCONVEX OPTIMIZATION UNDER HEAVY-TAILED NOISE: NORMALIZATION AND OPTIMAL CONVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Heavy-tailed noise in nonconvex stochastic optimization has garnered increasing research interest, as empirical studies, including those on training attention models, suggest it is a more realistic gradient noise condition. This paper studies first-order nonconvex stochastic optimization under heavy-tailed gradient noise in a decentralized setup, where each node can only communicate with its direct neighbors in a predefined graph. Specifically, we consider a class of heavy-tailed gradient noise that is zero-mean and has only p-th moment for $p \in (1, 2]$. We propose `GT-NSGDm`, Gradient Tracking based Normalized Stochastic Gradient Descent with momentum, that utilizes normalization, in conjunction with gradient tracking and momentum, to cope with heavy-tailed noise on distributed nodes. We show that, when the communication graph admits primitive and doubly stochastic weights, `GT-NSGDm` guarantees, for the *first* time in the literature, that the expected gradient norm converges at an *optimal non-asymptotic rate* $O\big(1/T^{(p-1)/(3p-2)}\big)$, which matches the lower bound in the centralized setup. When tail index p is unknown, `GT-NSGDm` attains a non-asymptotic rate $O\big(1/T^{(p-1)/(2p)}\big)$ that is, for $p < 2$, topology independent and has a speedup factor $n^{1-1/p}$ in terms of the number of nodes n. Finally, experiments on nonconvex linear regression with tokenized synthetic data and decentralized training of language models on a real-world corpus demonstrate that `GT-NSGDm` is more robust and efficient than baselines.

1 INTRODUCTION

In this paper, we address the problem of nonconvex stochastic optimization under heavy-tailed gradient noise in the decentralized setup. Consider a graph with n nodes connected by a predefined topology $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{1, \dots, n\}$ is the set of node indices, and $\mathcal{E}$ is the collection of directed pairs (i, r), $i, r \in \mathcal{V}$ such that node i can send information to the neighboring node r. Each node $i \in \mathcal{V}$ holds a local nonconvex differentiable cost function $f_i : \mathbb{R}^d \to \mathbb{R}$, and can access its stochastic gradient, subject to zero mean noise with a bounded p-th moment for some $p \in (1, 2]$. Cooperatively, these nodes aim to solve $\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) := (1/n) \sum_{i=1}^n f_i(\boldsymbol{x})$, through local computation and peer-to-peer communication only with their immediate neighbors.

Decentralized optimization in the above formulation has been studied for decades (Tsitsiklis et al., 1986), and has recently attracted growing research interest due to its advantages in scalability and privacy preservation across a wide range of distributed machine learning, signal processing, and control tasks over networks (Nedić et al., 2018; Li et al., 2020; Kairouz et al., 2021). For instance, in privacy-sensitive applications such as those in the medical domain (Brisimi et al., 2018), training data are often distributed across n nodes due to privacy constraints. In such cases, each f_i represents an empirical risk function, e.g., a neural network, defined over the local dataset on node i, and all nodes collaboratively train a global predictive model via peer-to-peer communication without sharing raw data. Moreover, decentralized optimization is also employed in data centers to reduce communication bottlenecks associated with the central node in traditional centralized training paradigms (Lian et al., 2017).

In decentralized optimization, first-order methods are widely favored for their simplicity and scalability (Xin et al., 2020b). However, computing the exact gradient of each local objective function f_i at every iteration can be computationally expensive, particularly in large-scale settings where each node holds a substantial volume of local data. To alleviate this computational burden, decentralized stochastic gradient methods, which approximate exact gradients, have been extensively studied. Most existing approaches, including decentralized stochastic (sub)gradient descent (Sundhar Ram et al., 2010; Koloskova et al., 2020; Wang & Joshi, 2021), variance reduction techniques (Yuan et al., 2018), and gradient tracking-based schemes (Di Lorenzo & Scutari, 2016; Pu & Nedić, 2021), typically assume that stochastic gradient noise has a *finite variance*. Nevertheless, recent empirical and theoretical evidence indicates that, when optimizing certain neural network architectures, especially attention-based models such as Transformers (Vaswani et al., 2017), the gradient noise often follows a *heavy-tailed distribution*[1] with significantly large or even *infinite variance* (Simsekli et al., 2019; Zhang et al., 2020; Gorbunov et al., 2020; Gurbuzbalaban et al., 2021; Ahn et al., 2024; Kunstner et al., 2024). The presence of heavy-tailed gradient noise poses substantial challenges for existing methods. Empirically, some stochastic gradient descent (SGD) based methods can suffer from instability and even dramatic drop of training accuracies (Zhang et al., 2020; Charles et al., 2021; Yang et al., 2022), particularly in distributed large-cohort training. Theoretically, unbounded variance renders many established analyses invalid, and in *centralized* settings it necessitates the use of nonlinear adaptive techniques such as clipping, sign, and normalization (Zhang et al., 2020; Sadiev et al., 2023; Compagnoni et al., 2025b; Hübler et al., 2024; Liu & Zhou, 2025; Armacki et al., 2025) to combat the strong noise. However, incorporating such adaptive strategies in *decentralized* algorithms introduces inherent *nonlinearity* into the algorithmic dynamics associated with the average-sum structured function f, making the design and analysis of decentralized algorithms under heavy-tailed noise significantly more challenging.

Decentralized optimization under heavy-tailed gradient noise remains underexplored. To the best of our knowledge, only recent studies Sun & Chen (2024); Yu et al. (2023) have attempted to address this problem under restrictive assumptions. Specifically, Sun & Chen (2024) considers zero-mean gradient noise with bounded p-th central moment ($p \in (1, 2]$) similar to our setting but assumes a *compact* domain or *bounded gradients*. Their proposed decentralized gradient descent method with ℓ_2 gradient clipping achieves almost sure convergence for *strongly convex* local functions. However, the restrictive compact domain or gradients assumption in Sun & Chen (2024) limits its practical applicability, and the convergence rate is not explicitly provided. Another work, Yu et al. (2023), also assumes strongly convex local objectives and develops a decentralized gradient method with smoothed clipping and error feedback under gradient noise that is zero-mean, *symmetric*, and has bounded first absolute moment, showing an *in-expectation* convergence rate of $1/t^{\delta}$ for some $\delta \in (0, 2/5)$. Although the noise assumption in Yu et al. (2023) is weaker than ours (as it requires only a first-moment bound), the additional assumptions of *noise symmetry* and the dependence of the rate exponent δ on both the problem dimension and condition number restrict its general applicability. Moreover, both works Sun & Chen (2024); Yu et al. (2023) assume strong convexity, whereas many practical optimization problems involving heavy-tailed noise, particularly in modern machine learning, are inherently nonconvex. Further, the convergence rates in Sun & Chen (2024); Yu et al. (2023) are either unclear or sub-optimal, even compared to the optimal iteration complexity bound $O\big(1/T^{(p-1)/(3p-2)}\big)$ for general nonconvex functions. In this work, we relax these restrictive assumptions and address the following question:

*Can we design a decentralized algorithm for **nonconvex** optimization under general zero-mean gradient noise with only a finite p-th moment for $p \in (1, 2]$ with **optimal iteration complexity**?*

1.1 Contributions

We answer this question affirmatively through the following key contributions:

- We develop a decentralized method, called `GT-NSGDm`, using normalization, coupled with momentum variance reduction, to combat heavy-tailed noise, and using gradient tracking

[1] A random variable X is called heavy-tailed if it exhibits a heavier tail than any exponential distribution; formally, for any constant $a > 0$, $\limsup_{x\to\infty} \mathbb{P}(X > x)e^{ax} = \infty$ (Nair et al., 2022). While some heavy-tailed distributions, such as log-normal and Weibull, still have bounded variance, this paper also considers the sub-class of heavy tailed gradient noise that may have unbounded (infinite) variance such as α-stable noise.

to handle cross-node heterogeneity. To further shed light on the design of GT−NSGDm, we provide a negative result for a vanilla variant of normalized decentralized SGD that employs no gradient tracking nor momentum.

- For general nonconvex and smooth local functions f_i's that are bounded from below, we show that GT−NSGDm converges in expectation at a rate $O\big(1/T^{(p-1)/(3p-2)}\big)$, which matches the lower bound in centralized setting and is order-optimal. Our convergence rate significantly improves upon related works (Sun & Chen, 2024; Yu et al., 2023), which assume strong convexity and lack an explicit rate exponent.
- When the tail index p is unknown, GT−NSGDm achieves a rate of $O(1/T^{(p-1)/(2p)})$, matching the best-known rate in the centralized setting without requiring knowledge of p. Notably, for $p \in (1, 2)$ and sufficiently large T, this rate is *independent* of the network topology and exhibits a *speedup* in the number of nodes, with a factor of $n^{1-1/p}$.
- We test our theoretical findings in nonconvex linear regression models on a synthetic dataset that is built to simulate language tokens under controlled heavy-tailed noise injections. We also test GT−NSGDm on distributed training of decoder-only Transformer models on Multi30k datasets (Elliott et al., 2016). Experiments on multiple variants of network topologies show that GT−NSGDm is more robust to injected and empirical heavy-tailed noise and converges faster.

1.2 Related Work

Heavy-tailed gradient noise. Recent empirical studies suggest that the distribution of gradient noise in training various deep learning models resembles heavy-tailed distributions, such as Lévy's α-stable distribution (Simsekli et al., 2019; Zhang et al., 2020; Barsbey et al., 2021; Battash et al., 2024). For instance, the work Zhang et al. (2020) demonstrates that the empirical distribution of gradient norm samples during BERT pre-training closely aligns with an α-stable distribution, rather than a Gaussian one (see their Figure 1). The presence of heavy-tailed gradient noise is also supported by theoretical insights (Simsekli et al., 2019; Peluchetti et al., 2020; Gurbuzbalaban et al., 2021; Barsbey et al., 2021). In particular, Simsekli et al. (2019) leverages generalized central limit theorems to show that the gradient noise in SGD can converge to an α-stable random variable.

Adaptive methods. Under heavy-tailed noise, vanilla SGD based methods are shown to suffer from slower convergence or even model collapses in *centralized* settings (Zhang et al., 2020) as well as *distributed settings with a central server* (Yang et al., 2022; Lee et al., 2025), and adaptive methods such as clipping and normalization are introduced to stabilize training dynamics. In *centralized* settings, the work Zhang et al. (2020) provides lower bounds for both nonconvex and strongly convex smooth functions, showing that SGD with gradient clipping achieves *in-expectation* upper bounds matching lower bounds. In Sadiev et al. (2023); Liu et al. (2023); Nguyen et al. (2023); Chezhegov et al. (2024), the authors show that when equipped with gradient clipping, SGD, accelerated methods, AdaGrad (Duchi et al., 2011), and Adam (Kingma & Ba, 2014) can achieve (near-)optimal *high-probability* convergence under various function assumptions. Besides, the work Compagnoni et al. (2025b) shows that signSGD is also robust to heavy-tailed noise through the lens of stochastic differential equations. Further, SGD with gradient normalization, which advantageously requires less hyper-parameter tuning than clipping, is shown to achieve optimal *in-expectation* convergence (Hübler et al., 2024; Liu & Zhou, 2025; Sun et al., 2024). Our method incorporates the same normalization and variance reduction approach as Liu & Zhou (2025). Notably, in another line of works Jakovetić et al. (2023); Armacki et al. (2025; 2024), the authors conduct a unified convergence analyses for generic nonlinear methods including clipping, sign, and normalization under *symmetric* noise with positive probability mass around zero without assuming any noise moment bound or only assuming first absolute noise moment bound. In *distributed settings with a server*, the work Gorbunov et al. (2024) proposes an algorithm that incorporates an error feedback mechanism, wherein clipping is applied to the discrepancy between a local gradient estimator and a stochastic gradient, and establishes optimal *high-probability* bounds. Moreover, the work Compagnoni et al. (2025a) shows that distributed signSGD converges to an asymptotic neighborhood depending on the 'fatness' of noise tail. When multiple local updates are permitted between communication rounds, the authors of Yang et al. (2022) show that clipping per local step achieves order-optimal in-expectation convergence, albeit under a restrictive *bounded gradient* assumption. More recently, the work Lee et al. (2025) introduces the TailOPT framework, which adaptively leverages gradient geometry by

applying clipping operators during local updates on distributed nodes and utilizing adaptive optimizers for global updates at the server, achieving in-expectation sublinear convergence rates that are independent of the moment parameter p.

Nonlinearities in decentralized optimization. Extending existing methods that are robust to heavy-tailed noise, whether developed for *centralized* settings or *distributed settings with a server*, to decentralized environments is highly nontrivial, primarily due to the *nonlinearities* introduced to peer-to-peer communication. This difficulty is reflected in that existing decentralized methods incorporating nonlinear adaptive techniques for other purposes often impose restrictive conditions (Yu & Kar, 2023; Li & Chi, 2025). For example, to achieve differential privacy through gradient clipping, the work Li & Chi (2025) establishes convergence in decentralized setups under the assumption of either *bounded gradient* or a *stringent similarity* condition, namely $\|\nabla f_i(\boldsymbol{x})-\nabla f(\boldsymbol{x})\| \leq (1/12)\|\nabla f(\boldsymbol{x})\|$ for all $i \in [n]$ and all $\boldsymbol{x}$. Similarly, to attain adversarial robustness against gradient attacks, the authors of Yu & Kar (2023) employ gradient clipping with momentum, assuming that all local functions are convex, share a *common minimizer*, and that $\sum_{i=1}^n f_i$ is strongly convex. In this work, we significantly relax these conditions and demonstrate the effective use of nonlinearity (specifically, normalization) in decentralized optimization, thereby motivating broader applications of nonlinear techniques in this setting.

