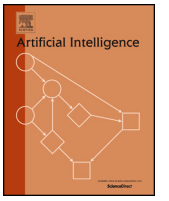
Artificial Intelligence 332 (2024) 104130

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



A unified momentum-based paradigm of decentralized SGD for non-convex models and heterogeneous data

Haizhou Du *, Chaoqian Cheng, Chengdong Ni

School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China

ARTICLE INFO

Keywords:
Decentralized training
Non-convexity optimization
Heterogeneous data
Gradient tracking
Momentum technique

ABSTRACT

Emerging distributed applications recently boosted the development of decentralized machine learning, especially in IoT and edge computing fields. In real-world scenarios, the common problems of non-convexity and data heterogeneity result in inefficiency, performance degradation, and development stagnation. The bulk of studies concentrate on one of the issues mentioned above without having a more general framework that has been proven optimal. To this end, we propose a unified paradigm called UMP, which comprises two algorithms `D-SUM` and `GT-DSUM` based on the momentum technique with decentralized stochastic gradient descent (SGD). The former provides a convergence guarantee for general non-convex objectives, while the latter is extended by introducing gradient tracking, which estimates the global optimization direction to mitigate data heterogeneity (*i.e.*, distribution drift). We can cover most momentum-based variants based on the classical heavy ball or Nesterov's acceleration with different parameters in UMP. In theory, we rigorously provide the convergence analysis of these two approaches for non-convex objectives and conduct extensive experiments, demonstrating a significant improvement in model accuracy up to 57.6% compared to other methods in practice.

1. Introduction

Distributed machine learning (DML) has emerged as an important paradigm in large-scale machine learning [53,64,39]. In terms of how to aggregate the model parameters/gradients among workers, researchers classify the system architecture into two main classes: parameter server (PS) and decentralized. The former is generally considered as the centralized paradigm where the central server acts as a coordinator for convenience, while the latter allows communication in a peer-to-peer fashion over an underlying topology, which could guarantee the model consistency across all workers with better scalability.

Meanwhile, multiple complementary studies [9,61,16] have focused on the issues of DML mainly based on the following two key aspects, which potentially deteriorate the model performance:

- *The property of non-convex objectives has emerged as a crucial component* in deep learning, in particular in distributed scenarios [19, 30]. Although some standard theoretical results have been obtained for convex models [50,6,49], much less is applicable in non-convex settings. Moreover, non-convex optimization problems may exhibit multiple local optima, which can lead to suboptimal solutions and negatively impact the overall accuracy of the model [46]. Additionally, it also induces model instability and impedes the training process since the algorithms under non-convexity optimization may affect lossy generalization [1].

* Corresponding author.
E-mail address: duhaizhou@shiep.edu.cn (H. Du).

https://doi.org/10.1016/j.artint.2024.104130
Received 18 May 2023; Received in revised form 26 February 2024; Accepted 3 April 2024

Available online 17 April 2024

- *It is well known that heterogeneity in the data is one of the key challenges in distributed training, resulting in a slow and unstable convergence as well as poor model generalization.* It is inevitable that the distribution of each worker varies considerably and incurs drift problems. However, there still exists a gap between the disappointing empirical performance and the degree of data heterogeneity [41,31,8].

Thus, it is necessary to develop the theoretical convergence results for the optimization of the distributed, non-convex objectives, and highly desirable to improve the real-world performance of DML from a comprehensive perspective by taking non-convexity and data heterogeneity into account.

Motivated by the momentum's effects on optimal convergence complexity and empirical evaluation successes [23,61,15,31], we propose **UMP**, a **U**nified, **M**omentum-based **P**aradigm in decentralized learning without considering the communication overhead throughout the paper. It consists of two algorithms named `D-SUM` and `GT-DSUM`. The former one `D-SUM` explores the potential of momentum by maintaining and scaling the momentum buffer to sharpen the loss landscape significantly and overcomes the restrictions of non-convexity, leading to better performance and faster convergence rate in the non-convex settings. Our latter algorithm `GT-DSUM` also aims to mitigate the impact of data heterogeneity on the discrepancy of local model parameters by introducing the gradient tracking (GT) technique [7]. The core insight is that the variance between workers is decreasing while the local gradient asymptotically aligns with the global optimization direction independent on the heterogeneity of the data. `GT-DSUM` accelerates decentralized learning achieving better model performance under the negative impacts generated from the non-convexity and different degrees of non-IID.

This paper makes the following **main contributions**:

- We propose a unified momentum-based paradigm UMP with two algorithms `D-SUM` and `GT-DSUM` for dealing with the poor performance caused by the non-convexity and the degree of non-IID. To be specific, the former provides a convergence guarantee for general non-convex objectives, while the latter is extended by introducing gradient tracking, which estimates the global optimization direction to mitigate data heterogeneity. Moreover, a variety of algorithms with the momentum technique could be obtained by specifying the parameters of our base algorithms.
- We design the first algorithm `D-SUM`, which achieves good model performance, demonstrating its applicability in terms of efficacy and efficiency. We provide its convergence result under the non-convex cases.
- Our second one `GT-DSUM`, which is robust to the distribution drift problem by applying the GT technique, is being further developed. We also rigorously prove its convergence bound in smooth, non-convex settings.
- We additionally conduct extensive experiments to evaluate the performance of UMP on common models, datasets, and dynamic real-world settings. Experimental results demonstrate that `D-SUM` and `GT-DSUM` improve the model accuracy by up to 35.8% and 57.6% respectively under different non-IID degrees compared with the well-known decentralized baselines. `GT-DSUM` performs better than `D-SUM` on model generalization across training tasks suffering from data skewness.

The remaining part of this paper is organized as follows. We first present some related works in Section 2. Section 3 presents the UMP framework and describes the rationale of our two proposed algorithms `D-SUM` and `GT-DSUM`. In Section 4, we theoretically provide the convergence analysis. Section 5 gives a comprehensive evaluation of our work and finally we draw the conclusion in Section 6.

2. Related work

SGD variants with momentum. It is worth noting that for existing deterministic optimization, there is much analysis on momentum SGD. Existing methods on the momentum SGD are widely exploded to develop the convergence analysis, and demonstrates that it is a critical component since it outperforms the conventional SGD with better generalization and faster convergence rate in practice [45,31]. The convex optimization of convergence analysis was first established [12] several years ago. A novel model-based method with heavy ball momentum [6] is presented proving its acceleration in (weakly) convex objectives. To fill the theory-practice gap, an adaptive momentum-based technique [49] is researched, achieving a speedup in convex settings and improving performance in deep networks. We also note that a large volume of work consists in understanding and applying the convergence effect of momentum methods when optimizing non-convex functions. Jelassi et al. [18] recently provide a novel perspective that momentum is beneficial in datasets since its historical gradients memorize those samples with the same features and different margins. While other researchers [28,5] assert that momentum holds the noise reduction property so that it cancels out the noise from the gradient and achieves improving generalization in neural networks. `Local SCGDM` [11] firstly applies momentum into the federated compositional optimization, achieving a competitive sampling complexity and communication complexity. A second-order SGD method using momentum [51] corrects the bias and leads to variance reduction in large-scale deep learning tasks. Ghosh et al. [13] provides a new analysis for implicit regularization for the heavy ball momentum gradient descent, showing that the momentum updates for gradient descent and its stochastic counterpart are closer to the modified regularized gradient flow than previously known. In addition, many studies [14,60] give a unified analysis of momentum SGD variants, filling the gap of how the momentum affects the performance of a variety of algorithms.

In distributed scenarios, prior works empirically apply momentum SGD with a locally maintained buffer rather than vanilla SGD to achieve better testing accuracy [23]. For instance, `PD-SGDm` [10] uses periodic communication and momentum in decentralized training, achieving efficient communication regarding the number of workers. `DFedAvgM` [44] extends `FedAvg` to the

Table 1
A review of decentralized optimization methods on the convergence result under the general non-convex cases. T, K denote the number of total training epochs and local updates, respectively. G is related with the bounded stochastic gradient, *i.e.*, $\mathbb{E}_{\xi\sim D}\|\nabla F(\mathbf{x},\xi)\|^2 \le G^2$. We consider the dynamic topology environment. Thus, our convergence results include ρ, whereas others not. The color indicates our proposed methods. (For interpretation of the references to color please refer to the web version of this article.)