1.3 Notation

We denote by $\mathbb{N}_+$, $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}^d$, respectively, the set of positive natural numbers, real numbers, nonnegative real numbers, and the d-dimensional Euclidean space. We use lowercase normal letters for scalars, lowercase boldface letters for vectors, and uppercase boldface letters for matrices. Further, we denote by $\mathbf{1}_k$ and $\mathbf{0}_k$ the all-ones and all-zeros vectors of size k, respectively, and by $\boldsymbol{I}_k$ the $k \times k$ identity matrix. We let $\|\boldsymbol{x}\|$ denote the ℓ_2 norm of $\boldsymbol{x}$, and $\|\boldsymbol{A}\|_2$ denote the operator norm of $\boldsymbol{A}$. For functions $p(t)$ and $q(t)$ in t, we write $p(t) = O(q(t))$ if $\limsup_{t\to\infty} p(t)/q(t) < \infty$. Finally, we use $\mathbb{E}$ to denote expectation over random quantities.

2 Problem Formulation

We consider a graph with n nodes, where each node holds a local and private function $f_i : \mathbb{R}^d \to \mathbb{R}$, and the nodes collectively minimize the unconstrained global objective $f(\boldsymbol{x}) := (1/n)\sum_{i=1}^n f_i(\boldsymbol{x})$ through peer-to-peer communication. We now present some standard assumptions on the problem.

Assumption 1 (Finite lower bound)**.** *There exists some* $f_* := \inf_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) > -\infty$.

Assumption 2 (L-smoothness)**.** *The local function f_i at each node $i \in [n]$ is differentiable and L-smooth, i.e., $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.*

We next introduce the heavy-tailed noise model. For each node $i \in \mathcal{V}$, at t-th iteration with query $\boldsymbol{x}_i^t$, the stochastic first-order oracle returns the gradient estimator $\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t)$, where $\boldsymbol{\xi}_i^t$ denotes the random sample. Let $\Omega, \emptyset$ denote the universe, empty set, respectively. We use the following natural filtration, i.e., an increasing family of sub-σ-algebras, to denote the past history up to iteration t:

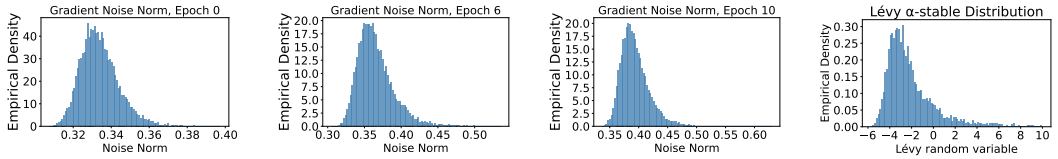
$$\mathcal{F}_{-1} := \{\Omega, \emptyset\}, \quad \mathcal{F}_t := \sigma\big(\{\boldsymbol{\xi}_i^0, \ldots, \boldsymbol{\xi}_i^{t-1} : i \in [n]\}\big), \forall t \geq 0.$$

We then assume this stochastic first-order oracle have the following properties.

Assumption 3 (Heavy-tailed noise)**.** *For any $\mathcal{F}_t$-measurable random vectors $\boldsymbol{x} \in \mathbb{R}^d$, we have the following: $\forall i \in [n], \forall t \geq 0$, (1) $\mathbb{E}[\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i^t) \mid \mathcal{F}_t] = \nabla f_i(\boldsymbol{x})$; (2) There exist $p \in (1, 2]$, some constant $\sigma \geq 0$ such that $\mathbb{E}\big[\|\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x})\|^p \mid \mathcal{F}_t\big] \leq \sigma^p$; (3) The family $\{\boldsymbol{\xi}_i^t : \forall t \geq 0, i \in [n]\}$ of random samples is independent.*

Remark 1 (Heavy-tailed distributions). Assumption 3 covers a broad class of heavy-tailed distributions, including Lévy's α-stable distributions, Student's t-distributions, and Pareto distributions. Note that we do not assume noise symmetry as in Yu et al. (2023), and when $p = 2$, Assumption 3 reduces to the standard bounded variance condition commonly assumed in the literature.

Remark 2 (Empirical evidence). Similar to Zhang et al. (2020); Yang et al. (2022), we investigate the empirical distribution of the gradient noise norm $\|\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\xi}) - \nabla f(\boldsymbol{x})\|$ in a centralized setting by training a GPT model (Radford et al., 2018) with 3M parameters on the Multi30k dataset (Elliott



(a) Epoch 0 (b) Epoch 6 (c) Epoch 10 (d) α-stable distribution

Figure 1: Comparisons of the empirical density of gradient noise norm in different epochs of training a Transformer model with a synthetic Lévy α-stable distribution.

et al., 2016), where $\boldsymbol{g}(\boldsymbol{x})$ denotes the mini-batch stochastic gradient and $\nabla f(\boldsymbol{x})$ denotes the full-batch gradient. We train the model for 12 epochs using SGD and plot the empirical density of the noise norm at the beginning of epochs 0, 6, and 10. As shown in Figure 1, as training progresses, the tail of the empirical gradient noise norm distribution becomes heavier (and longer) and increasingly resembles that of a synthetic α-stable distribution.

For peer-to-peer communication in decentralized settings, we need to specify a mixing matrix $\boldsymbol{W}$ on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Assumption 4 (Weight matrix)**.** *The nonnegative weight matrix $\boldsymbol{W}$, whose (i, r)-th component of $\boldsymbol{W}$, denoted as w_{ir}, is positive if and only if $(i, r) \in \mathcal{E}$ or $i = r$, is primitive and doubly stochastic, i.e., $\mathbf{1}_n \boldsymbol{W} = \mathbf{1}_n$ and $\boldsymbol{W}\mathbf{1}_n = \mathbf{1}_n$.*

Assumption 4 is standard in the decentralized optimization literature (Xin et al., 2020a), and it guarantees that there exists some nonnegative λ, i.e., spectral gap, such that

$$\|\boldsymbol{W} - \mathbf{1}_n \mathbf{1}^\top / n\|_2 := \lambda < 1.$$

The assumed weight matrix $\boldsymbol{W}$ can be constructed on undirected and connected graphs (Olshevsky, 2014), and also on some directed and strongly connected graphs that are weight-balanced (Gharesifard & Cortés, 2012). For instance, the family of directed exponential graphs, is weight-balanced and serves as an important topology configuration in decentralized training (Assran et al., 2019).

3 Algorithm Development: GT−NSGDm

We now describe the proposed Algorithm GT−NSGDm and discuss the intuition of its construction. We use $\boldsymbol{x}_i^t$ to denote the estimate of a stationary point for the global cost function f at node i and t-th iteration, and recall that $\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t)$ denotes the corresponding stochastic gradient returned from local first-order oracle. Motivated by the error-feedback approach in Yu et al. (2023), which serves as a momentum-type of variance reduction after applying a nonlinear operator to handle heavy-tailed noise, we also employ local momentum variance reduction

$$\boldsymbol{v}_i^t = \beta \boldsymbol{v}_i^{t-1} + (1-\beta)\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t), \tag{1}$$

where $\beta \in [0, 1)$ serves as the momentum coefficient. Then, we use gradient tracking (Di Lorenzo & Scutari, 2016) to handle heterogeneous local functions $\{f_i\}_{i=1}^n$. Specifically, we use an estimator $\boldsymbol{y}_i^t$ to track global gradient

$$\boldsymbol{y}_i^t = \sum_{r=1}^n w_{ir}\big(\boldsymbol{y}_r^{t-1} + \boldsymbol{v}_r^t - \boldsymbol{v}_r^{t-1}\big). \tag{2}$$

It is known that gradient tracking helps eliminate the dependence on heterogeneity among local functions $\{f_i\}_{i=1}^n$, such as the requirement of bounded gradient similarity. Furthermore, similar to the approach in Liu & Zhou (2025), which uses normalization to address heavy-tailed noise in *centralized* settings, we avoid applying normalization in the recursive updates of the local gradient estimator $\boldsymbol{v}_i^t$ in (1) and the global gradient tracker $\boldsymbol{y}_i^t$. Instead, normalization is applied only during the update of $\boldsymbol{x}_i^t$, with step size α, and nonnegative mixing weights $\{w_{ir}\}$ where $w_{ir} > 0$ only when $(i, r) \in \mathcal{E}$ or $i = r$,

$$\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\Big(\boldsymbol{x}_r^t - \alpha \frac{\boldsymbol{y}_r^t}{\|\boldsymbol{y}_r^t\|}\Big). \tag{3}$$

We combine the local updates (1)(2)(3) on node $i \in \mathcal{V}$ and call it GT−NSGDm, Gradient Tracking based Normalized Stochastic Gradient Descent with momentum. When taking $\beta = 0$, this simplifies to momentum-free gradient tracking with normalization in step 3. However, our analysis shows that GT−NSGDm performs optimal for some $\beta \in (0, 1)$, making GT−NSGDm a non-trivial and optimal algorithmic design for the considered problem class. We provide a tabular description for GT−NSGDm in Algorithm 1, where all $\{\boldsymbol{x}_i^0\}$ are initialized from the same point $\bar{\boldsymbol{x}}^0$ for simplicity.

Algorithm 1 GT−NSGDm at each node i

Require: $\boldsymbol{x}_i^{-1} = \boldsymbol{x}_i^0 = \overline{\boldsymbol{x}}^0, \boldsymbol{v}_i^{-1} = \boldsymbol{y}_i^{-1} = \mathbf{0}_d, \alpha, \beta, \{w_{ir}\}, T.$

1: **for** $t = 0$ to $T - 1$ **do**
2: Sample $\boldsymbol{\xi}_i^t$; (random sample for stochastic gradient)
3: $\boldsymbol{v}_i^t \leftarrow \beta \boldsymbol{v}_i^{t-1} + (1-\beta)\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t)$; (local gradient estimator)
4: $\boldsymbol{y}_i^t \leftarrow \sum_{r=1}^n w_{ir}(\boldsymbol{y}_r^{t-1} + \boldsymbol{v}_r^t - \boldsymbol{v}_r^{t-1})$; (local gradient tracker)
5: $\boldsymbol{x}_i^{t+1} \leftarrow \sum_{r=1}^n w_{ir}\big(\boldsymbol{x}_r^t - \alpha \frac{\boldsymbol{y}_r^t}{\|\boldsymbol{y}_r^t\|}\big)$; (peer-to-peer communication)
6: **end for**

Remark 3 (Why vanilla gradient normalization fails?). Although vanilla normalization is successfully used in *centralized* settings to robustify SGD against heavy-tailed noise (Hübler et al., 2024), its direct extension to the *decentralized* settings fails. Suppose we run a vanilla decentralized normalized (noiseless) gradient descent, i.e., in parallel $\forall i \in \mathcal{V}$,

$$\boldsymbol{x}_i^{t+1} = \sum_{i=1}^n w_{ir}\big(\boldsymbol{x}_r^t - \alpha \frac{\nabla f_r(\boldsymbol{x}_r^t)}{\|\nabla f_r(\boldsymbol{x}_r^t)\|}\big). \tag{4}$$

Then, the global average $\bar{\boldsymbol{x}}^t$ would update in the negative direction of *the sum of normalized local gradients*: $\bar{\boldsymbol{x}}^{t+1} = \bar{\boldsymbol{x}}^t - \frac{\alpha}{n}\sum_{r=1}^n \frac{\nabla f_r(\boldsymbol{x}_r^t)}{\|\nabla f_r(\boldsymbol{x}_r^t)\|}$. Let for some $t, \forall r \in \mathcal{V}, \boldsymbol{x}_r^t = \boldsymbol{x}_* = \arg\min \sum_{i=1}^n f_i(\boldsymbol{x})$, i.e., all nodes hold the optimal global solution that $\sum_{r=1}^n \nabla f_r(\boldsymbol{x}_*) = \mathbf{0}$. Since $\|\nabla f_r(\boldsymbol{x}_*)\|$ can be different quantities for $r = 1, \ldots, n$, due to function heterogeneity, then $\bar{\boldsymbol{x}}^{t+1}$ will move away from $\boldsymbol{x}_*$. Therefore, vanilla gradient normalization adds some intrinsic errors from heterogeneous local normalizations. By incorporating gradient tracking, we expect that $\boldsymbol{y}_r^t$ would converge to its global average $\bar{\boldsymbol{y}}^t$, and $\bar{\boldsymbol{y}}^t$ would converge to $(1/n)\sum_{i=r}^n \nabla f_i(\boldsymbol{x}_r^t)$. In this way, $\boldsymbol{x}_r^t$ would move along the direction of the *normalized sum of local gradients*, and thus emulating the centralized setting.

In the following claim, we further demonstrate that vanilla gradient normalization can cause the iterates $\boldsymbol{x}_i^t$ to remain arbitrarily far from the optimal solution (see Appendix A for a proof).

Claim 1. *Consider algorithm 4. For any even n, for any $B \geq 1$, there exist $\{f_i\}_{i=1}^n$ satisfying Assumptions 1-2, a gradient oracle satisfying Assumption 3, a mixing matrix satisfying Assumption 4, and an initialization $\boldsymbol{x}_0$, such that the associated parameters f_*, L, σ, $\boldsymbol{W}$, $\boldsymbol{x}_0$, are independent of B. Then, $\forall T \geq 1, \forall \alpha > 0$, it holds that $\frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^n \mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big] \geq B$.*

We next break the limitations of vanilla gradient normalization in Claim 1 by incorporating gradient tracking and momentum variance reduction. This enables the successful use of normalization to suppress heavy-tailed noise while maintaining optimal convergence despite the added nonlinearity.

4 Main Results

We present the main convergence results of GT−NSGDm and discuss their implications. The detailed analyses are deferred to the Appendix B. We first consider the case where the tail index p is known.