Method	Convergence Result
PD-DSGDm [10]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\eta^2K^2G^2\right)$
LD-SGD [27]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\eta^2K\left(\sigma^2+\zeta^2\right)\right)$
PR-SGDm [61]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\eta^2K\sigma^2\right)+\mathcal{O}\left(\eta^2K^2\zeta^2\right)$
QG-DSGDm [31]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\eta^2K\left(\sigma^2+\zeta^2\right)\right)$
DFedAvgM [44]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\eta^2K(\sigma^2+\zeta^2+G^2)}{T^{3/2}}\right)$
Local SGDAm [42]	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\eta K(\sigma^2+\zeta^2)}{T}\right)$
D-SUM (Ours)	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta(1+\alpha^2)\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\eta^2\zeta^2}{n\rho^2}\right)+\mathcal{O}\left(\frac{\eta^2\sigma^2(1+\alpha^2)}{n\rho}\right)$
GT-DSUM (Ours)	$\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta(1+\alpha^2)\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\zeta^2+\sigma^2}{n\rho}\right)+\mathcal{O}\left(\frac{\varepsilon}{n}\right)$

decentralized setting and uses momentum to guarantee reducing communication costs. PR-SGDm [61] provides a rigorous convergence analysis on momentum-based SGD in a decentralized pattern, showing its ability to achieve linear speedup in both identical and non-identical data sets. Another similar work [2] introduces DMSGD to decentralized optimization through local and consensus momentum. DecentLaM [63], a decentralized large-batch momentum SGD, addresses the momentum-incurred bias in vanilla decentralized momentum SGD. Momentum Tracking [47], a decentralized method with momentum, exhibits a proven convergence rate independent of data heterogeneity. SlowMo [54] proposes to perform a periodical, slow update with momentum for both centralized or decentralized paradigms. QG-SGDm [31] instead introduces an efficient momentum-based method to approximate the global optimization direction without causing extra communication overhead. Local SGDAm [42] extends SGDA to the decentralized setting and uses momentum to mitigate the impact of stochastic gradient noise for NC-PL minimax problems.

In contrast to the above research, our work integrates some widely used momentum methods into a general framework under a decentralized setting. Each worker stores its past momentum buffer for the future update and the consensus model can converge eventually. Besides, there is no existing work that theoretically provides a unified analysis on how momentum acts on model generalization in decentralized training. Therefore, it is challenging but non-trivial to pursue better training performance with the effect of momentum.

Algorithmic approaches for data heterogeneity. Achieving a competitive performance under the non-IID assumption is one of the challenging problems in distributed training because workers suffer from different degrees of data skewness generated from heterogeneous users and locations, causing distribution drift [19] and harming the convergence [16]. FedProx [26] is proposed in federated learning by adding a proximal term to the objective that eliminates data heterogeneity. Besides, there are several large families of algorithms that adjust gradient directions to mitigate data heterogeneity. Gradient tracking technique [7,56,57,38,21,32] adds gradient trackers to the conventional decentralized SGD, estimating the global update corrections asymptotically, and its convergence rate only depends on the data heterogeneity at the initial point. Similarly, SARSH/SPIDER-type schemes [9,55] are two branches of decentralized variance reduced methods in handling non-IID. Exact-diffusion [62,48], another popular algorithmic mechanism, addresses data variance among workers in decentralized training and exhibits competitive convergence rates. A unified framework [17] recovers many existing methods by leveraging a proper sampling strategy on update rules to tackle data heterogeneity. RelaySum [52] is inspired by the spanning trees to ensure asymptotic uniform distribution of covering workers. A novel algorithm named CGA [8] aggregates cross-gradient information to demonstrate its efficiency in the non-IID cases. QG-SGDm [31] mitigates the influence of data heterogeneity by locally approximating the global optimization direction without causing extra communication overhead. In particular, recent research [33,41,29,59] find that under highly non-IID scenarios, the local classifier is biased and far from optimal. These methods can alleviate this case to some extent.

The above approaches are orthogonal to the scope of base optimizers (*e.g.*, SGD) and can be used in conjunction with them for achieving better performance. However, further investigation remains on analyzing their oracle complexities [56] under different cases (e.g., non-smooth, non-convex).

We summarize the convergence rate of different decentralized optimization methods with momentum for heterogeneous data in Table 1.

3. The unified paradigm: UMP

In this section, we first begin with the notation and revisit two momentum approaches: the heavy ball (HB) method [37] and Nesterov's momentum [35]. Inspired by them, we generalize a unified momentum-based paradigm with two algorithms D-SUM and

GT-DSUM, which could cover the above two classical methods and other momentum-based variants, aiming to address issues on non-convexity and data heterogeneity in real-world decentralized learning applications. Finally, we provide the convergence result that they could converge almost to a stationary point for general smooth, non-convex objectives.

3.1. Notation and preliminary

To better demonstrate the applicable effect in real-world complex scenarios, we consider a decentralized setting with a network topology where n workers jointly deal with an optimization problem. Assume that for every worker i, it holds its own datasets drawn from D_i distribution, which corresponds to data heterogeneity. Let $f_i : \mathbb{R}^d \to \mathbb{R}$ be the training datasets loss function of worker i and can be given in a stochastic form $\mathbb{E}_{\xi_i \sim D_i}\left[\nabla F_i(\mathbf{x}, \xi_i)\right] = \nabla f_i(\mathbf{x})$, where $F_i(\mathbf{x}, \xi_i)$ is the per-data loss function related with the mini-batch sample $\xi_i \sim D_i$. Then, we formulate the empirical risk minimization with sum-structure objectives:

$$f^* = \min_{\mathbf{x}\in\mathbb{R}^d}\left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n}\left[f_i(\mathbf{x}) = \mathbb{E}_{\xi_i\sim D_i} F_i(\mathbf{x}, \xi_i)\right]\right]. \tag{1}$$

Among workers, there is an underlying topology graph $\mathbf{W} \in \mathbb{R}^{n\times n}$, which is convenient to encode the communication between arbitrary two workers, *i.e.*, we let $w_{ij} = 0$ if and only if worker i and j are not connected.

Definition 1 *(Consensus Matrix [21]).* A matrix with non-negative entries $\mathbf{W} \in [0, 1]^{n\times n}$ that is symmetric ($\mathbf{W} = \mathbf{W}^\top$), and doubly stochastic ($\mathbf{W1} = \mathbf{1}, \mathbf{1}^\top\mathbf{W} = \mathbf{1}$), where $\mathbf{1}$ denotes the all-one vector in $\mathbb{R}^n$.

Throughout the paper, we use the notation $\mathbf{x}_i^{(t),\tau}$ to denote the sequence of model parameters on worker i at the τ-th local update in epoch t. For any vector $\mathbf{a}_i \in \mathbb{R}^d$, we denote its model averaging $\bar{\mathbf{a}} = \frac{1}{n}\sum_{i=1}^n \mathbf{a}_i$. Let $\|\cdot\|$, $\|\cdot\|_F$ denote the l_2 vector norm and Frobenius matrix norm, respectively.

For ease of presentation, we apply both vector and matrix notation whenever it is more convenient. We denote by a capital letter for the matrix form combining by $\mathbf{a}_i$ as follows,

$$\mathbf{A} = \left[\mathbf{a}_1, \cdots, \mathbf{a}_n\right] \in \mathbb{R}^{d\times n}, \quad \bar{\mathbf{A}} = [\bar{\mathbf{a}}, \cdots, \bar{\mathbf{a}}] = \mathbf{A}\frac{1}{n}\mathbf{1}\mathbf{1}^\top. \tag{2}$$

The introduction of a *momentum* term is one of the most common modifications, which is viewed as a critical component for training the state-of-the-art deep neural networks [39,31]. Corresponding to its empirical success, momentum attempts to enhance the convergence rate on non-convex objectives by setting the optimized searching direction as the combination of stochastic gradient and historical directions.

The HB method (*i.e.*, also known as Polyak's momentum) is first proposed for the smooth and convex settings, written as

$$\begin{cases} \mathbf{u}_i^{(t+1)} = \beta\mathbf{u}_i^{(t)} + \mathbf{g}_i^{(t)} \\ \mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta\mathbf{u}_i^{(t+1)}, \end{cases} \tag{3}$$

where $\mathbf{u}_i^{(t)}$, $\mathbf{g}_i^{(t)}$ are denoted as the momentum buffer, and the stochastic gradient of worker i at epoch t, respectively. η presents the learning rate. The momentum variable β adjusts the magnitude of updating direction provided by the past information estimation with the stochastic gradient, indicating the direction of the steepest descent. Equivalently, (3) can be also updated below

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta\mathbf{g}_i^{(t)} + \beta\left(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t-1)}\right), \tag{4}$$

when $t \geq 1$. Holding the past gradient values, this style of update can have better stability to some extent and enables improvement compared with some vanilla SGD methods [4].

Another kind of technique called Nesterov's shows that choosing with suitable parameters, the extrapolation step can be accelerated from $\mathcal{O}\left(\frac{1}{t}\right)$ to $\mathcal{O}\left(\frac{1}{t^2}\right)$, which is the optimal rate for the smooth convex problems. Concretely, its update step is described as follows

$$\begin{cases} \mathbf{u}_i^{(t+1)} = \beta\mathbf{u}_i^{(t)} + \mathbf{g}_i^{(t)} \\ \mathbf{v}_i^{(t+1)} = \beta\mathbf{u}_i^{(t+1)} + \mathbf{g}_i^{(t)} \\ \mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta\mathbf{v}_i^{(t+1)}. \end{cases} \tag{5}$$

The model parameters are updated by introducing the momentum vector $\mathbf{u}_i$ and extra auxiliary $\mathbf{v}_i$ sequences. Compared with (3), through decaying the momentum buffer $\mathbf{u}_i^{(t)}$, it effectively improves the rate of convergence without causing oscillations. Similarly, the above steps can be written as

Algorithm 1: vanilla SGD and D-SUM; colors indicate the two alternative variants. (For interpretation of the references to color please refer to the web version of this article.)

Input: $\forall i$, initialize $\mathbf{x}_i^{(0),0} = \mathbf{v}_i^{(0),0} = \mathbf{x}_0$; constant parameters η, α, and β; $\forall i, j$, consensus matrix $\mathbf{W}$ with entries w_{ij}; the number of epochs T and local steps K.

for $t \in \{0, \cdots, T-1\}$ **at worker** i **in parallel do**
  Set $\mathbf{x}_i^{(t),0} = \mathbf{x}_i^{(t)}, \mathbf{v}_i^{(t),0} = \mathbf{v}_i^{(t)}$.
  for $\tau \in \{0, \cdots, K-1\}$ **do**
    Sample $\xi_i^{(t),\tau}$ and compute $\mathbf{g}_i^{(t),\tau} = \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i^{(t),\tau})$.
    $\mathbf{x}^{(t),\tau+1} = \mathbf{x}^{(t),\tau} - \eta \mathbf{g}_i^{(t),\tau}$.
    Compute local model $\mathbf{x}_i^{(t),\tau}$ from (7).
  end
  Perform gossip averaging via (8).
  $\mathbf{v}_i^{(t+1)} = \sum_{j=1}^{n} w_{ij} \mathbf{v}_j^{(t),K}$.
end

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta \mathbf{g}_i^{(t)} + \beta \left(\mathbf{x}_i^{(t)} - \eta \mathbf{g}_i^{(t)} - \mathbf{x}_i^{(t-1)} + \eta \mathbf{g}_i^{(t-1)} \right). \tag{6}$$

Based on (4) and (6), it is not difficult to observe that the former could evaluate the gradient and add momentum simultaneously, while the latter applies momentum after evaluating gradients, which intuitively causes more computation cost. Meanwhile, leveraging the idea of HB momentum, Nesterov's acceleration brings us closer to the minimum (*i.e.*, $\mathbf{x}^*$) by introducing an additional gradient descent rule by adding the subtracted gradients $\eta(\mathbf{g}_i^{(t-1)} - \mathbf{g}_i^{(t)})$ for general convex cases. The above two basic momentum-based approaches are first investigated in convex settings, showing their advantage compared with vanilla SGD. However, a comprehensive analysis of momentum-based SGD under non-convex conditions in common real-world scenarios is still lacking.

3.2. D-SUM algorithm

In this section, we present UMP and its first algorithm D-SUM, which is employed in decentralized training under non-convex cases.

Under each epoch, workers first perform K local updates using different optimizers (*i.e.*, SGD, Adam [20], etc.) with or without momentum. In this paper, we mainly focus on the momentum-based SGD variants, which are demonstrated in (3), and (5) for example. From a comprehensive view, we apply the key update of the stochastic unified momentum (SUM) is according to

$$\begin{cases} \mathbf{u}_i^{(t),\tau+1} = \mathbf{x}_i^{(t),\tau} - \eta \mathbf{g}_i^{(t),\tau} \\ \mathbf{v}_{i,}^{(t),\tau+1} = \mathbf{x}_i^{(t),\tau} - \alpha\eta \mathbf{g}_i^{(t),\tau} \\ \mathbf{x}_i^{(t),\tau+1} = \mathbf{u}_i^{(t),\tau+1} + \beta \left(\mathbf{v}_i^{(t),\tau+1} - \mathbf{v}_i^{(t),\tau} \right), \end{cases} \tag{7}$$

where $\alpha \geq 0$, and $\beta \in [0, 1)$. $\mathbf{a}_i^{(t),\tau}$ ($\mathbf{a}_i$ could be the instance for $\mathbf{x}_i$, $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{g}_i$) is denoted as the related variables for worker i after τ local updates in epoch t. After K local steps, worker i communicates with its neighbors according to the communication pattern $\mathbf{W}$ for exchanging their local model parameters. We call this synchronization operation as gossip averaging which can be compactly written as

$$\mathbf{x}_i^{(t+1)} = \sum_{j=1}^{n} w_{ij} \mathbf{x}_j^{(t),K}. \tag{8}$$

To present the difference between vanilla SGD and stochastic unified momentum in (7), we summarize the training procedure in Algorithm 1. Specially, we update the auxiliary variable sequences $\{\mathbf{v}_i\}$ for any worker i by using the same gossip synchronization as in (8) interpreted as a restart in the next training epoch to simplify theoretical analysis. The specific algorithm instance is obtained by tuning the hyperparameters α, β, η, K, and n:

- We cover the basic Heavy Ball method [37] when $n = 1$ and setting $\alpha = 0$.
- We cover the basic Nesterov's momentum [35] when $n = 1$ and setting $\alpha = 1$.
- We cover the standard mini-batch SGD with momentum acceleration [60] when $n = 1$ and setting $K = 1$.
- We cover PR-SGDm [61] when $n > 1$ and setting $\alpha = 1$ or $\alpha = 0$.

However, there is no theoretical or empirical analysis to demonstrate that the momentum gets rid of heterogeneity which degrades the distributed deep training due to the discrepancies between local activation statistics [16]. Not only taking non-convex functions into account, but we also incorporate a technique that is agnostic to data heterogeneity, gradient tracking into D-SUM to alleviate the impact of heterogeneous data in decentralized training for better model generalization in the following.

Algorithm 2: GT-DSUM.

Input: $\forall i$, initialize $\mathbf{x}_i^{(0),0} = \mathbf{v}_i^{(0),0} = \mathbf{x}_0$, $\mathbf{y}_i^{(0)} = \mathbf{g}_i^{(0),0} = \nabla F_i(\mathbf{x}_i^{(0),0}, \xi_i^{(0),0})$, and $\mathbf{d}_i^{(-1)} = \mathbf{0}_p$; constant parameters $\alpha \geq 0, \beta \in [0,1), \eta, \lambda \in [0,1]$; $\forall i, j$, consensus matrix $\mathbf{W}$ with entries w_{ij}; the number of epochs T, and local steps K.