Theorem 1. *Let Assumptions 1, 2, 3, 4 hold. Denote $f(\bar{\boldsymbol{x}}^0) - f_* = \Delta_0$, $[\nabla f_1(\bar{\boldsymbol{x}}^0), \ldots, \nabla f_n(\bar{\boldsymbol{x}}^0)]^\top = \nabla F(\mathbf{1}_n \otimes \bar{\boldsymbol{x}}^0)$. Take*

$$\alpha = \min\Big(1, \sqrt{\frac{\Delta_0(1-\beta)(1-\lambda)}{4LT}}, \sqrt{\frac{\Delta_0(1-\lambda)}{3.5LT}}, \sqrt{\frac{(1-\lambda)^2\Delta_0}{2n^{\frac{1}{2}}LT}}\Big), \tag{5}$$

and $1-\beta = 1/T^{\frac{p}{3p-2}}$. *Assume* $\beta \geq 1/10$, *then the sequence generated by* `GT-NSGDm` *satisfies that*

$$\frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big] = O\Big(\frac{\sigma}{n^{1-\frac{1}{p}}T^{\frac{p-1}{3p-2}}} + \frac{1}{T^{\frac{p-1}{3p-2}}}\sqrt{\frac{L\Delta_0}{1-\lambda}} + \frac{\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{T^{\frac{2p-2}{3p-2}}} + \sqrt{\frac{3.5L\Delta_0}{(1-\lambda)T}}$$
$$+\sqrt{\frac{n^{\frac{1}{2}}L\Delta_0}{(1-\lambda)^2T}} + \frac{\sigma n^{\frac{1}{2}}}{(1-\lambda)^{\frac{1}{p}}T^{\frac{p}{3p-2}}} + \frac{\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{(1-\lambda)n^{\frac{1}{2}}T^{\frac{p}{3p-2}}} + \frac{\sigma}{1-\lambda}\frac{n^{\frac{1}{2}}}{T^{\frac{2p-1}{3p-2}}} + \frac{\Delta_0}{T}\Big).$$

Remark 4 (Order-optimal rate). Theorem 1 establishes a non-asymptotic upper bound on the mean ℓ_2 norm stationary gap of `GT-NSGDm` over any finite time horizon T. The $O(\cdot)$ here only absorbs universal constants and preserves all problem parameters. It achieves the *optimal* $O\big(1/T^{\frac{p-1}{3p-2}}\big)$ convergence rate in terms of T as it matches the lower bound proved in Zhang et al. (2020). This optimal guarantee is achieved in decentralized settings *for the first time.*

Remark 5 (Speedup in n). We discuss the asymptotic speedup in number of nodes n. For sufficiently large T (or sufficiently small target optimality gap), the upper bound in Theorem 1 is dominated by the leading terms $(1/T^{\frac{p-1}{3p-2}})\big(\sigma/n^{1-1/p} + \sqrt{L\Delta_0/(1-\lambda)}\big)$. In the high-noise regime $\sigma \gg n^{1-1/p}\sqrt{L\Delta_0/(1-\lambda)}$, the upper bound has a speedup factor $n^{1-1/p}$. In practice, the noise scale (measured by σ) in training attention models or in other high-dimensional problems can be very large, and the speedup in n contributes as a noise reduction.

When the tail index p is unknown in advance, we establish the following convergence rate.

Theorem 2. *Let Assumptions 1, 2, 3, 4 hold and take* α *as in* (5). *Take* $1-\beta = 1/\sqrt{T}$ *and assume* $\beta \geq 1/10$. *Then* `GT-NSGDm` *guarantees that*

$$\frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big] \leq O\Big(\frac{\sigma}{n^{1-\frac{1}{p}}T^{\frac{p-1}{2p}}} + \frac{1}{T^{\frac{1}{4}}}\sqrt{\frac{L\Delta_0}{1-\lambda}} + \frac{\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{\sqrt{T}} + \sqrt{\frac{3.5L\Delta_0}{(1-\lambda)T}}$$
$$+\frac{\sigma n^{\frac{1}{2}}}{(1-\lambda)^{\frac{1}{p}}\sqrt{T}} + \frac{\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{(1-\lambda)n^{\frac{1}{2}}\sqrt{T}} + \sqrt{\frac{n^{\frac{1}{2}}L\Delta_0}{(1-\lambda)^2T}} + \frac{\sigma n^{\frac{1}{2}}}{(1-\lambda)T^{\frac{2p-1}{2p}}} + \frac{\Delta_0}{T}\Big).$$

Theorem 2 establishes an upper bound of $O(1/T^{\frac{p-1}{2p}})$ when the tail index p is unknown, matching the best-known rate in the *centralized* setting where algorithm parameters do not rely on p (Liu & Zhou, 2025). While the convergence rate in Yu et al. (2023) is also independent of the knowledge of p, it is only for strongly convex functions and its exact rate exponent remains unspecified.

Remark 6 (Speedup in n and topology independent rate). Consider $p \in (1,2)$, i.e., the heavy-tailed case with unbounded variance this paper focuses on. When T is sufficiently large (as required to achieve sufficiently small target optimality gap), the upper bound in Theorem 2 is dominated by $\frac{\sigma}{n^{1-1/p}} \cdot \frac{1}{T^{(p-1)/2p}}$. Significantly, this upper bound is *independent* of network topology (λ) and exhibits a speedup factor $n^{1-1/p}$ in all regimes.

5 Experiments

We assess the performance of `GT-NSGDm` through numerical experiments. We first conduct studies on synthetic datasets that mimic language modeling under controlled heavy-tailed noise injection, following Lee et al. (2025). We also present experiments on decentralized training of a decoder-only Transformer (GPT) model with 3M parameters on the Multi30k dataset.

Baselines. We compare `GT-NSGDm` with four decentralized baselines: `DSGD`(Nedic & Ozdaglar, 2009), `GT-DSGD` (Xin et al., 2020b), `DSGD-Clip` (Sun & Chen, 2024), and `SClip-EF-Network` (Yu et al., 2023). `DSGD` and `GT-DSGD` handle regular stochastic noise with bounded variance. `DSGD-Clip` converges for strongly convex functions under bounded domains or gradients (Sun & Chen, 2024). `SClip-EF-Network` achieves convergence under symmetric noise with bounded $\mathbb{E}\big[\|\boldsymbol{\xi}_i^t\|^p \mid \mathcal{F}_{t-1}\big]$ for $p=1$. All methods are initialized identically and tuned via grid search. Detailed baseline descriptions appear in Table 2 (Appendix C.1).

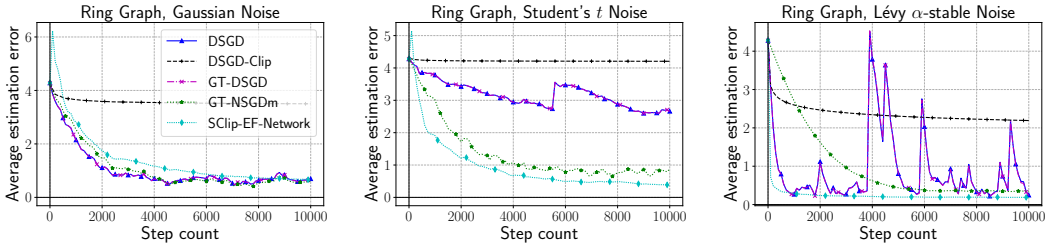
Graph topology. We consider three graph topologies: undirected ring, directed exponential, and complete graphs (see Lian et al. (2017); Nedić et al. (2018); Assran et al. (2019)). Weight matrices

use Metropolis weights (Xiao et al., 2005). For synthetic experiments, we set the number of nodes to $n = 20$, we obtain $\lambda = 0.904$, 0.714, and 0 for the ring, exponential, and complete graphs, respectively. For Transformer training with $n = 8$, we have corresponding $\lambda = 0.804$, 0.6, and 0.

5.1 Robust linear regression on synthetic tokenized data

We use this synthetic experiment to test our convergence rates under controlled heavy-tailed noise. We consider nonconvex regularized linear regression on synthetic data mimicking language tokens. In language modeling, token frequencies exhibit heavy-tailed distributions: few tokens appear frequently, while most are rare but contextually important. We construct the following synthetic dataset $\boldsymbol{X}$ of 1k samples of dimension $d = 20$. The first two features simulate frequent tokens, sampled from Bernoulli distributions Bern(0.9) and Bern(0.5), respectively. The remaining 8 features represent rare tokens, each sampled from Bern(0.1). The optimal weight $\boldsymbol{w}_*$ is Gaussian-sampled, with labels $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}_*$. The synthetic dataset $(\boldsymbol{X}, \boldsymbol{y})$ is evenly distributed over $n = 20$ nodes, where each node i holds a sub-dataset $(\boldsymbol{X}_i, \boldsymbol{y}_i)$, estimate $\boldsymbol{w}_i$, and a linear regression model with nonconvex robust Tukey's biweight loss function (Beaton & Tukey, 1974) to estimate $\boldsymbol{w}_*$. We inject three different *zero-mean* noises, Gaussian noise ($\mathcal{N}(\boldsymbol{0}, 3\boldsymbol{I}_d)$), Student's t noise (degrees of freedom 1.5, scale 1.0), and Lévy α-stable noise (stability parameter 1.5, skewness parameter 0.5, scale 1.0, nonsymmetric, multiplied by 0.1) into exact gradient, using corrupted stochastic gradients for updates. See Appendix C.2 for additional details.

In Figure 2, we evaluate `GT-NSGDm` against baselines on ring graphs under various gradient noise. `DSGD` and `GT-DSGD` converge under Gaussian noise but become unstable under heavy-tailed noise. `DSGD-Clip` remains stable but fails to reach optimum. Both `GT-NSGDm` and `SClip-EF-Network` exhibit robust convergence and near-optimal performance across all scenarios, consistent with their theoretical guarantees under heavy-tailed noise.



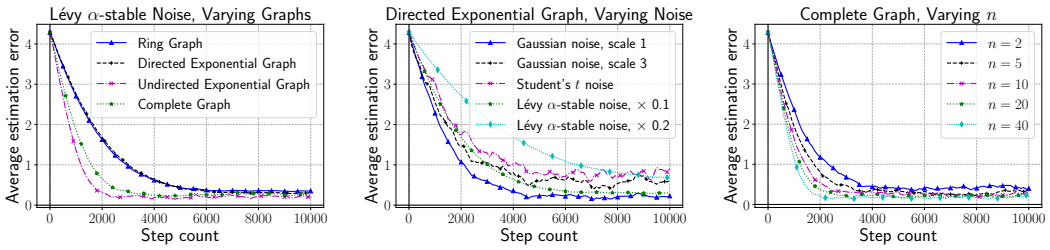
(a) Gaussian Noise (b) Student's t noise (c) Lévy α-stable noise

Figure 2: Comparison of performance on a ring graph under various types of injected stochastic gradient noise, measured by the average estimation error $(1/n)\sum_{i=1}^{n}\|\boldsymbol{w}_i^t - \boldsymbol{w}_*\|$ over step count t.

In Figure 3, we test `GT-NSGDm`'s dependence on connectivity (λ), noise level (σ), and the number of nodes (n), varying each while fixing others. In Figure 3(a), we inject Lévy α-stable noise and test the performance of `GT-NSGDm` on ring, directed exponential, undirected exponential, and complete graphs with $\lambda = 0.904, 0.714, 0.6, 0$, respectively. `GT-NSGDm` achieves comparable final errors under weak connectivity (i.e., large λ) versus complete graphs, showing favorable dependence on network connectivity under heavy-tailed noise. In Figure 3(b), we evaluate `GT-NSGDm`'s performance under different noise levels on a directed exponential graph. Under Gaussian noise with scale 1 (unit variance), `GT-NSGDm` reaches the best optimality; the final error increases as σ grows, as observed under both Gaussian and Lévy α-stable noise. In Figure 3(c), we inject Lévy α-stable noise on complete graphs ($\lambda = 0$ for all n) with varying number of nodes. As n increases from 2 to 40, convergence speed improves with final errors $[0.4, 0.35, 0.29, 0.20, 0.21]$, demonstrating speedup over certain n ranges, supporting theoretical discussions in Remarks 5 and 6.

5.2 Decentralized training of Transformers

We evaluate `GT-NSGDm`'s empirical performance on language modeling using a 3M-parameter GPT model (Radford et al., 2018) for auto-regressive modeling on Multi30k (29k sentences, 4.4M tokens). We assess performance using validation log-perplexity. On 8-node graphs with three topolo-



(a) Dependence on λ (b) Dependence on σ (c) Dependence on n

Figure 3: Empirical studies on GT−NSGDm's dependence on problem parameters λ, σ, n.

gies, we distribute training data evenly and initialize identical GPT models per node. We introduce three additional baselines: DSGD−GClip (DSGD with constant step size and ℓ_2 gradient clipping level), DSGD−CClip (DSGD with constant step size and component-wise gradient clipping level), QG−DSGDm (Lin et al., 2021) and GT−Adam (Carnevale et al., 2022) (all without theoretical guarantees under heavy-tailed noise; see Table 2 in Appendix C.1). We run all methods for 12 epochs with batch size 64. See Appendix C.3 for model and hyperparameter details.

Table 1 presents average validation loss and standard deviation over five independent runs for each algorithm across three topologies. Results show that GT−NSGDm nearly matches the best baseline DSGD−GClip (which lacks theoretical guarantees under heavy-tailed gradient noise) while significantly outperforming the other two theoretically-guaranteed baselines across all topologies. We note that this decentralized training experiment is simulated to demonstrate algorithm effectiveness and has practical limitations.

Table 1: Topologies ring, directed exponential (Exp.), and complete (Comp.) graphs. Algorithms are grouped by theoretical (Theo.) guarantees under heavy-tailed noise: with (w/) or without (w/o).