1 **for** $t \in \{0, \cdots, T-1\}$ **at worker** i **in parallel do**
2 **for** $\tau \in \{0, \cdots, K-1\}$ **do**
3 Sample $\xi_i^{(t),\tau}$, compute $\mathbf{g}_i^{(t),\tau} = \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i^{(t),\tau})$.
4 $\mathbf{m}_i^{(t),\tau} = \lambda \mathbf{g}_i^{(t),\tau} + (1-\lambda)\mathbf{y}_i^{(t)}$.
5 Substitute $\mathbf{g}_i^{(t),\tau}$ with $\mathbf{m}_i^{(t),\tau}$ as the local gradient estimation, perform (7).
6 **end**
7 Gossip averaging $\mathbf{x}_i^{(t+1)} = \sum_{j=1}^n w_{ij} \mathbf{x}_j^{(t),K}$.
8 $\mathbf{v}_i^{(t+1)} = \sum_{j=1}^n w_{ij} \mathbf{v}_j^{(t),K}$.
9 $\mathbf{d}_i^{(t)} = \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{K\eta}$.
10 Gradient tracking based on $\mathbf{y}_i^{(t+1)} = \sum_{j=1}^n w_{ij} \left(\mathbf{y}_j^{(t)} + \mathbf{d}_j^{(t)} - \mathbf{d}_j^{(t-1)} \right)$.
11 **end**

3.3. GT-DSUM algorithm

In this subsection, we go further the fact that heterogeneity hinders the local momentum acceleration [31] and provides our second algorithm in UMP, termed GT-DSUM, which aims to generalize the consensus model parameters better and alleviate the impact of heterogeneous data by applying the gradient tracking technique.

Taking the discrepancies between workers' local data partition into account, GT introduces an extra worker-sided auxiliary variable $\mathbf{y}_i^{(t)}, \forall i$ aiming to asymptotically track the average of ∇f_i assuming the local accurate gradients are accessible at any epoch t. Intuitively, GT is agnostic to the heterogeneity, while $\mathbf{y}_i^{(t)}$ is approximately equivalent to the global gradient direction along with the epoch t increases. Inspired by this, we introduce GT into D-SUM, yields GT-DSUM. Concretely, we normalize the applied gradient $\mathbf{m}_i^{(t),\tau}$ using the mini-batch gradient $\mathbf{g}_i^{(t),\tau}$, and the $\mathbf{y}_i^{(t)}$ with the dampening factor λ to highlight the necessity of local updates. The detailed algorithm is described in Algorithm 2. Within local updates, the model parameters are updated on line 5 with D-SUM but using a normalization term $\mathbf{m}_i^{(t),\tau}$. Line 7 and 8 are the same as the basic D-SUM procedures in Algorithm 1. For GT-DSUM, we apply the difference of two consecutive synchronized models shown in line 9 to update the gradient tracker variable in line 10 using the gossip-liked style [57,56]. Especially, when $K = 1$, $\lambda = 0$ and $\beta = 0$, the Algorithm 2 can be reduced to the original GT algorithm [21] instance.

By our observations, GT-DSUM requires more rounds of communication per main iteration, as opposed to 1 round in D-SUM. Since Algorithm 1 and 2 employ multiple consensus steps from parameters exchanging which significantly increase communication cost, we apply the communication compression technique GRACE [58] to trade-off between model generalization and communication overhead in Section 5.

3.4. Theoretical results

In what follows, we present the convergence results of two algorithms in the UMP for general non-convex settings. The detailed proof is presented in Section 4. Firstly, we state our assumptions throughout the paper.

Assumption 1 *(L-smooth).* For each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is differentiable, and there exists a constant $L > 0$ such that for each $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$: $\left\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\right\| \leq L \left\|\mathbf{x} - \mathbf{x}'\right\|$.

Assumption 2 *(Bounded variances).* We assume that there exists σ>0 and ζ>0 for any $i, \mathbf{x} \in \mathbb{R}^d$ such that $\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \left\|\nabla F_i(\mathbf{x}, \xi_i) - \nabla f_i(\mathbf{x})\right\|^2 \leq \sigma^2$, and $\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\right\|^2 \leq \zeta^2$.

Assumptions 1 and 2 are standard in general non-convex objective literature [31,61,22] in order to ensure the basis of loss functions continuous and the limited influence of heterogeneity among distributed scenarios. Noted that when $\zeta = 0$, we have $\nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x})$, *i.e.*, it reduces to the case of IID data distribution across all participating workers.

Assumption 3. The mixing matrix is doubly stochastic by Definition 1. Further, define $\bar{\mathbf{Z}} = \mathbf{Z}\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ for any matrix $\mathbf{Z} \in \mathbb{R}^{d \times n}$. Then, the mixing matrix satisfies $\mathbb{E}_{\mathbf{W}} \left\|\mathbf{Z}\mathbf{W} - \bar{\mathbf{Z}}\right\|_F^2 \leq (1-\rho)\left\|\mathbf{Z} - \bar{\mathbf{Z}}\right\|_F^2$.

In Assumption 3, we assume that $\rho := 1 - \max\left\{\left|\lambda_2(\mathbf{W})\right|, \left|\lambda_n(\mathbf{W})\right|\right\}^2 > 0$, where let $\lambda_i(\mathbf{W})$ denote the i-th largest eigenvalue of the mixing matrix $\mathbf{W}$ with $-1 \leq \lambda_n(\mathbf{W}) \leq \cdots \leq \lambda_2(\mathbf{W}) \leq \lambda_1(\mathbf{W}) \leq 1$. For example, the value of ρ is commonly used when $\rho = 1$ for the full-mesh (complete) communication topology.

3.4.1. Convergence result of D-SUM

We now state our convergence result for D-SUM (red highlight) in Algorithm 1.

Theorem 1. *Considering problem (1) under the above mentioned assumptions, we denote* $\beta_0 = \max\{1+\beta, 1+\alpha\beta\}$. *The sequence of iterates generated by (7) in Algorithm 1 for learning rate* $\eta \le \min\left\{\frac{\rho}{5}, \frac{(1-\beta)^2}{(1+\alpha^2\beta)}\right\}\frac{1}{L}$, *hyperparameter* $\beta_0 < \frac{2\sqrt{3}}{3}$ *and* $0 \le \alpha \le \frac{1}{1-\beta}$ *ensuring* $\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 \le \varepsilon$. *For all* $T \ge 1$ *and* $K \ge 1$, *the convergence bound* ε *satisfies:*

$$\mathcal{O}\left(\frac{1}{\eta KT} + \frac{\eta(1+\alpha^2)\sigma^2}{n}\right) + \mathcal{O}\left(\frac{\eta^2\zeta^2}{n\rho^2}\right) + \mathcal{O}\left(\frac{\eta^2\sigma^2(1+\alpha^2)}{n\rho}\right).$$

Remark 1. Theorem 1 proposes a non-asymptotic convergence bound of D-SUM for the general non-convex setting. $\mathcal{O}\left(\frac{1}{\eta KT} + \frac{\eta(1+\alpha^2)\sigma^2}{n}\right)$ is generated by the typical fully synchronization SGD, and it matches the convergence bounds of D-SGD with Nesterov's acceleration [27,61]. The second drift term $\mathcal{O}\left(\frac{\eta^2\zeta^2}{n\rho^2}\right)$ would arise since the non-IID data distribution. Intuitively, the convergence rate of D-SUM is non-negligible on α due to its higher order and may converge to an arbitrary suboptimal point. In Section 5, we perform related experiments to confirm this speculation.

3.4.2. Convergence result of GT-DSUM

The next theorem is the convergence result of GT-DSUM in Algorithm 2, and the detailed proof is in Section 4. Based on the GT addressing the issue of how to apply the mini-batch gradient estimates to track the global optimization descent direction, we define the following proposition to clarify this illustration.

Proposition 1 *(Gradients averaging tracker [7]). We assume a loose constraint that the auxiliary variables* $\mathbf{y}_i^{(t)}$ *are considered as the tracker of the average* $\frac{1}{n}\sum_{j=1}^n \nabla f_j(\mathbf{x}_i^{(t)})$, *which means for any epoch t, we have* $\mathbb{E}\left\|\mathbf{y}_i^{(t)} - \frac{1}{n}\sum_{j=1}^n \nabla f_j(\mathbf{x}_i^{(t)})\right\|^2 \le \epsilon^2$.

Theorem 2. *Consider problem (1) under the listed specific assumptions, we denote* $\beta_0 = \max\{1+\alpha\beta, 1+\beta\}$, *and set* $T \ge 1$ *in Algorithm 2 with learning rate* η *chosen as* $\frac{3}{2\beta_0 L} \le \eta \le \min\left\{\frac{1}{2L}, \frac{(1-\beta)^2}{2\beta\alpha^2+(2-\beta)L}\right\}$ *and hyperparameters satisfy* $\max\left\{1 - \frac{1}{8\beta_0^2(1+8L^2)}, 1 - \frac{L^2}{2\beta_0^2(1+9L^2)}\right\} < \rho \le 1$ *ensuring* $\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 \le \varepsilon$. *For all* $T \ge 1$, *the convergence bound* ε *satisfies:*

$$\mathcal{O}\left(\frac{1}{\eta KT} + \frac{\eta(1+\alpha^2)\sigma^2}{n}\right) + \mathcal{O}\left(\frac{\zeta^2+\sigma^2}{n\rho}\right) + \mathcal{O}\left(\frac{\epsilon^2}{n}\right).$$

Remark 2. By comparing Theorem 1 and Theorem 2, we conclude that the convergence rate for both algorithms has the same dependence on η, n, and α. To some extent, GT-DSUM still suffers from the stochastic noise level σ^2, which matches the observation in recent researches [21,32]. Finally, our theorem imposes the constraint on the learning rate η. In Section 5, however, GT-DSUM performs well when this constraint is violated via gird search.

4. Unified convergence analysis

In this section, we present the unified convergence analysis of two algorithms in the UMP framework with non-convex functions. Firstly, we will repeatedly use the following facts and state some assumptions throughout the paper.

- **Fact 1:** For any random vector $\mathbf{a}$, it holds for $\mathbb{E}\|\mathbf{a}\|^2 = \mathbb{E}\|\mathbf{a} - \mathbb{E}[\mathbf{a}]\|^2 + \|\mathbb{E}[\mathbf{a}]\|^2$.
- **Fact 2:** For any $a > 0$, we have $\pm\langle\mathbf{a},\mathbf{b}\rangle \le \frac{1}{2a}\|\mathbf{a}\|^2 + \frac{a}{2}\|\mathbf{b}\|^2$.
- **Fact 3:** $\langle\mathbf{a},\mathbf{b}\rangle = \frac{1}{2}\|\mathbf{a}\|^2 + \frac{1}{2}\|\mathbf{b}\|^2 - \frac{1}{2}\|\mathbf{a}-\mathbf{b}\|^2$.
- **Fact 4:** For given two vectors $\mathbf{a}$ and $\mathbf{b}, \forall a > 0$, we have $\|\mathbf{a}+\mathbf{b}\|^2 \le (1+a)\|\mathbf{a}\|^2 + (1+\frac{1}{a})\|\mathbf{b}\|^2$.
- **Fact 5:** For arbitrary set of n vectors $\{\mathbf{a}_i\}_{i=1}^n$, we have $\left\|\sum_{i=1}^n \mathbf{a}_i\right\|^2 \le n\sum_{i=1}^n \|\mathbf{a}_i\|^2$.
- **Fact 6:** Suppose $\{b_i\}_{i=1}^n$, and $\{\mathbf{a}_i\}_{i=1}^n$ are a set of non-negative scalars and vectors, respectively. We define $s = \sum_{i=1}^n b_i$. Then according to Jensen's inequality, we have $\left\|\sum_{i=1}^n b_i\mathbf{a}_i\right\|^2 = s^2\left\|\sum_{i=1}^n \frac{b_i}{s}\mathbf{a}_i\right\|^2 \le s^2\cdot\sum_{i=1}^n \frac{b_i}{s}\|\mathbf{a}_i\|^2 = s\cdot\sum_{i=1}^n b_i\|\mathbf{a}_i\|^2$.

The inequalities of **Fact 4** also hold for the sum of two matrices $\mathbf{A},\mathbf{B}$ in the Frobenius norm.

Proposition 2. *One step of gossip averaging with the mixing matrix* $\mathbf{W}$ *defined in the Definition 1 preserves the averaging of the iterates,* i.e., $\mathbf{X}\mathbf{W}\frac{\mathbf{1}\mathbf{1}^\top}{n} = \mathbf{X}\frac{\mathbf{1}\mathbf{1}^\top}{n}$.

Proposition 3. *Let $\{V_t\}_{t\geq 0}$ be a non-negative sequence and $C \geq 0$ be some constant such that $\forall t \geq 1, V_{t+1} \leq qV_t + qV_{t-1} + C_2$, where $q \in (0,1)$. Then the following inequality holds if we recursively apply the inequality on V_t from $t+1$ to 1 to obtain: $\forall t \geq 1$,*

$$\begin{aligned} V_{t+1} &\leq qV_t + qV_{t-1} + C \\ &\leq qV_t + V_{t-1} + C \\ &\leq q^2 V_{t-1} + (qV_{t-2} + V_{t-1}) + (q+1)C \\ &\cdots \\ &\leq q^t V_1 + \sum_{i=0}^{t-1} q^{t-1-i} V_i + C \sum_{i=0}^{t-1} q^i. \end{aligned} \tag{9}$$

We sum up (9) over t from 1 to $T-1$ to obtain: $\forall T \geq 2$,

$$\begin{aligned} \sum_{t=0}^{T-1} V_{t+1} &\leq V_1 \sum_{t=0}^{T-1} q^t + \sum_{t=0}^{T-1} \sum_{i=0}^{t-1} q^{t-1-i} V_i + C \sum_{t=0}^{T-1} \sum_{i=0}^{t-1} q^i \\ &\leq V_1 \sum_{t=0}^{\infty} q^t + \sum_{t=0}^{T-1} \left(\sum_{i=0}^{\infty} q^i \right) V_t + C \sum_{t=0}^{T-1} \sum_{i=0}^{\infty} q^i \\ &\leq \frac{V_1}{1-q} + \frac{1}{1-q} \sum_{t=0}^{T-1} V_t + \frac{CT}{1-q}. \end{aligned} \tag{10}$$

Note that our schemes have the following observations on the role of momentum. We now state two basic lemmas.