Algorithms	Theo.	Ring	Exp.	Comp.
DSGD	w/o	$5.633_{\pm 0.008}$	$5.632_{\pm 0.007}$	$5.635_{\pm 0.007}$
DSGD−GClip	w/o	$\mathbf{0.253}_{\pm 0.007}$	$\mathbf{0.249}_{\pm 0.010}$	$\mathbf{0.267}_{\pm 0.010}$
DSGD−CClip	w/o	$2.725_{\pm 3.179}$	$5.058_{\pm 2.388}$	$8.225_{\pm 1.695}$
GT−DSGD	w/o	$5.362_{\pm 0.002}$	$5.632_{\pm 0.002}$	$5.631_{\pm 0.002}$
GT−Adam	w/o	$0.520_{\pm 0.038}$	$0.587_{\pm 0.096}$	$0.524_{\pm 0.045}$
QG−DSGDm	w/o	$0.394_{\pm 0.007}$	$0.388_{\pm 0.013}$	$0.353_{\pm 0.011}$
SClip−EF−Network	w/	$5.653_{\pm 0.012}$	$5.632_{\pm 0.003}$	$5.636_{\pm 0.004}$
DSGD−Clip	w/	$5.633_{\pm 0.004}$	$5.659_{\pm 0.013}$	$5.661_{\pm 0.006}$
GT−NSGDm	w/	$\mathbf{0.258}_{\pm 0.007}$	$\mathbf{0.261}_{\pm 0.007}$	$\mathbf{0.282}_{\pm 0.009}$

6 Conclusion and Future Work

In this paper, we have proposed GT−NSGDm for solving decentralized nonconvex smooth optimization to address heavy-tailed noise. The key idea is to leverage normalization, together with momentum variance reduction, to combat heavy-tailed noise, and use gradient tracking to handle cross-node heterogeneity and the nonlinearity brought by normalization. Theoretical analyses establish that GT−NSGDm attains optimal convergence rate when the tail index p is known, and a rate that matches the best centralized one when p is unknown. Extensive experiments on nonconvex linear regression and decentralized Transformer training show that GT−NSGDm is robust and efficient under heavy-tailed noise across various topologies, and achieves a speedup in n. Future directions include extending the current analysis to other nonlinearities, such as sign and clipping (Zhang et al., 2020), and generalizing GT−NSGDm to handle objective functions under relaxed smoothness conditions (Liu & Zhou, 2025).

REPRODUCIBLE STATEMENT

We provide detailed proofs for Claim 1 in Appendix A and for our main theoretical results, Theorems 1 and 2, in Appendix B. In Appendix 5, we provide detailed hardware configurations, algorithm descriptions, and hyperparameter settings for our numerical experiments.

REFERENCES

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand transformer optimization). In *The Twelfth International Conference on Learning Representations*, 2024.

Aleksandar Armacki, Shuhua Yu, Dragana Bajovic, Dusan Jakovetic, and Soummya Kar. Large deviations and improved mean-squared error rates of nonlinear sgd: Heavy-tailed noise and power of symmetry. *arXiv preprint arXiv:2410.15637*, 2024.

Aleksandar Armacki, Shuhua Yu, Pranay Sharma, Gauri Joshi, Dragana Bajovic, Dusan Jakovetic, and Soummya Kar. High-probability convergence bounds for online nonlinear stochastic gradient descent under heavy-tailed noise. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pp. 344–353. PMLR, 2019.

Melih Barsbey, Milad Sefidgaran, Murat A Erdogdu, Gael Richard, and Umut Simsekli. Heavy tails in sgd and compressibility of overparametrized neural networks. *Advances in neural information processing systems*, 34:29364–29378, 2021.

Barak Battash, Lior Wolf, and Ofir Lindenbaum. Revisiting the noise model of stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics*, pp. 4780–4788. PMLR, 2024.

Albert E Beaton and John W Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Guido Carnevale, Francesco Farina, Ivano Notarnicola, and Giuseppe Notarstefano. Gtadam: Gradient tracking with adaptive momentum for distributed online optimization. *IEEE Transactions on Control of Network Systems*, 10(3):1436–1448, 2022.

Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. *Advances in neural information processing systems*, 34: 20461–20475, 2021.

Savelii Chezhegov, Yaroslav Klyukin, Andrei Semenov, Aleksandr Beznosikov, Alexander Gasnikov, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. Gradient clipping improves adagrad when the noise is heavy-tailed. *arXiv preprint arXiv:2406.04443*, 2024.

Enea Monzio Compagnoni, Rustem Islamov, Frank Norbert Proske, and Aurelien Lucchi. Unbiased and sign compression in distributed learning: Comparing noise resilience via SDEs. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025a.

Enea Monzio Compagnoni, Tianlin Liu, Rustem Islamov, Frank Norbert Proske, Antonio Orvieto, and Aurelien Lucchi. Adaptive methods through the lens of SDEs: Theoretical insights on the role of noise. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*, 2016.

Bahman Gharesifard and Jorge Cortés. Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, 18(6):539–557, 2012.

Eduard Gorbunov, Marina Danilova, and Alexander Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *Advances in Neural Information Processing Systems*, 33:15042–15053, 2020.

Eduard Gorbunov, Abdurakhmon Sadiev, Marina Danilova, Samuel Horváth, Gauthier Gidel, Pavel Dvurechensky, Alexander Gasnikov, and Peter Richtárik. High-probability convergence for composite and distributed stochastic minimization and variational inequalities with heavy-tailed noise, 2024. URL https://openreview.net/forum?id=qOFLn0pMoe.

Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. The heavy-tail phenomenon in sgd. In *International Conference on Machine Learning*, pp. 3964–3975. PMLR, 2021.

Florian Hübler, Ilyas Fatkhullin, and Niao He. From gradient clipping to normalization for heavy tailed sgd. *arXiv preprint arXiv:2410.13849*, 2024.

Dusan Jakovetić, Dragana Bajović, Anit Kumar Sahu, Soummya Kar, Nemanja Milosević, and Dusan Stamenković. Nonlinear gradient mappings and stochastic optimization: A general framework with applications to heavy-tail noise. *SIAM Journal on Optimization*, 33(2):394–423, 2023.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393. PMLR, 2020.

Frederik Kunstner, Alan Milligan, Robin Yadav, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *Advances in Neural Information Processing Systems*, 37:30106–30148, 2024.

Su Hyeong Lee, Manzil Zaheer, and Tian Li. Efficient distributed optimization under heavy-tailed noise. *arXiv preprint arXiv:2502.04164*, 2025.

Boyue Li and Yuejie Chi. Convergence and privacy of decentralized nonconvex optimization with gradient clipping and communication compression. *IEEE Journal of Selected Topics in Signal Processing*, 2025.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017.

Tao Lin, Sai Praneeth Karimireddy, Sebastian Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *International Conference on Machine Learning*, pp. 6654–6665. PMLR, 2021.

Zijian Liu and Zhengyuan Zhou. Nonconvex stochastic optimization under heavy-tailed noises: Optimal convergence without gradient clipping. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zijian Liu, Jiawei Zhang, and Zhengyuan Zhou. Breaking the lower bound with (little) structure: Acceleration in non-convex stochastic optimization with heavy-tailed noise. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 2266–2290. PMLR, 2023.

Jayakrishnan Nair, Adam Wierman, and Bert Zwart. *The fundamentals of heavy tails: Properties, emergence, and estimation*, volume 53. Cambridge University Press, 2022.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

Ta Duy Nguyen, Thien H Nguyen, Alina Ene, and Huy Nguyen. Improved convergence in high probability of clipped gradient methods with heavy tailed noise. *Advances in Neural Information Processing Systems*, 36:24191–24222, 2023.

Alex Olshevsky. Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. *arXiv preprint arXiv:1411.4186*, 2014.

Stefano Peluchetti, Stefano Favaro, and Sandra Fortini. Stable behaviour of infinitely wide deep neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 1137–1146. PMLR, 2020.

Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187(1):409–457, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Abdurakhmon Sadiev, Marina Danilova, Eduard Gorbunov, Samuel Horváth, Gauthier Gidel, Pavel Dvurechensky, Alexander Gasnikov, and Peter Richtárik. High-probability bounds for stochastic optimization and variational inequalities: the case of unbounded variance. In *International Conference on Machine Learning*, pp. 29563–29648. PMLR, 2023.

Egor Shulgin, Sarit Khirirat, and Peter Richtárik. Smoothed normalization for efficient distributed private optimization. *arXiv preprint arXiv:2502.13482*, 2025.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

Chao Sun and Bo Chen. Distributed stochastic strongly convex optimization under heavy-tailed noises. In *2024 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 150–155. IEEE, 2024.

Tao Sun, Xinwang Liu, and Kun Yuan. Gradient normalization provably benefits nonconvex sgd under heavy-tailed noise. *arXiv preprint arXiv:2410.16561*, 2024.

S Sundhar Ram, Angelia Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147:516–545, 2010.

John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9): 803–812, 1986.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22(213):1–50, 2021.

Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pp. 63–70. IEEE, 2005.

Ran Xin, Usman A Khan, and Soummya Kar. Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68:6255–6271, 2020a.

Ran Xin, Shi Pu, Angelia Nedić, and Usman A Khan. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11):1869–1889, 2020b.

Haibo Yang, Peiwen Qiu, and Jia Liu. Taming fat-tailed ("heavier-tailed" with potentially infinite variance) noise in federated learning. *Advances in Neural Information Processing Systems*, 35: 17017–17029, 2022.

Shuhua Yu and Soummya Kar. Secure distributed optimization under gradient attacks. *IEEE Transactions on Signal Processing*, 2023.

Shuhua Yu, Dusan Jakovetic, and Soummya Kar. Smoothed gradient clipping and error feedback for decentralized optimization under symmetric heavy-tailed noise. *arXiv preprint arXiv:2310.16920*, 2023.

Kun Yuan, Bicheng Ying, Jiageng Liu, and Ali H Sayed. Variance-reduced stochastic learning by networked agents under random reshuffling. *IEEE Transactions on Signal Processing*, 67(2): 351–366, 2018.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

APPENDIX

A PROOF OF CLAIM 1

Proof. Consider n scalar functions that for each $i \in \mathcal{V}$, $f_i(x) = (1/2)(x - a_i)^2$ for some a_i, and complete graph with $\boldsymbol{W} = (1/n)\mathbf{1}_n\mathbf{1}_n^\top$. Let $a_i = a, \forall i = 1, \ldots, n/2$, and $a_i = b, \forall i = n/2+1, \ldots, n$, and $b-a > 2B+1$. Let $x_i^0 = a+0.5, \forall i \in \mathcal{V}$. Then, $\forall i \in \mathcal{V}$, vanilla normalization reduces to

$$\begin{aligned} x_i^1 &= \frac{1}{n}\sum_{r=1}^{n}\Big(x_r^0 - \alpha\text{sign}(x_r^0 - a_r)\Big) \\ &= x_r^0 - \frac{\alpha}{n}\sum_{r=1}^{n}\text{sign}(x_r^0 - a_r) \\ &= x_r^0 - \frac{\alpha}{n}\sum_{r=1}^{n/2}\text{sign}(0.5) - \frac{\alpha}{n}\sum_{r=n/2+1}^{n}\text{sign}(0.5 - (b-a)) \\ &= x_r^0. \end{aligned}$$

Therefore, $x_r^t = a + 0.5, \forall r \in \mathcal{V}, \forall t \geq 0$. Since the optimal solution to the original problem is $\frac{a+b}{2}$, the optimality gap is $\frac{b-a}{2} - 0.5 \geq B$. □

Remark 7. Note that the proof above can be further extended to the case where the gradient oracle admits almost surely bounded gradient noise. We can use the noise bound to adapt the choices of a, b, ε such that all signs still get canceled. Similar examples have been used to show divergence results in Shulgin et al. (2025).

B PROOFS OF THEOREMS

B.1 PRELIMINARIES

We define some stacked long vectors,

$$\begin{aligned}
F(\boldsymbol{x}^t) &:= [f_1(\boldsymbol{x}_1^t), \ldots, f_n(\boldsymbol{x}_n^t)]^\top, \\
\nabla F(\boldsymbol{x}^t) &:= [\nabla f_1(\boldsymbol{x}_1^t)^\top, \ldots, \nabla f_n(\boldsymbol{x}_n^t)^\top]^\top, \\
\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) &:= [\boldsymbol{g}_1(\boldsymbol{x}_1^t, \boldsymbol{\xi}_1^t)^\top, \ldots, \boldsymbol{g}_n(\boldsymbol{x}_n^t, \boldsymbol{\xi}_n^t)^\top]^\top \\
\boldsymbol{v}^t &:= [(\boldsymbol{v}_1^t)^\top, \ldots, (\boldsymbol{v}_n^t)^\top]^\top, \\
\mathcal{N}(\boldsymbol{y}^t) &:= \Big[\frac{(\boldsymbol{y}_1^t)^\top}{\|\boldsymbol{y}_1^t\|}, \ldots, \frac{(\boldsymbol{y}_n^t)^\top}{\|\boldsymbol{y}_n^t\|}\Big]^\top, \\
\boldsymbol{x}^t &:= [(\boldsymbol{x}_1^t)^\top, \ldots, (\boldsymbol{x}_n^t)^\top]^\top.
\end{aligned}$$

Then, Algorithm 1 can be rewritten in the compact long-vector form:

$$\boldsymbol{v}^t = \beta \boldsymbol{v}^{t-1} + (1-\beta)\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t); \tag{6}$$

$$\boldsymbol{y}^t = \big(\boldsymbol{W} \otimes \boldsymbol{I}_d\big)(\boldsymbol{y}^{t-1} + \boldsymbol{v}^t - \boldsymbol{v}^{t-1}), \tag{7}$$

$$\boldsymbol{x}^{t+1} = \big(\boldsymbol{W} \otimes \boldsymbol{I}_d\big)(\boldsymbol{x}^t - \alpha \mathcal{N}(\boldsymbol{y}^t)). \tag{8}$$

We define the following averages over network:

$$\bar{\boldsymbol{v}}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{v}_i^t, \quad \bar{\boldsymbol{y}}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^t, \quad \tilde{\boldsymbol{y}}^t = \frac{1}{n}\sum_{i=1}^n \frac{\boldsymbol{y}_i^t}{\|\boldsymbol{y}_i^t\|}, \quad \bar{\boldsymbol{x}}^t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^t, \quad \overline{\nabla} F(\boldsymbol{x}^t) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\boldsymbol{x}_i^t). \tag{9}$$

From the doubly stochasticity of $\boldsymbol{W}$, the global average updates as

$$\bar{\boldsymbol{x}}^{t+1} = \bar{\boldsymbol{x}}^t - \frac{\alpha}{n}\sum_{r=1}^n \frac{\boldsymbol{y}_r^t}{\|\boldsymbol{y}_r^t\|} = \bar{\boldsymbol{x}}^t - \alpha \tilde{\boldsymbol{y}}^t. \tag{10}$$

B.2 INTERMEDIATE LEMMAS

We first present some standard useful relations to be used in our analyses.

Lemma 1. *The following relations hold:*

1. $\bar{\boldsymbol{y}}^t = \bar{\boldsymbol{v}}^t$;
2. $\boldsymbol{W} - \mathbf{1}_n\mathbf{1}_n^\top/n = (\boldsymbol{W} - \mathbf{1}_n\mathbf{1}_n^\top/n)(\boldsymbol{I}_n - \mathbf{1}_n\mathbf{1}_n^\top/n) = (\boldsymbol{I}_n - \mathbf{1}_n\mathbf{1}_n^\top/n)(\boldsymbol{W} - \mathbf{1}_n\mathbf{1}_n^\top/n)$;
3. $\boldsymbol{W}^k - \mathbf{1}_n\mathbf{1}_n^\top/n = (\boldsymbol{W} - \mathbf{1}_n\mathbf{1}_n^\top/n)^k, \forall k \in \mathbb{N}_+$;
4. $(1/\sqrt{n})\sum_{i=1}^n \|\boldsymbol{a}_i\| \le \|\boldsymbol{a}\| \le \sum_{i=1}^n \|\boldsymbol{a}_i\|$, $\forall \boldsymbol{a} = [\boldsymbol{a}_1^\top, \ldots, \boldsymbol{a}_n^\top]^\top \in \mathbb{R}^{nd}$,
5. $\sum_{i=1}^m a_i^p \le \big(\sum_{i=1}^m a_i\big)^p \le m^{p-1}\sum_{i=1}^m a_i^p, \forall m \in \mathbb{N}_+, \forall a_i \in \mathbb{R}_+$.

We then present a standard decent lemma for L-smooth functions .

Lemma 2 (Decent lemma for L-smooth functions)**.** *Let Assumption 2 hold. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, there holds*

$$f(\boldsymbol{y}) \le f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

We next present the main descent lemma on the network average.

Lemma 3 (Decent lemma for network average)**.** *Let Assumption 2 hold. Let $\boldsymbol{\epsilon}^t = \bar{\boldsymbol{y}}^t - \nabla f(\bar{\boldsymbol{x}}^t)$. We have*

$$\sum_{t=0}^{T-1} \alpha \|\nabla f(\bar{\boldsymbol{x}}^t)\| \le f(\bar{\boldsymbol{x}}^0) - f_* + \sum_{t=0}^{T-1} 2\alpha\|\boldsymbol{\epsilon}^t\| + \sum_{t=0}^{T-1}\frac{\alpha}{n}\sum_{i=1}^n \|\bar{\boldsymbol{y}}^t - \boldsymbol{y}_i^t\| + \sum_{t=0}^{T-1}\frac{L}{2}\alpha^2.$$

Proof. Since $\|\bar{\boldsymbol{x}}^{t+1} - \bar{\boldsymbol{x}}^t\| = \alpha\|\tilde{\boldsymbol{y}}^t\| = \alpha$, applying Lemma 2 on $\bar{\boldsymbol{x}}^{t+1}, \bar{\boldsymbol{x}}^t$ gives that

$$\begin{aligned} f(\bar{\boldsymbol{x}}^{t+1}) &\le f(\bar{\boldsymbol{x}}^t) + \nabla f(\bar{\boldsymbol{x}}^t)^\top(\bar{\boldsymbol{x}}^{t+1} - \bar{\boldsymbol{x}}^t) + \frac{L}{2}\|\bar{\boldsymbol{x}}^{t+1} - \bar{\boldsymbol{x}}^t\|^2 \\ &\overset{(i)}{\le} f(\bar{\boldsymbol{x}}^t) - \alpha(\bar{\boldsymbol{y}}^t - \boldsymbol{\epsilon}^t)^\top\tilde{\boldsymbol{y}}^t + \frac{L}{2}\alpha^2 \\ &\overset{(ii)}{\le} f(\bar{\boldsymbol{x}}^t) - \alpha(\bar{\boldsymbol{y}}^t)^\top\tilde{\boldsymbol{y}}^t + \alpha\|\boldsymbol{\epsilon}^t\| + \frac{L}{2}\alpha^2, \end{aligned} \tag{11}$$

where we used the definitions (9)(10) in (i), and used Cauchy-Schwartz inequality followed by $\|\tilde{\boldsymbol{y}}^t\| \le 1$ in (ii). Next,

$$\begin{aligned} -(\bar{\boldsymbol{y}}^t)^\top\tilde{\boldsymbol{y}}^t &= -(\bar{\boldsymbol{y}}^t)^\top\Big[\frac{\bar{\boldsymbol{y}}^t}{\|\bar{\boldsymbol{y}}^t\|} + \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^t\big(\frac{1}{\|\boldsymbol{y}_i^t\|} - \frac{1}{\|\bar{\boldsymbol{y}}^t\|}\big)\Big] \\ &\le -\|\bar{\boldsymbol{y}}^t\| + \|\frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i^t\big(\frac{\|\bar{\boldsymbol{y}}^t\|}{\|\boldsymbol{y}_i^t\|} - 1\big)\| \\ &\overset{(i)}{\le} -\|\nabla f(\bar{\boldsymbol{x}}^t)\| + \|\boldsymbol{\epsilon}^t\| + \frac{1}{n}\sum_{i=1}^n \big|\|\bar{\boldsymbol{y}}^t\| - \|\boldsymbol{y}_i^t\|\big| \\ &\overset{(ii)}{\le} -\|\nabla f(\bar{\boldsymbol{x}}^t)\| + \|\boldsymbol{\epsilon}^t\| + \frac{1}{n}\sum_{i=1}^n \|\bar{\boldsymbol{y}}^t - \boldsymbol{y}_i^t\|, \end{aligned} \tag{12}$$

where we used $\|\bar{\boldsymbol{y}}^t\| = \|\nabla f(\bar{\boldsymbol{x}}^t) + \boldsymbol{\epsilon}^t\| \ge \|\nabla f(\bar{\boldsymbol{x}}^t)\| - \|\boldsymbol{\epsilon}^t\|$, and Cauchy-Schwartz inequality in (i), and $|\|\boldsymbol{a}\| - \|\boldsymbol{b}\|| \le \|\boldsymbol{a} - \boldsymbol{b}\|$ for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ in (ii). Plugging in (12) into (11), and summing over $t = 0, \dots, T-1$, we have

$$f(\bar{\boldsymbol{x}}^T) \le f(\bar{\boldsymbol{x}}^0) - \sum_{t=0}^{T-1}\alpha\|\nabla f(\bar{\boldsymbol{x}}^t)\| + \sum_{t=0}^{T-1} 2\alpha\|\boldsymbol{\epsilon}^t\| + \sum_{t=0}^{T-1}\frac{\alpha}{n}\sum_{i=1}^n \|\bar{\boldsymbol{y}}^t - \boldsymbol{y}_i^t\| + \sum_{t=0}^{T-1}\frac{L}{2}\alpha^2.$$

Using $f(\bar{\boldsymbol{x}}^T) \ge f_*$ and rearranging terms above give the desired result. □

With Lemma 3, it remains to bound the gradient estimation error $\|\boldsymbol{\epsilon}^t\|$ and the consensus error $\boldsymbol{y}_i^t - \bar{\boldsymbol{y}}^t$. Let us decompose the gradient estimation error as follows:

$$\boldsymbol{\epsilon}^t = \bar{\boldsymbol{y}}^t - \nabla f(\bar{\boldsymbol{x}}^t) = \bar{\boldsymbol{v}}^t - \nabla f(\bar{\boldsymbol{x}}^t) = \underbrace{\bar{\boldsymbol{v}}^t - \overline{\nabla} F(\boldsymbol{x}^t)}_{:=\boldsymbol{\epsilon}_1^t \in \mathbb{R}^d} + \underbrace{\overline{\nabla} F(\boldsymbol{x}^t) - \nabla f(\bar{\boldsymbol{x}}^t)}_{:=\boldsymbol{\epsilon}_2^t \in \mathbb{R}^d}. \tag{13}$$

It is clear that $\boldsymbol{\epsilon}_1^t$ is the gradient estimation error, and $\boldsymbol{\epsilon}_2^t$, exploiting the smoothness property in 2, can be bounded by the consensus error $\boldsymbol{x}_i^t - \bar{\boldsymbol{x}}^t$. Since the consensus error is also used in bounding $\boldsymbol{\epsilon}_1^t$, we need to first bound the consensus errors $\boldsymbol{x}_i^t - \bar{\boldsymbol{x}}^t$ and $\boldsymbol{y}_i^t - \bar{\boldsymbol{y}}^t$.

Lemma 4 (Consensus errors of $\{\boldsymbol{x}_i^t\}$)**.** *We have for all $t = 0, \dots, T$,*

$$\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{x}_i^t - \bar{\boldsymbol{x}}^t\| \le \frac{\alpha\lambda}{1-\lambda}. \tag{14}$$

Proof. Using the relation 4 in Lemma 1 we have

$$\frac{1}{n}\sum_{i=1}^n \|\boldsymbol{x}_i^t - \bar{\boldsymbol{x}}^t\| \le \frac{1}{\sqrt{n}}\|\boldsymbol{x}^t - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^t\|. \tag{15}$$

From the compact form update in (8), we have

$$\boldsymbol{x}^t = \big(\boldsymbol{W} \otimes \boldsymbol{I}_d\big)\boldsymbol{x}^0 - \alpha\sum_{k=0}^{t-1}(\boldsymbol{W} \otimes \boldsymbol{I}_d)^{t-k}\mathcal{N}(\boldsymbol{y}^k).$$

It follows that

$$
\begin{aligned}
&\|\boldsymbol{x}^t - \mathbf{1}_n \otimes \bar{\boldsymbol{x}}^t\| \\
&= \|\big(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\boldsymbol{x}^t\| \\
&\overset{(i)}{=} \alpha\|\sum_{k=0}^{t-1}\big(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\big(\boldsymbol{W}\otimes \boldsymbol{I}_d\big)^{t-k}\mathcal{N}(\boldsymbol{y}^k)\| \\
&\leq \alpha\|\sum_{k=0}^{t-1}\|\boldsymbol{W}^{t-k} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\|_2\|\mathcal{N}(\boldsymbol{y}^k)\| \\
&\overset{(ii)}{\leq} \alpha\|\sum_{k=0}^{t-1}\|\boldsymbol{W} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\|_2^{t-k}\|\mathcal{N}(\boldsymbol{y}^k)\| \\
&\leq \alpha\sqrt{n}\sum_{k=0}^{t-1}\lambda^{t-k} \qquad (16) \\
&\overset{(iii)}{\leq} \frac{\alpha\sqrt{n}\lambda}{1-\lambda}. \qquad (17)
\end{aligned}
$$

where we used the double stochasticity of $\boldsymbol{W}$ and $\boldsymbol{x}_i^0 = \bar{\boldsymbol{x}}^0, \forall i \in [n]$ in (i), the relation 3 in Lemma 1 in (ii), and Assumption 4 in (iii). Substituting (17) into (15) gives the desired bound in (14). □

Before proceeding to bound consensus errors for $\{\boldsymbol{y}_i^t\}$, we present the following bound on vector-valued martingale difference sequence from Liu & Zhou (2025).