Lemma 1. *Consider $\bar{\mathbf{g}}^{(t),\tau} = \frac{1}{n} \sum_{i=1}^n \left[\mathbf{g}_i^{(t),\tau} := \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i) \right]$ with the above mentioned assumptions, we have*

$$\mathbb{E} \left\| \bar{\mathbf{g}}^{(t),\tau} \right\|^2 \leq \frac{\sigma^2}{n} + \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2.$$

Proof.

$$\begin{aligned} \mathbb{E} \left\| \bar{\mathbf{g}}^{(t),\tau} \right\|^2 &= \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i) \pm \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2 \\ &\overset{(a)}{=} \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i) - \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2 + \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2 \\ &\leq \frac{1}{n^2} \sum_{i=1}^n \mathbb{E} \left\| \nabla F_i(\mathbf{x}_i^{(t),\tau}, \xi_i) - \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2 + \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2 \\ &\overset{(b)}{\leq} \frac{\sigma^2}{n} + \mathbb{E} \left\| \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau}) \right\|^2, \end{aligned} \tag{11}$$

where (a) follows since the **Fact 1** with $\mathbb{E}_{\xi_i \sim D_i} \left[\nabla F_i(\mathbf{x}, \xi_i) \right] = \nabla f_i(\mathbf{x})$; while (b) is followed because of the Assumption 2. □

Lemma 2. *Let introduce an auxiliary variable $\mathbf{y}_i^{(t),\tau} = \frac{\beta}{1-\beta} \left(\mathbf{x}_i^{(t),\tau} - \mathbf{v}_i^{(t),\tau} \right)$, we define $\mathbf{z}_i^{(t),\tau} \triangleq \mathbf{x}_i^{(t),\tau} + \mathbf{y}_i^{(t),\tau}$ and $\mathbf{c}_i^{(t),\tau} \triangleq \frac{1-\beta}{\beta} \mathbf{y}_i^{(t),\tau}$. Then we have*

$$\mathbf{z}_i^{(t),\tau+1} = \mathbf{z}_i^{(t),\tau} - \frac{\eta}{1-\beta} \mathbf{g}_i^{(t),\tau} \tag{12}$$

and

$$\mathbf{c}_i^{(t),\tau+1} = \beta \mathbf{c}_i^{(t),\tau} + (\alpha - \alpha\beta - 1)\eta \mathbf{g}_i^{(t),\tau}. \tag{13}$$

Proof. For (12), starting from the definition of $\mathbf{v}_i^{(t),\tau}$ in (7),

$$\mathbf{z}_i^{(t),\tau+1} = \frac{1}{1-\beta} \mathbf{x}_i^{(t),\tau+1} - \frac{\beta}{1-\beta} \left(\mathbf{x}_i^{(t),\tau} - \alpha\eta \mathbf{g}_i^{(t),\tau} \right)$$

$$= \mathbf{x}_i^{(t),\tau} + \frac{\beta}{1-\beta}\left(\mathbf{x}_i^{(t),\tau} - \mathbf{x}_i^{(t),\tau-1} + \alpha\eta\mathbf{g}_i^{(t),\tau-1}\right) - \frac{\eta}{1-\beta}\mathbf{g}_i^{(t),\tau}$$
$$= \mathbf{x}_i^{(t),\tau} + \mathbf{y}_i^{(t),\tau} - \frac{\eta}{1-\beta}\mathbf{g}_i^{(t),\tau}.$$

Similarly for (13),

$$\mathbf{c}_i^{(t),\tau+1} = \mathbf{x}^{(t),\tau} - \eta\mathbf{g}_i^{(t),\tau} - \mathbf{x}_i^{(t),\tau} + \alpha\eta\mathbf{g}_i^{(t),\tau} + \beta\left(\mathbf{x}^{(t),\tau} - \alpha\eta\mathbf{g}_i^{(t),\tau} - \mathbf{x}^{(t),\tau-1} + \alpha\eta\mathbf{g}_i^{(t),\tau-1}\right)$$
$$= \beta\left(\mathbf{x}_i^{(t),\tau} - \mathbf{v}_i^{(t),\tau}\right) + (\alpha - \alpha\beta - 1)\eta\mathbf{g}_i^{(t),\tau}. \quad \square$$

Lemma 3. *Due to Proposition 2, for any $i \in \{1, \cdots, n\}$, the model parameters $\mathbf{x}_i^{(t)}$ and gradient tracking variable $\mathbf{y}_i^{(t)}$ will asymptotically agree [7], which means* $\lim_{t\to\infty}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 = 0$, $\lim_{t\to\infty}\left\|\mathbf{y}_i^{(t)} - \bar{\mathbf{y}}^{(t)}\right\|^2 = 0$.

4.1. Proof of Theorem 1

Since the smoothness of function f, it follows that

$$f(\bar{\mathbf{z}}^{(t),\tau+1}) \le f(\bar{\mathbf{z}}^{(t),\tau}) + \left\langle \nabla f(\bar{\mathbf{z}}^{(t),\tau}), \bar{\mathbf{z}}^{(t),\tau+1} - \bar{\mathbf{z}}^{(t),\tau}\right\rangle + \frac{L}{2}\left\|\bar{\mathbf{z}}^{(t),\tau+1} - \bar{\mathbf{z}}^{(t),\tau}\right\|^2. \tag{14}$$

According to the described factors, the second term on the right-hand side for (14):

$$\begin{aligned}
&\left\langle \nabla f(\bar{\mathbf{z}}^{(t),\tau}), \bar{\mathbf{z}}^{(t),\tau+1} - \bar{\mathbf{z}}^{(t),\tau}\right\rangle \\
&\overset{(a)}{=} -\frac{\eta}{1-\beta}\left\langle \nabla f(\bar{\mathbf{z}}^{(t),\tau}), \bar{\mathbf{g}}^{(t),\tau}\right\rangle \\
&\overset{(b)}{=} -\frac{\eta}{1-\beta}\left\langle \nabla f(\bar{\mathbf{z}}^{(t),\tau}), \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\rangle \\
&= \underbrace{-\frac{\eta}{1-\beta}\left\langle \nabla f(\bar{\mathbf{z}}^{(t),\tau}) - \nabla f(\bar{\mathbf{x}}^{(t),\tau}), \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\rangle}_{E_1} \underbrace{-\frac{\eta}{1-\beta}\left\langle \nabla f(\bar{\mathbf{x}}^{(t),\tau}), \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\rangle}_{E_2},
\end{aligned} \tag{15}$$

where (a) follows from the averaged form of Lemma 2; (b) follows because the stochastic gradient $\mathbf{x}_i^{(t),\tau}$ is determined by the variables ξ_i but is dependent of it. For E_1, we note that:

$$\begin{aligned}
E_1 &\overset{(a)}{\le} \frac{1-\beta}{2\beta L}\left\|\nabla f(\bar{\mathbf{z}}^{(t),\tau}) - \nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 + \frac{\beta L\eta^2}{2(1-\beta)^3}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 \\
&\overset{(b)}{\le} \frac{(1-\beta)L}{2\beta}\left\|\bar{\mathbf{z}}^{(t),\tau} - \bar{\mathbf{x}}^{(t),\tau}\right\|^2 + \frac{\beta L\eta^2}{2(1-\beta)^3}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2,
\end{aligned} \tag{16}$$

(a) follows by applying the basic inequality of **Fact 2** where a=1, $\mathbf{a}$=$-\frac{\sqrt{1-\beta}}{\sqrt{\beta L}}\left(\nabla f(\bar{\mathbf{z}}^{(t),\tau}) - \nabla f(\bar{\mathbf{x}}^{(t),\tau})\right)$ and $\mathbf{b}$=$\frac{\eta\sqrt{\beta L}}{(1-\beta)^{3/2}}\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})$; and (b) follows by the smoothness of Assumption 1. Then, for E_2:

$$\begin{aligned}
E_2 &\overset{(a)}{=} \frac{1}{2}\left(\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 + \left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 - \left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau}) - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2\right) \\
&= \frac{1}{2}\left(\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 + \left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 - \left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\mathbf{x}}^{(t),\tau}) - \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2\right) \\
&\overset{(b)}{\ge} \frac{1}{2}\left(\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 + \left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 - \frac{L^2}{n}\sum_{i=1}^n \left\|\bar{\mathbf{x}}^{(t),\tau} - \mathbf{x}_i^{(t),\tau}\right\|^2\right),
\end{aligned} \tag{17}$$

where (a) follows by applying the **Fact 3** with $\mathbf{a} = \nabla f(\bar{\mathbf{x}}^{(t),\tau})$ and $\mathbf{b} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t),\tau})$; (b) follows because of the Assumption 1 and the Jensen's inequality with the convexity of $\|\cdot\|^2$. By Lemma 2, we can directly derive the following inequality from the third term on the right-hand side of (14) yielding

$$\frac{L}{2}\mathbb{E}_{t,\tau}\left\|\bar{\mathbf{z}}^{(t),\tau+1} - \bar{\mathbf{z}}^{(t),\tau}\right\|^2 \le \frac{L\eta^2}{2(1-\beta)^2}\mathbb{E}_{t,\tau}\left\|\bar{\mathbf{g}}^{(t),\tau}\right\|^2. \tag{18}$$