Lemma 5. *Given a sequence of random vectors $\boldsymbol{d}_t \in \mathbb{R}^d$, $\forall t$ such that $\mathbb{E}[\boldsymbol{d}_t \mid \mathcal{F}_{t-1}] = \mathbf{0}$ where $\mathcal{F}_t = \sigma(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_t)$ is the natural filtration, then for any $p \in [1, 2]$, there is*

$$
\mathbb{E}\Big[\|\sum_{t=1}^{T}\boldsymbol{d}_t\|\Big] \leq 2\sqrt{2}\mathbb{E}\Big[\Big(\sum_{t=1}^{T}\|\boldsymbol{d}_t\|^p\Big)^{\frac{1}{p}}\Big], \forall T \geq 0.
$$

Lemma 6 (Consensus errors for $\{\boldsymbol{y}_i^t\}$)**.** *We have for all $t = 0, \ldots, T$,*

$$
\begin{aligned}
&\frac{1}{n}\mathbb{E}\big[\sum_{i=1}^{n}\|\boldsymbol{y}_i^t - \bar{\boldsymbol{y}}^t\|\big] \\
&\leq 2\sqrt{2}n^{\frac{1}{2}}\big(\frac{1}{\beta}-1\big)\Big(\sum_{k=0}^{t}\lambda^{(t-k+1)p}\Big)^{\frac{1}{p}}\sigma + \frac{1}{\sqrt{n}}\big(\frac{1}{\beta}-1\big)\sum_{k=0}^{t}\lambda^{t-k+1}\mathbb{E}\big[\|\nabla F(\boldsymbol{x}^k) - \boldsymbol{v}^k\|\big].
\end{aligned}
$$

Proof. Similar to (15), we have

$$
\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{y}_i^t - \bar{\boldsymbol{y}}^t\| \leq \frac{1}{\sqrt{n}}\|\boldsymbol{y}^t - \mathbf{1}_n \otimes \bar{\boldsymbol{y}}^t\|. \qquad (18)
$$

Following from (7),

$$
\begin{aligned}
&\boldsymbol{y}^t - \mathbf{1}_n \otimes \bar{\boldsymbol{y}}^t \qquad (19) \\
&\overset{(7)}{=} \big(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\big(\boldsymbol{W}\otimes \boldsymbol{I}_d\big)(\boldsymbol{y}^{t-1} + \boldsymbol{v}^t - \boldsymbol{v}^{t-1}) \\
&= \big(\boldsymbol{W}\otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\boldsymbol{y}^{t-1} + \big(\boldsymbol{W}\otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)(\boldsymbol{v}^t - \boldsymbol{v}^{t-1}) \\
&\overset{(i)}{=} \big(\boldsymbol{W}\otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\big(\boldsymbol{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)\boldsymbol{y}^{t-1} + \big(\boldsymbol{W}\otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)(\boldsymbol{v}^t - \boldsymbol{v}^{t-1}) \\
&\overset{(ii)}{=} \sum_{k=0}^{t}\big(\boldsymbol{W}\otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)^{t-k+1}(\boldsymbol{v}^t - \boldsymbol{v}^{t-1}), \qquad (20)
\end{aligned}
$$

where we used relation 3 in Lemma 1 in (i) and used $\boldsymbol{y}_i^0 = \mathbf{0}_d, \forall i \in [n]$ in (ii). From the update in (6), we have

$$\boldsymbol{v}^t - \boldsymbol{v}^{t-1} = (\beta - 1)\boldsymbol{v}^{t-1} + (1-\beta)\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) = (1-\beta)(\boldsymbol{v}^t - \boldsymbol{v}^{t-1}) + (1-\beta)(\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) - \boldsymbol{v}^t).$$

Then, there holds,

$$\boldsymbol{v}^t - \boldsymbol{v}^{t-1} = (\frac{1}{\beta} - 1)(\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) - \boldsymbol{v}^t) = (\frac{1}{\beta} - 1)(\boldsymbol{g}(\boldsymbol{x}^t, \boldsymbol{\xi}^t) - \nabla F(\boldsymbol{x}^t) + \nabla F(\boldsymbol{x}^t) - \boldsymbol{v}^t). \quad (21)$$

Putting the relation above into (20) and applying (20) recursively, from $\boldsymbol{y}_i^0 = \bar{\boldsymbol{y}}^0$, we have

$$\begin{aligned}\|\boldsymbol{y}^t - \mathbf{1}_n \otimes \bar{\boldsymbol{y}}^t\| \leq (\frac{1}{\beta} - 1)\|\sum_{k=0}^{t} \big(\boldsymbol{W} \otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)^{t-k+1}\big(\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{\xi}^k) - \nabla F(\boldsymbol{x}^k)\big)\| \\ + (\frac{1}{\beta} - 1)\|\sum_{k=0}^{t} \big(\boldsymbol{W} \otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)^{t-k+1}\big(\nabla F(\boldsymbol{x}^k) - \boldsymbol{v}^k\big)\|\end{aligned} \quad (22)$$

We note that the first half of the right hand side above can be addressed by Lemma 5:

$$\begin{aligned}&\mathbb{E}\Big[\|\sum_{k=0}^{t} \big(\boldsymbol{W} \otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)^{t-k+1}\big(\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{\xi}^k) - \nabla F(\boldsymbol{x}^k)\big)\|\Big] \\ &\leq 2\sqrt{2}\mathbb{E}\Big[\Big(\sum_{k=0}^{t} \lambda^{(t-k+1)p}\|\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{\xi}^k) - \nabla F(\boldsymbol{x}^k)\|^p\Big)^{\frac{1}{p}}\Big].\end{aligned} \quad (23)$$

We observe that

$$\begin{aligned}&2\sqrt{2}\mathbb{E}\Big[\Big(\sum_{k=0}^{t} \lambda^{(t-k+1)p}\|\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{\xi}^k) - \nabla F(\boldsymbol{x}^k)\|^p\Big)^{\frac{1}{p}} \mid \mathcal{F}_{t-1}\Big] \\ &\leq 2\sqrt{2}\mathbb{E}\Big[\Big(\sum_{k=0}^{t} \lambda^{(t-k+1)p}\big(\sum_{i=1}^{n}\|\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k)\|\big)^p\Big)^{\frac{1}{p}} \mid \mathcal{F}_{t-1}\Big] \\ &\overset{(i)}{\leq} 2\sqrt{2}\mathbb{E}\Big[\Big(\sum_{k=0}^{t}\sum_{i=1}^{n} \lambda^{(t-k+1)p}n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k)\|^p\Big)^{\frac{1}{p}} \mid \mathcal{F}_{t-1}\Big] \\ &\overset{(ii)}{\leq} 2\sqrt{2}\Big(\mathbb{E}\big[\sum_{i=1}^{n} \lambda^p n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t) - \nabla f_i(\boldsymbol{x}_i^t)\|^p \mid \mathcal{F}_{t-1}\big] \\ &\qquad + \sum_{k=0}^{t-1}\sum_{i=1}^{n} \lambda^{(t-k+1)p}n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}^k)\|^p\Big)^{\frac{1}{p}} \\ &\overset{(iii)}{\leq} 2\sqrt{2}\Big(\lambda^p n^p\sigma^p + \sum_{k=0}^{t-1}\sum_{i=1}^{n} \lambda^{(t-k+1)p}n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k)\|^p\Big)^{\frac{1}{p}},\end{aligned} \quad (24)$$

where we used relation 5 from Lemma 1 in (i), Jensen's inequality in (ii), and Assumption 3 in (iii). From (23), taking expectations on both sides of (24), and applying the above arguments recursively from $\mathcal{F}_{t-2}$ to $\mathcal{F}_0$, we have

$$\mathbb{E}\Big[\|\sum_{k=0}^{t} \big(\boldsymbol{W} \otimes \boldsymbol{I}_d - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \boldsymbol{I}_d\big)^{t-k+1}\big(\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{\xi}^k) - \nabla F(\boldsymbol{x}^k)\big)\|\Big] \leq 2\sqrt{2}\Big(\sum_{k=0}^{t} \lambda^{(t-k+1)p}\Big)^{\frac{1}{p}} n\sigma.$$

Therefore, using the relation above, and (18), (22), we reach the desired relation. □

We then bound average gradient estimation errors $\boldsymbol{\epsilon}_1^t = \bar{\boldsymbol{v}}^t - \overline{\nabla} F(\boldsymbol{x}^t)$.

Lemma 7 (Average gradient estimation errors)**.** *For all $t = 0, \ldots, T$, we have*

$$\mathbb{E}\big[\|\bar{\boldsymbol{v}}^t - \overline{\nabla} F(\boldsymbol{x}^t)\|\big] \leq \beta^{t+1}\|\nabla f(\bar{\boldsymbol{x}}^0)\| + \frac{2\sqrt{2}}{n^{1-\frac{1}{p}}}\Big(\sum_{k=0}^{t} \beta^{(t-k)p}(1-\beta)^p\Big)^{\frac{1}{p}}\sigma + \sum_{k=0}^{t} \beta^{t-k+1}\Big(\frac{2\alpha\lambda}{1-\lambda} + \alpha\Big)L.$$

Proof. Following from the step 4 in Algorithm 1, $\forall i \in [n]$,

$$\boldsymbol{v}_i^t - \nabla f_i(\boldsymbol{x}_i^t) = \beta(\boldsymbol{v}_i^{t-1} - \nabla f_i(\boldsymbol{x}_i^{t-1})) + (1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t) - \nabla f_i(\boldsymbol{x}_i^t)) + \beta(\nabla f_i(\boldsymbol{x}_i^{t-1}) - \nabla f_i(\boldsymbol{x}_i^t)). \tag{25}$$

Averaging the above relation over $i = 1, \ldots, n$ leads to that:

$$\begin{aligned}
\boldsymbol{\epsilon}_1^t &= \bar{\boldsymbol{v}}^t - \overline{\nabla} F(\boldsymbol{x}^t) \\
&= \beta(\bar{\boldsymbol{v}}^{t-1} - \overline{\nabla} F(\boldsymbol{x}^{t-1})) + (1-\beta) \cdot \underbrace{\frac{1}{n}\sum_{i=1}^n (\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t) - \nabla f_i(\boldsymbol{x}_i^t))}_{:=\boldsymbol{s}^t \in \mathbb{R}^d} + \beta \cdot \underbrace{\frac{1}{n}\sum_{i=1}^n (\nabla f_i(\boldsymbol{x}_i^{t-1}) - \nabla f_i(\boldsymbol{x}_i^t))}_{:=\boldsymbol{z}^t \in \mathbb{R}^d} \\
&= \beta^{t+1}\boldsymbol{\epsilon}_1^{-1} + \sum_{k=0}^t \beta^{t-k}(1-\beta)\boldsymbol{s}^k + \sum_{k=0}^t \beta^{t-k+1}\boldsymbol{z}^k.
\end{aligned}$$

Taking Euclidean norms on both sides gives that

$$\|\boldsymbol{\epsilon}_1^t\| \le \beta^{t+1}\|\boldsymbol{\epsilon}_1^{-1}\| + \|\sum_{k=0}^t \beta^{t-k}(1-\beta)\boldsymbol{s}^k\| + \|\sum_{k=0}^t \beta^{t-k+1}\boldsymbol{z}^k\|. \tag{26}$$

We now bound the terms on the right hand side of (26) one by one. First,

$$\|\boldsymbol{\epsilon}_1^{-1}\| = \|\bar{\boldsymbol{v}}^{-1} - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\boldsymbol{x}}^0)\| = \|\nabla f(\bar{\boldsymbol{x}}^0)\|. \tag{27}$$

Second, notice that $\{\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\}$ is a martingale difference sequence that falls into the pursuit of Lemma 5, and thus we obtain

$$\begin{aligned}
&\mathbb{E}\Big[\|\sum_{k=0}^t \beta^{t-k}(1-\beta)\boldsymbol{s}^k\|\Big] \\
&= \frac{1}{n}\mathbb{E}\Big[\|\sum_{k=0}^t\sum_{i=1}^n \beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|\Big] \\
&\le \frac{2\sqrt{2}}{n}\mathbb{E}\Big[\Big(\sum_{k=0}^t\sum_{i=1}^n \|\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|^p\Big)^{\frac{1}{p}}\Big].
\end{aligned} \tag{28}$$

Note that

$$\begin{aligned}
&\frac{2\sqrt{2}}{n}\mathbb{E}\Big[\Big(\sum_{k=0}^t\sum_{i=1}^n \|\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|^p\Big)^{\frac{1}{p}} \mid \mathcal{F}_{t-1}\Big] \\
\overset{(i)}{\le}\ &\frac{2\sqrt{2}}{n}\Big(\mathbb{E}\Big[\sum_{k=0}^t\sum_{i=1}^n \|\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|^p \mid \mathcal{F}_{t-1}\Big]\Big)^{\frac{1}{p}} \\
\le\ &\frac{2\sqrt{2}}{n}\Big(\mathbb{E}\Big[\sum_{i=1}^n (1-\beta)^p\|(\boldsymbol{g}_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t) - \nabla f_i(\boldsymbol{x}_i^t))\|^p \mid \mathcal{F}_{t-1}\Big] \\
&\quad + \sum_{k=0}^{t-1}\sum_{i=1}^n \|\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|^p\Big)^{\frac{1}{p}} \\
\overset{(ii)}{\le}\ &\frac{2\sqrt{2}}{n}\Big(n(1-\beta)^p\sigma^p + \sum_{k=0}^{t-1}\sum_{i=1}^n \|\beta^{t-k}(1-\beta)(\boldsymbol{g}_i(\boldsymbol{x}_i^k, \boldsymbol{\xi}_i^k) - \nabla f_i(\boldsymbol{x}_i^k))\|^p\Big)^{\frac{1}{p}},
\end{aligned} \tag{29}$$

where we used Jensen's inequality in (i) and Assumption 3 in (ii). From (28), taking expectations on (29), and recursively applying the preceding arguments from $\mathcal{F}_{t-2}$ to $\mathcal{F}_0$, we have

$$\mathbb{E}\Big[\|\sum_{k=0}^t \beta^{t-k}(1-\beta)\boldsymbol{s}^k\|\Big] \le \frac{2\sqrt{2}}{n^{1-\frac{1}{p}}}\Big(\sum_{k=0}^t \beta^{(t-k)p}(1-\beta)^p\Big)^{\frac{1}{p}}\sigma. \tag{30}$$

Third,

$$
\begin{aligned}
&\|\sum_{k=0}^{t}\beta^{t-k+1}\boldsymbol{z}^k\| \\
&\le \sum_{k=0}^{t}\beta^{t-k+1}\|\frac{1}{n}\sum_{i=1}^{n}(\nabla f_i(\boldsymbol{x}_i^{k-1})-\nabla f_i(\boldsymbol{x}_i^k))\| \\
&\le \sum_{k=0}^{t}\beta^{t-k+1}\Big(\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\boldsymbol{x}_i^{k-1})-\nabla f_i(\bar{\boldsymbol{x}}^{k-1})\|+\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\bar{\boldsymbol{x}}^{k-1})-\nabla f_i(\bar{\boldsymbol{x}}^k)\| \\
&\qquad +\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\bar{\boldsymbol{x}}^k-\nabla f_i(\boldsymbol{x}_i^k)\|\Big) \\
&\overset{(i)}{\le} \sum_{k=0}^{t}\beta^{t-k+1}\Big(L\cdot\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i^{k-1}-\bar{\boldsymbol{x}}^{k-1}\|+L\cdot\frac{1}{n}\sum_{i=1}^{n}\|\bar{\boldsymbol{x}}^{k-1}-\bar{\boldsymbol{x}}^k\|+L\cdot\frac{1}{n}\sum_{i=1}^{n}\|\bar{\boldsymbol{x}}^k-\boldsymbol{x}_i^k\|\Big) \\
&\overset{(ii)}{\le} \sum_{k=0}^{t}\beta^{t-k+1}\Big(\frac{2\alpha\lambda}{1-\lambda}+\alpha\Big)L.
\end{aligned}
\tag{31}
$$

where in (i) we used Assumption 2 and in (ii) we used (14) in Lemma 4. Putting relations (27)(30)(31) together leads to the final bound for this lemma. □

We next bound the stacked gradient estimation errors.

Lemma 8 (Stacked gradient estimation errors)**.** *For all $t=0,\dots,T$, we have*

$$
\begin{aligned}
&\mathbb{E}\big[\|\boldsymbol{v}^t-\nabla F(\boldsymbol{x}^t)\|\big] \\
&\le \beta^{t+1}\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|+2\sqrt{2}\Big(\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\Big)^{\frac{1}{p}}n\sigma+n\sum_{k=0}^{t}\beta^{t-k+1}\Big(\frac{2\alpha\lambda}{1-\lambda}+\alpha\Big)L.
\end{aligned}
$$

Proof. Define $\tilde{\boldsymbol{\epsilon}}_1^t:=\boldsymbol{v}^t-\nabla F(\boldsymbol{x}^t)\in\mathbb{R}^{nd}$. Similar to (25), we have

$$
\begin{aligned}
\boldsymbol{v}^t-\nabla F(\boldsymbol{x}^t)&=\beta(\boldsymbol{v}^{t-1}-\nabla F(\boldsymbol{x}^{t-1}))+(1-\beta)\underbrace{(\boldsymbol{g}(\boldsymbol{x}^t,\boldsymbol{\xi}^t)-\nabla F(\boldsymbol{x}^t))}_{:=\tilde{\boldsymbol{s}}^t\in\mathbb{R}^{nd}}+\beta\underbrace{(\nabla F(\boldsymbol{x}^{t-1})-\nabla F(\boldsymbol{x}^t))}_{:=\tilde{\boldsymbol{z}}^t\in\mathbb{R}^{nd}} \\
&=\beta^{t+1}\tilde{\boldsymbol{\epsilon}}_1^{-1}+\sum_{k=0}^{t}\beta^{t-k}(1-\beta)\tilde{\boldsymbol{s}}^k+\sum_{k=0}^{t}\beta^{t-k+1}\tilde{\boldsymbol{z}}^k.
\end{aligned}
$$

Taking Euclidean norms on both sides gives that

$$
\|\tilde{\boldsymbol{\epsilon}}_1^t\|\le\beta^{t+1}\|\tilde{\boldsymbol{\epsilon}}_1^{-1}\|+\|\sum_{k=0}^{t}\beta^{t-k}(1-\beta)\tilde{\boldsymbol{s}}^k\|+\|\sum_{k=0}^{t}\beta^{t-k+1}\tilde{\boldsymbol{z}}^k\|.
$$

Similar to the analysis in Lemma 7, we bound the right hand side above term by term. First,

$$
\|\tilde{\boldsymbol{\epsilon}}_1^{-1}\|=\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|.
$$

Second, notice also that $\{\beta^{t-k}(1-\beta)\tilde{\boldsymbol{s}}^k\}$ is a martingale difference sequence and can be dealt with using Lemma 5. We have

$$
\begin{aligned}
&\mathbb{E}\big[\|\sum_{k=0}^{t}\beta^{t-k}(1-\beta)\tilde{\boldsymbol{s}}^k\|\big] \\
&=\mathbb{E}\big[\|\sum_{k=0}^{t}\beta^{t-k}(1-\beta)(\boldsymbol{g}(\boldsymbol{x}^t,\boldsymbol{\xi}^{t,b})-\nabla F(\boldsymbol{x}^t))\|\big] \\
&\le 2\sqrt{2}\mathbb{E}\big[\big(\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\|\boldsymbol{g}(\boldsymbol{x}^t,\boldsymbol{\xi}^t)-\nabla F(\boldsymbol{x}^t)\|^p\big)^{\frac{1}{p}}\big].
\end{aligned}
\tag{32}
$$

In addition,

$$
\begin{aligned}
&2\sqrt{2}\mathbb{E}\big[\big(\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\|\boldsymbol{g}(\boldsymbol{x}^t,\boldsymbol{\xi}^t)-\nabla F(\boldsymbol{x}^t)\|^p\big)^{\frac{1}{p}}\mid\mathcal{F}_{t-1}\big]\\
&\overset{(i)}{\leq} 2\sqrt{2}\Big(\mathbb{E}\big[\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\|\boldsymbol{g}(\boldsymbol{x}^t,\boldsymbol{\xi}^t)-\nabla F(\boldsymbol{x}^t)\|^p\big]\mid\mathcal{F}_{t-1}\Big)^{\frac{1}{p}}\\
&\overset{(ii)}{\leq} 2\sqrt{2}\Big(\mathbb{E}\big[\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\big(\sum_{i=1}^{n}\|\boldsymbol{g}_i(\boldsymbol{x}_i^t,\boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x}_i^t)\|\big)^p\big]\mid\mathcal{F}_{t-1}\Big)^{\frac{1}{p}}\\
&\overset{(iii)}{\leq} 2\sqrt{2}\Big(\mathbb{E}\big[\sum_{k=0}^{t}\sum_{i=1}^{n}\beta^{(t-k)p}(1-\beta)^p n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^t,\boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x}_i^t)\|^p\big]\mid\mathcal{F}_{t-1}\Big)^{\frac{1}{p}}\\
&=2\sqrt{2}\Big(\mathbb{E}\big[\sum_{i=1}^{n}(1-\beta)^p n^{p-1}\|\nabla\boldsymbol{g}_i(\boldsymbol{x}_i^t,\boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x}_i^t)\|^p\mid\mathcal{F}_{t-1}\big]\\
&\qquad+\sum_{k=0}^{t-1}\sum_{i=1}^{n}\beta^{(t-k)p}(1-\beta)^p n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^t,\boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x}_i^t)\|^p\Big)^{\frac{1}{p}}\\
&\leq 2\sqrt{2}\Big((1-\beta)^p n^p\sigma^p+\sum_{k=0}^{t-1}\sum_{i=1}^{n}\beta^{(t-k)p}(1-\beta)^p n^{p-1}\|\boldsymbol{g}_i(\boldsymbol{x}_i^t,\boldsymbol{\xi}_i^t)-\nabla f_i(\boldsymbol{x}_i^t)\|^p\Big)^{\frac{1}{p}},
\end{aligned}
\tag{33}
$$

where in (i) we used Jensen's inequality, and in $(ii), (iii)$ we used relations 4,and 5 in Lemma 1, respectively. Based on (32), taking expectations on both sides of (33), and applying the above arguments from $\mathcal{F}_{t-2}$ to $\mathcal{F}_0$, we obtain

$$
\mathbb{E}\big[\|\sum_{k=0}^{t}\beta^{t-k}(1-\beta)\tilde{\boldsymbol{s}}^k\|\big]\leq 2\sqrt{2}\Big(\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\Big)^{\frac{1}{p}}n\sigma.
$$

Third,

$$
\begin{aligned}
&\|\sum_{k=0}^{t}\beta^{t-k+1}\tilde{\boldsymbol{z}}^k\|\\
&\leq\sum_{k=0}^{t}\beta^{t-k+1}\|\nabla F(\boldsymbol{x}^{k-1})-\nabla F(\boldsymbol{x}^k)\|\\
&\leq\sum_{k=0}^{t}\sum_{i=1}^{n}\beta^{t-k+1}\|\nabla f_i(\boldsymbol{x}_i^{k-1})-\nabla f_i(\boldsymbol{x}_i^k)\|\\
&\leq\sum_{k=0}^{t}\sum_{i=1}^{n}\beta^{t-k+1}\Big(\|\nabla f_i(\boldsymbol{x}_i^{k-1})-\nabla f_i(\bar{\boldsymbol{x}}^{k-1})\|+\|\nabla f_i(\bar{\boldsymbol{x}}^{k-1})-\nabla f_i(\bar{\boldsymbol{x}}^k)\|+\|\nabla f_i(\boldsymbol{x}_i^k)-f_i(\bar{\boldsymbol{x}}^k)\|\Big)\\
&\overset{(i)}{\leq} n\sum_{k=0}^{t}\beta^{t-k+1}\Big(\frac{2\alpha\lambda}{1-\lambda}+\alpha\Big)L.
\end{aligned}
$$

where (i) follows from similar arguments in (30). □

Now we are ready to prove our main theorems.

Proof of Theorem 1. We observe that

$$
\begin{aligned}
\frac{1}{n}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big]&\leq\frac{1}{n}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)-\nabla f(\bar{\boldsymbol{x}}^t)\|+\|\nabla f(\bar{\boldsymbol{x}}^t)\|\big]\\
&\overset{(14)}{\leq} T\cdot\frac{\alpha\lambda L}{1-\lambda}+\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla f(\bar{\boldsymbol{x}}^t)\|\big].
\end{aligned}
\tag{34}
$$

From Lemmas 3, (13), and Lemma 6,

$$
\begin{aligned}
&\sum_{t=0}^{T-1}\alpha\|\nabla f(\bar{\boldsymbol{x}}^t)\| \\
&\le f(\bar{\boldsymbol{x}}^0)-f_*+\sum_{t=0}^{T-1}2\alpha\big(\|\boldsymbol{\epsilon}_1^t\|+\|\overline{\nabla}F(\boldsymbol{x}^t)-\nabla f(\bar{\boldsymbol{x}}^t)\|\big)+\sum_{t=0}^{T-1}\frac{\alpha}{n}\sum_{i=1}^{n}\|\bar{\boldsymbol{y}}^t-\boldsymbol{y}_i^t\|+\sum_{t=0}^{T-1}\frac{L}{2}\alpha^2 \\
&\le f(\bar{\boldsymbol{x}}^0)-f_* \\
&\quad+\sum_{t=0}^{T-1}2\alpha\Big[\beta^{t+1}\|\nabla f(\bar{\boldsymbol{x}}^0)\|+\frac{2\sqrt{2}}{n^{1-\frac{1}{p}}}\Big(\sum_{k=0}^{t}\beta^{(t-k)p}(1-\beta)^p\Big)^{\frac{1}{p}}\sigma+\sum_{k=0}^{t}\beta^{t-k+1}\Big(\frac{2\alpha\lambda}{1-\lambda}+\alpha\Big)L+\frac{\alpha\lambda L}{1-\lambda}\Big] \\
&\quad+\sum_{t=0}^{T-1}\alpha\Big[2\sqrt{2}n^{\frac{1}{2}}\big(\frac{1}{\beta}-1\big)\Big(\sum_{k=0}^{t}\lambda^{(t-k+1)p}\Big)^{\frac{1}{p}}\sigma+\frac{1}{\sqrt{n}}\big(\frac{1}{\beta}-1\big)\sum_{k=0}^{t}\lambda^{t-k+1}\mathbb{E}\big[\|\nabla F(\boldsymbol{x}^k)-\boldsymbol{v}^k\|\big]\Big] \\
&\quad+\frac{1}{2}\alpha^2LT \\
&\overset{(i)}{\le} f(\bar{\boldsymbol{x}}^0)-f_*+2\|\nabla f(\bar{\boldsymbol{x}}^0)\|\cdot\frac{\alpha}{1-\beta}+\frac{4\sqrt{2}\sigma}{n^{1-\frac{1}{p}}}\cdot\alpha(1-\beta)^{1-\frac{1}{p}}T+\frac{4L}{1-\lambda}\cdot\frac{\alpha^2T}{1-\beta}+\frac{2L}{1-\lambda}\cdot\alpha^2T \\
&\quad+\frac{2\sqrt{2}\sigma n^{\frac{1}{2}}}{(1-\lambda)^{\frac{1}{p}}}\cdot\big(\frac{1}{\beta}-1\big)\alpha T+\frac{1}{2}L\cdot\alpha^2T \\
&\quad+\frac{1}{\sqrt{n}}\big(\frac{1}{\beta}-1\big)\alpha\sum_{t=0}^{T-1}\sum_{k=0}^{t}\lambda^{t-k+1}\Big(\beta^{t+1}\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|+2\sqrt{2}n\sigma(1-\beta)^{1-\frac{1}{p}}+\frac{2nL}{1-\lambda}\cdot\frac{\alpha\beta}{1-\beta}\Big)
\end{aligned}
$$

where in (i) we used $\beta\le 1,\lambda<1$ and Lemma 8. Denote $f(\bar{\boldsymbol{x}}^0)-f_*=\Delta_0$. Dividing αT from the above relation on both sides, and putting it into (34), then rearranging terms leads to