Using that $\sum_{i=1}^n \left\|\mathbf{a}_i\right\|^2 = \|\mathbf{A}\|_F^2$ where $\mathbf{A} := \left[\mathbf{a}_1, \cdots, \mathbf{a}_n\right]$. Substituting (16), (17) and (18) to (14), which yields

$$f(\bar{\mathbf{z}}^{(t),\tau+1}) \le f(\bar{\mathbf{z}}^{(t),\tau}) + \frac{(1-\beta)L}{2\beta}\left\|\bar{\mathbf{z}}^{(t),\tau} - \bar{\mathbf{x}}^{(t),\tau}\right\|^2 + \left(\frac{\beta L\eta^2}{2(1-\beta)^3} - \frac{\eta}{2(1-\beta)}\right)\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 - \frac{\eta}{2(1-\beta)}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 + \frac{\eta L^2}{2n(1-\beta)}\left\|\bar{\mathbf{X}}^{(t),\tau} - \mathbf{X}^{(t),\tau}\right\|_F^2 + \frac{\eta^2 L}{2(1-\beta)^2}\left\|\bar{\mathbf{g}}^{(t),\tau}\right\|^2. \tag{19}$$

Dividing both sides by $\frac{\eta}{2(1-\beta)}$ and rearranging the terms, we have

$$\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 \le \frac{2(1-\beta)}{\eta}\left(f(\bar{\mathbf{z}}^{(t),\tau}) - f(\bar{\mathbf{z}}^{(t),\tau+1})\right) + \frac{(1-\beta)^2 L}{\beta\eta}\left\|\bar{\mathbf{z}}^{(t),\tau} - \bar{\mathbf{x}}^{(t),\tau}\right\|^2 - \left(1 - \frac{\beta L\eta}{(1-\beta)^2}\right)\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 + \frac{L^2}{n}\left\|\bar{\mathbf{X}}^{(t),\tau} - \mathbf{X}^{(t),\tau}\right\|_F^2 + \frac{\eta L}{1-\beta}\left\|\bar{\mathbf{g}}^{(t),\tau}\right\|^2. \tag{20}$$

Based on Algorithm 1, we apply multiple local updates with a fixed K. From (8) and Proposition 2, we can obtain $\bar{\mathbf{z}}^{(t),K} = \bar{\mathbf{z}}^{(t+1),0}$. Summing from $\tau = 0$ to $K-1$, $t = 0$ to $T-1$ and taking expected values on both sides, we have

$$\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 \le \frac{2(1-\beta)}{\eta}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0}) - \mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right) + \frac{(1-\beta)^2 L}{\beta\eta}\underbrace{\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\bar{\mathbf{z}}^{(t),\tau} - \bar{\mathbf{x}}^{(t),\tau}\right\|^2}_{E_3} - \left(1 - \frac{\beta L\eta}{(1-\beta)^2}\right)\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 + \frac{L^2}{n}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\bar{\mathbf{X}}^{(t),\tau} - \mathbf{X}^{(t),\tau}\right\|_F^2 + \frac{\eta L}{1-\beta}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\bar{\mathbf{g}}^{(t),\tau}\right\|^2. \tag{21}$$

For E_3, we omit the used time slot including epoch (*i.e.*, t) and local updates (*i.e.*, τ) and replace them by a more general term: **iteration** (*i.e.*, l). Begin with (13) in Lemma 2, we have

$$\begin{aligned}
\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\bar{\mathbf{z}}^{(t),\tau} - \bar{\mathbf{x}}^{(t),\tau}\right\|^2 &= \sum_{l=0}^{KT-1}\left\|\bar{\mathbf{z}}^{l} - \bar{\mathbf{x}}^{l}\right\|^2 \\
&= \frac{\beta^2}{(1-\beta)^2}\sum_{l=0}^{KT-1}\left\|\bar{\mathbf{c}}_i^{l}\right\|^2 \\
&\overset{(a)}{=} \frac{\beta^2\hat{\eta}^2 s^2}{(1-\beta)^2}\sum_{l=1}^{KT-1}\left\|\sum_{j=0}^{l-1}\frac{\beta^{l-1-j}}{s}\bar{\mathbf{g}}^{j}\right\|^2 \\
&\overset{(b)}{\le} \frac{\beta^2\hat{\eta}^2 s^2}{(1-\beta)^2}\sum_{l=1}^{KT-1}\sum_{j=0}^{l-1}\frac{\beta^{l-1-j}}{s}\left\|\bar{\mathbf{g}}^{j}\right\|^2 \\
&\overset{(c)}{\le} \frac{\beta^2\hat{\eta}^2}{(1-\beta)^3}\sum_{l=1}^{KT-1}\sum_{j=0}^{l-1}\beta^{l-1-j}\left\|\bar{\mathbf{g}}^{j}\right\|^2 \\
&= \frac{\beta^2\hat{\eta}^2}{(1-\beta)^3}\sum_{j=0}^{KT-2}\left(\left\|\bar{\mathbf{g}}^{j}\right\|^2\sum_{l=j+1}^{KT-1}\beta^{l-1-j}\right) \\
&\le \frac{\beta^2\hat{\eta}^2}{(1-\beta)^4}\sum_{l=0}^{KT-1}\left\|\bar{\mathbf{g}}^{l}\right\|^2 \\
&\le \frac{\alpha^2\beta^2\eta^2}{(1-\beta)^4}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\bar{\mathbf{g}}^{(t),\tau}\right\|^2,
\end{aligned} \tag{22}$$

where we note that $\bar{\mathbf{z}}^{l} - \bar{\mathbf{x}}^{l} = \mathbf{0}$ when $l = 0$. We further recursively apply (13) and **Fact 6** where $s = \sum_{j=0}^{l-1}\beta^{l-1-j}$ in (a); (b) follows because of the convexity of $\|\cdot\|^2$ and Jensen's inequality; (c) follows because $s = \frac{1-\beta^l}{1-\beta} < \frac{1}{1-\beta}$. Substituting (22) into (21) and applying Lemma 1, yielding

$$\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2 \le \frac{2(1-\beta)}{\eta}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)-\left(1-\frac{(1+\alpha^2\beta)L\eta}{(1-\beta)^2}\right)\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2 + L^2\underbrace{\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\frac{1}{n}\mathbb{E}\left\|\bar{\mathbf{X}}^{(t),\tau}-\mathbf{X}^{(t),\tau}\right\|_F^2}_{E_4}+\frac{(\alpha^2\beta+1-\beta)\eta L\sigma^2}{n(1-\beta)^2}KT, \tag{23}$$

where the above inequality is held by setting $0\le\alpha\le\frac{1}{1-\beta}$. We now try to bound the consensus error E_4 between the nodes' parameters and its averaging. We first reiterate the update scheme of (7) in a matrix form regardless of epoch t and local update τ and denote l as the index of updating iteration:

$$\mathbf{X}^{l+1}=\mathbf{X}^l-\eta\mathbf{G}^l+\beta\left(\mathbf{X}^l-\alpha\eta\mathbf{G}^l-\mathbf{X}^{l-1}+\alpha\eta\mathbf{G}^{l-1}\right). \tag{24}$$

For the averaged parameters which are performed model averaging across all nodes, we can also simply its update rule since $\mathbf{W}$ is doubly stochastic, which is described as follows:

$$\bar{\mathbf{X}}^{l+1}=\bar{\mathbf{X}}^l-\eta\bar{\mathbf{G}}^l+\beta\left(\bar{\mathbf{X}}^l-\alpha\eta\bar{\mathbf{G}}^l-\bar{\mathbf{X}}^{l-1}+\alpha\eta\bar{\mathbf{G}}^{l-1}\right). \tag{25}$$