$$
\begin{aligned}
&\frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big] \\
&\le\frac{\Delta_0}{\alpha T}+\frac{2\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{(1-\beta)T}+4\sqrt{2}\sigma\cdot\frac{(1-\beta)^{1-\frac{1}{p}}}{n^{1-\frac{1}{p}}}+\frac{4L}{1-\lambda}\cdot\frac{\alpha}{1-\beta}+\Big(\frac{3L}{1-\lambda}+\frac{L}{2}\Big)\alpha \\
&\quad+\frac{2\sqrt{2}\sigma}{(1-\lambda)^{\frac{1}{p}}}\cdot n^{\frac{1}{2}}\big(\frac{1}{\beta}-1\big)+\frac{\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{1-\lambda}\cdot\frac{\frac{1}{\beta}-1}{n^{\frac{1}{2}}}+\frac{2\sqrt{2}\sigma}{1-\lambda}\cdot n^{\frac{1}{2}}\big(\frac{1}{\beta}-1\big)(1-\beta)^{1-\frac{1}{p}} \\
&\quad+\frac{2L}{(1-\lambda)^2}\cdot n^{\frac{1}{2}}\alpha \\
&\overset{(i)}{\le}\frac{\Delta_0}{\alpha T}+\frac{2\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{(1-\beta)T}+4\sqrt{2}\sigma\cdot\frac{(1-\beta)^{1-\frac{1}{p}}}{n^{1-\frac{1}{p}}}+\frac{4L}{1-\lambda}\cdot\frac{\alpha}{1-\beta}+\frac{3.5L}{1-\lambda}\alpha \\
&\quad+\frac{20\sqrt{2}\sigma}{(1-\lambda)^{\frac{1}{p}}}\cdot n^{\frac{1}{2}}(1-\beta)+\frac{10\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{1-\lambda}\cdot\frac{1-\beta}{n^{\frac{1}{2}}}+\frac{20\sqrt{2}\sigma}{1-\lambda}\cdot n^{\frac{1}{2}}(1-\beta)^{2-\frac{1}{p}}+\frac{2L}{(1-\lambda)^2}\cdot n^{\frac{1}{2}}\alpha \\
&\overset{(ii)}{\le} O\Big(\frac{\Delta_0}{T}+\frac{2\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{(1-\beta)T}+4\sqrt{2}\sigma\cdot\frac{(1-\beta)^{1-\frac{1}{p}}}{n^{1-\frac{1}{p}}}+\sqrt{\frac{L\Delta_0}{(1-\lambda)(1-\beta)T}}+\sqrt{\frac{3.5L\Delta_0}{(1-\lambda)T}} \\
&\quad+\frac{20\sqrt{2}\sigma}{(1-\lambda)^{\frac{1}{p}}}\cdot n^{\frac{1}{2}}(1-\beta)+\frac{10\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{1-\lambda}\cdot\frac{1-\beta}{n^{\frac{1}{2}}}+\frac{\sigma}{1-\lambda}\cdot n^{\frac{1}{2}}(1-\beta)^{2-\frac{1}{p}}+\sqrt{\frac{n^{\frac{1}{2}}L\Delta_0}{(1-\lambda)^2T}}\Big) \\
&\overset{(iii)}{\le} O\Big(\frac{\Delta_0}{T}+\frac{\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{T^{\frac{2p-2}{3p-2}}}+\frac{\sigma}{n^{1-\frac{1}{p}}T^{\frac{p-1}{3p-2}}}+\sqrt{\frac{L\Delta_0}{(1-\lambda)T^{\frac{2p-2}{3p-2}}}}+\sqrt{\frac{3.5L\Delta_0}{(1-\lambda)T}} \\
&\quad+\frac{\sigma n^{\frac{1}{2}}}{(1-\lambda)^{\frac{1}{p}}T^{\frac{p}{3p-2}}}+\frac{\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{(1-\lambda)n^{\frac{1}{2}}T^{\frac{p}{3p-2}}}+\frac{\sigma}{1-\lambda}\frac{n^{\frac{1}{2}}}{T^{\frac{2p-1}{3p-2}}}+\sqrt{\frac{n^{\frac{1}{2}}L\Delta_0}{(1-\lambda)^2T}}\Big)
\end{aligned}
\tag{35}
$$

where in (i) we take $\beta \geq 1/10$, in (ii) we used

$$\alpha = \min\Big(1,\ \sqrt{\frac{\Delta_0(1-\beta)(1-\lambda)}{4LT}}, \sqrt{\frac{\Delta_0(1-\lambda)}{3.5LT}}, \sqrt{\frac{(1-\lambda)^2\Delta_0}{2n^{\frac{1}{2}}LT}}\Big), \tag{36}$$

and in (iii) we used $1-\beta = \frac{1}{T^{\frac{p}{3p-2}}}$. □

Proof of Theorem 2. Note that (35)(ii) still holds under the same choice of α in (36) and $\beta \geq 1/10$. Continuing with $1-\beta = 1/\sqrt{T}$, we have

$$\begin{aligned}
&\frac{1}{nT}\sum_{t=0}^{T-1}\sum_{i=1}^{n}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_i^t)\|\big] \\
&\leq O\Big(\frac{\Delta_0}{T} + \frac{\|\nabla f(\bar{\boldsymbol{x}}^0)\|}{\sqrt{T}} + \frac{\sigma}{n^{1-\frac{1}{p}}}\cdot\frac{1}{T^{\frac{p-1}{2p}}} + \frac{1}{T^{\frac{1}{4}}}\sqrt{\frac{L\Delta_0}{1-\lambda}} + \sqrt{\frac{3.5L\Delta_0}{(1-\lambda)T}} + \frac{\sigma n^{\frac{1}{2}}}{(1-\lambda)^{\frac{1}{p}}}\frac{1}{\sqrt{T}} + \\
&\frac{\|\nabla F(\mathbf{1}_n\otimes\bar{\boldsymbol{x}}^0)\|}{(1-\lambda)n^{\frac{1}{2}}}\cdot\frac{1}{\sqrt{T}} + \frac{\sigma n^{\frac{1}{2}}}{1-\lambda}\cdot\frac{1}{T^{\frac{2p-1}{2p}}} + \sqrt{\frac{n^{\frac{1}{2}}L\Delta_0}{(1-\lambda)^2T}}\Big).
\end{aligned}$$

Rearranging above terms leads to the desired upper bound. □

C Additional Experiment Details

C.1 Baseline descriptions

Please see Table 2 for detailed descriptions of baselines.

C.2 Additional details for synthetic experiments

Loss function. Let $(\boldsymbol{X}_{i,k}, \boldsymbol{y}_{i,k})$ denote the k-th sample of sub-dataset $(\boldsymbol{X}_i, \boldsymbol{y}_i)$ on node i. The loss function of the considered nonconvex linear regression model on this sample is $\ell(\boldsymbol{y}_{i,k} - \boldsymbol{X}_{i,k}\boldsymbol{w}_i^t)$, where the

$$\ell(r) = \begin{cases} \frac{c^2}{6}\left(1 - \left[1 - \left(\frac{r}{c}\right)^2\right]^3\right) & \text{if } |r| \leq c, \\ \frac{c^2}{6} & \text{otherwise} \end{cases},$$

and we use the suggested value $c = 4.6851$ in the robust statistics literature.

Hyperparameter tuning. Please see Table 3 for hyperparameter searching ranges for this experiment.

Hardware. We ran this experiment on Mac OS X 15.3, CPU M4 10 Cores, RAM 16GB.

C.3 Additional details for decentralized training of Transformers

Transformer architecture. We consider the following decoder-only Transformer model (GPT): vocabulary size is 10208, context length is 64, embedding size is 128, number of attention heads is 4, number of attention layers is 2, the linear projection dimension within attention block is 512, and LayerNorm is applied after the 2nd attention block. The total number of parameters of this model is 3018240.

Hyperparameter tuning. See Table 4 for our grid search range for algorithm hyperparameters.

Hardware. We simulate the distributed training on one NVIDIA H100 GPU, using PyTorch 3.2 with CUDA 12. The total hyperparameter search and training procedure took around 100 GPU hours.

Table 2: Summary of Baseline Methods

Method	**Parallel update on node i**	**Hyper-parameters**
`DSGD`	$\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\big(\boldsymbol{x}_r^t - \alpha g_r(\boldsymbol{x}_r^t, \boldsymbol{\xi}_r^t)\big)$	α: constant stepsize
`DSGD-GClip`	$\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\boldsymbol{x}_r^t - \alpha\,\mathsf{clip}(g_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t), \tau)$	α, τ: stepsize α, and ℓ_2 clipping levels τ
`DSGD-CClip`	$\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\boldsymbol{x}_r^t - \alpha\,\mathsf{clip}(g_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t), \tau)$	α, τ: stepsize α, and component-wise clipping levels τ
`DSGD-Clip`	$\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\boldsymbol{x}_r^t - \alpha_t\,\mathsf{clip}(g_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t), \tau_t)$	α, τ: stepsize $\alpha_t = \alpha/(t+1)$, and ℓ_2 clipping levels $\tau_t = \tau(t+1)^{2/5}$
`GT-DSGD`	$\boldsymbol{y}_i^{t+1} = \sum_{r=1}^n w_{ir}\big(\boldsymbol{y}_r^t + g_r(\boldsymbol{x}_r^t, \boldsymbol{\xi}_r^t) - g_r(\boldsymbol{x}_r^{t-1}, \boldsymbol{\xi}_r^{t-1})\big)$ $\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\big(\boldsymbol{x}_i^t - \alpha\boldsymbol{y}_r^{t+1}\big)$	α: constant stepsize
`GT-Adam`	$\boldsymbol{m}_i^{t+1} = \beta_1\boldsymbol{m}_i^t + (1-\beta_1)\boldsymbol{s}_i^t$ $\boldsymbol{v}_i^{t+1} = \min\big(\beta_2\boldsymbol{v}_i^t + (1-\beta_2)\boldsymbol{s}_i^t \odot \boldsymbol{s}_i^t, G\big)$ $\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\boldsymbol{x}_r^t - \alpha\frac{\boldsymbol{m}_i^{t+1}}{\sqrt{\boldsymbol{v}_i^{t+1}+\epsilon}}$ $\boldsymbol{g}_i^{t+1} = \nabla f_i(\boldsymbol{x}_i^{t+1})$ $\boldsymbol{s}_i^{t+1} = \sum_{r=1}^n w_{ir}\boldsymbol{s}_r^t + \boldsymbol{g}_i^{t+1} - \boldsymbol{g}_i^t$	α, G: constant stepsize α, and upper bound G, stabilization factor ϵ
`QG-DSGDm`	$\boldsymbol{m}_i^{t+1} = \beta\hat{\boldsymbol{m}}_i^t + g_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t)$ $\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\big(\boldsymbol{x}_i^t - \eta\boldsymbol{m}_i^{t+1}\big)$ $\boldsymbol{d}_i^t = (\boldsymbol{x}_i^{t+1} - \boldsymbol{x}_i^t)/\eta$ $\hat{\boldsymbol{m}}_i^{t+1} = \mu\hat{\boldsymbol{m}}_i^t + (1-\mu)\boldsymbol{d}_i^t$	η, β, μ: constant stepsize η, momentum parametes β, μ
`SClip-EF-Network`	$\boldsymbol{m}_i^{t+1} = \beta_t\boldsymbol{m}_i^t + (1-\beta_t)\boldsymbol{\Psi}_t(g_i(\boldsymbol{x}_i^t, \boldsymbol{\xi}_i^t) - \boldsymbol{m}_i^t)$ $\boldsymbol{x}_i^{t+1} = \sum_{r=1}^n w_{ir}\big(\boldsymbol{x}_r^t - \alpha_t\boldsymbol{m}_r^{t+1}\big)$	$c_\varphi, \tau, \alpha, \beta$: Component-wise smooth clipping operator: $\boldsymbol{\Psi}_t(y) = \frac{c_\varphi}{\sqrt{t+1}}\frac{y}{\sqrt{y^2+\tau(t+1)^{3/5}}}$, stepsize $\alpha_t = \alpha/(t+1)^{1/5}$, momentum stepsize $\beta_t = \beta/\sqrt{t+1}$.

Table 3: Hyperparameter grid search in synthetic experiments

Method	**Hyperparameter search set**
DSGD	$\alpha \in \{10^{-5}, 5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}$
DSGD-Clip	$\alpha \in \{10^{-5}, 5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}, \tau \in \{10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10, 50, 10^2\}$
GT-DSGD	$\alpha \in \{10^{-5}, 5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}$
GT-NSGDm	$\alpha \in \{10^{-5}, 5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}, \beta \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$
SClip-EF-Network	$\alpha \in \{10^{-3}, 10^{-2}, 0.1, 1, 10, 30\}, \beta \in \{10^{-2}, 0.1, 0.5, 0.8, 0.99\}, c_\varphi \in \{1, 5, 10, 20, 30, 50\}, \tau \in \{0.1, 1, 10, 50, 100\}$

Table 4: Hyperparameter grid search in decentralized training of Transformers

Method	**Hyperparameter search set**
DSGD	$\alpha \in \{10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1\}$
DSGD-GClip	$\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}, \tau \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$
DSGD-CClip	$\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}, \tau \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$
DSGD-Clip	$\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}, \tau \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$
GT-DSGD	$\alpha \in \{10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1\}$
GT-Adam	$\alpha \in \{5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}, G \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}, \epsilon = 10^{-8}$
QG-DSGDm	$\eta \in \{5*10^{-5}, 10^{-4}, 5*10^{-4}, 10^{-3}, 5*10^{-3}, 10^{-2}, 5*10^{-2}, 10^{-1}, 0.5, 1, 5, 10\}, \beta = \mu \in \{0.01, 0.2, 0.4, 0.6, 0.8, 0.99\}$
GT-NSGDm	$\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}, \beta \in \{0.01, 0.2, 0.4, 0.6, 0.8, 0.99\}$
SClip-EF-Network	$\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}, \beta \in \{0.01, 0.4, 0.8, 0.99\}, c_\varphi \in \{0.1, 1, 10, 10^2\}, \tau \in \{0.01, 0.1, 1, 10\}$