According to the above two equations, we begin with Assumption 3 yielding

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{l+1}-\bar{\mathbf{X}}^{l+1}\right\|_F^2\\
&\le\frac{1-\rho}{n}\mathbb{E}\left\|\mathbf{X}^l-\eta\mathbf{G}^l+\beta\left(\mathbf{X}^l-\alpha\eta\mathbf{G}^l-\mathbf{X}^{l-1}+\alpha\eta\mathbf{G}^{l-1}\right)-\bar{\mathbf{X}}^l+\eta\bar{\mathbf{G}}^l-\beta\left(\bar{\mathbf{X}}^l-\alpha\eta\bar{\mathbf{G}}^l-\bar{\mathbf{X}}^{l-1}+\alpha\eta\bar{\mathbf{G}}^{l-1}\right)\right\|_F^2\\
&\overset{(a)}{\le}\frac{1-\rho}{n}\mathbb{E}\Big\|\mathbf{X}^l-\eta\mathbb{E}_l\left[\mathbf{G}^l\right]+\beta\left(\mathbf{X}^l-\alpha\eta\mathbb{E}_l\left[\mathbf{G}^l\right]-\mathbf{X}^{l-1}+\alpha\eta\mathbb{E}_l\left[\mathbf{G}^{l-1}\right]\right)\\
&\qquad-\bar{\mathbf{X}}^l+\eta\mathbb{E}_l\left[\bar{\mathbf{G}}^l\right]-\beta\left(\bar{\mathbf{X}}^l-\alpha\eta\mathbb{E}_l\left[\bar{\mathbf{G}}^l\right]-\bar{\mathbf{X}}^{l-1}+\alpha\eta\mathbb{E}_l\left[\bar{\mathbf{G}}^{l-1}\right]\right)\Big\|_F^2+\underbrace{\frac{2(1-\rho)\eta^2\sigma^2}{n}\left(1+2\alpha^2\beta^2\right)}_{:=\Delta}\\
&=\frac{1-\rho}{n}\mathbb{E}\Big\|(1+\beta)\left(\mathbf{X}^l-\bar{\mathbf{X}}^l\right)-\beta\left(\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right)-(1+\alpha\beta)\eta\left(\mathbb{E}_l\left[\mathbf{G}^l\right]-\mathbb{E}_l\left[\bar{\mathbf{G}}^l\right]\right)\\
&\qquad+\alpha\beta\eta\left(\mathbb{E}_l\left[\mathbf{G}^{l-1}\right]-\mathbb{E}_l\left[\bar{\mathbf{G}}^{l-1}\right]\right)\Big\|_F^2+\Delta\\
&\overset{(b)}{\le}\frac{(1-\rho)(1+\beta)^2(1+\frac{\rho}{2})}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{(1-\rho)\beta^2(1+\frac{\rho}{2})}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2\\
&\qquad+\frac{3\alpha^2\beta^2\eta^2}{n\rho}\mathbb{E}\left\|\mathbb{E}_l\left[\mathbf{G}^{l-1}\right]-\mathbb{E}_l\left[\bar{\mathbf{G}}^{l-1}\right]\right\|_F^2+\frac{3\eta^2(1+\alpha\beta)^2}{n\rho}\mathbb{E}\left\|\mathbb{E}_l\left[\mathbf{G}^l\right]-\mathbb{E}_l\left[\bar{\mathbf{G}}^l\right]\right\|_F^2+\Delta,
\end{aligned} \tag{26}$$

where (a) follows because we add the expectation term of $\mathbf{G}$ and $\bar{\mathbf{G}}$ making sure that $\mathbf{G}-\bar{\mathbf{G}}=(\mathbf{G}-\mathbb{E}[\mathbf{G}])-\left(\bar{\mathbf{G}}-\mathbb{E}\left[\bar{\mathbf{G}}\right]\right)+\left(\mathbb{E}[\mathbf{G}]-\mathbb{E}\left[\bar{\mathbf{G}}\right]\right)$, which satisfies the condition of Assumption 2 generalizing the constant Δ; (b) follows from the **Fact 1** and **Fact 4** by setting $a=\rho/2$. We can further proceed as

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{l+1}-\bar{\mathbf{X}}^{l+1}\right\|_F^2\\
&\overset{(a)}{\le}\frac{\left(1-\frac{\rho}{2}\right)(1+\beta)^2}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{\left(1-\frac{\rho}{2}\right)\beta^2}{n}\mathbb{E}\left\|\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right\|_F^2\\
&\qquad+\frac{6\alpha^2\beta^2\eta^2L^2}{n\rho}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{6(1+\alpha\beta)^2\eta^2L^2}{n\rho}\mathbb{E}\left\|\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right\|_F^2+\frac{12(1+\alpha\beta)^2\eta^2\zeta^2}{\rho}+\Delta\\
&\overset{(b)}{\le}\frac{\left(1-\frac{\rho}{2}\right)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{\left(1-\frac{\rho}{2}\right)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right\|_F^2\\
&\qquad+\frac{6\beta_0^2\eta^2L^2}{n\rho}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{6\beta_0^2\eta^2L^2}{n\rho}\mathbb{E}\left\|\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right\|_F^2+\frac{12\beta_0^2\eta^2\zeta^2}{\rho}+\Delta\\
&\overset{(c)}{\le}\frac{\left(1-\frac{\rho}{4}\right)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2+\frac{\left(1-\frac{\rho}{4}\right)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^{l-1}-\bar{\mathbf{X}}^{l-1}\right\|_F^2+\frac{12\beta_0^2\eta^2\zeta^2}{\rho}+\Delta
\end{aligned} \tag{27}$$

where (a) follows because $\mathbb{E}\left\|\mathbb{E}_l\left[\mathbf{G}^l\right]-\mathbb{E}_l\left[\bar{\mathbf{G}}^l\right]\right\|_F^2\le\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^l)\pm\nabla f_i(\bar{\mathbf{x}}^l)-\nabla f(\bar{\mathbf{x}}_i^l)\right\|^2\le 2L^2\sum_{i=1}^n\mathbb{E}\left\|\mathbf{x}_i^l-\bar{\mathbf{x}}^l\right\|^2+2n\zeta^2$; we denote $\beta_0=\max\{1+\beta,1+\alpha\beta\}$ in (b); (c) follows because the assumption that the learning rate $\eta\le\frac{\rho}{5L}$ ensures that $6\eta^2L^2\le\rho^2/4$. We denote

$(1-\rho/4)\,\beta_0^2$ as Q_1 ensuring that $Q_1<1$, so that we can apply Proposition 3. Since $\mathbf{x}_i^{(0)}=\bar{\mathbf{x}}^{(0)}=\mathbf{x}_0$ and assuming that $\mathbf{x}_i^{(-1)}=\bar{\mathbf{x}}^{(-1)}=\mathbf{0}$, we get $\frac{1}{n}\mathbb{E}\|\mathbf{X}^1-\bar{\mathbf{X}}^1\|_F^2=12\beta_0^2\eta^2\zeta^2/\rho+\Delta$. We denote $\Phi=\max_{0\le l\le T-2}\left\{\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2\right\}$, then for E_4:

$$\begin{aligned}E_4&=\sum_{l=0}^{KT-1}\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^l-\bar{\mathbf{X}}^l\right\|_F^2\\&\le KT\left(\frac{12\beta_0^2\eta^2\zeta^2}{n\rho(1-Q_1)}+\frac{2\eta^2\sigma^2\left(1+2\alpha^2\beta^2\right)(1-\rho)}{n^2\left(1-Q_1\right)}\right)+\frac{KT\Phi}{1-Q_1}.\end{aligned}\tag{28}$$

Let the learning rate $\eta\le\frac{(1-\beta)^2}{(1+\alpha^2\beta)L}$ to ensure that $1-\frac{(1+\alpha^2\beta)L\eta}{(1-\beta)^2}>0$. Substituting (28) into (23), then dividing both sides by KT, yielding

$$\begin{aligned}\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2&\le\frac{2(1-\beta)}{\eta KT}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)+\frac{(\alpha^2\beta+1-\beta)\eta L\sigma^2}{n(1-\beta)^2}+\frac{\Phi L^2}{1-Q_1}\\&\quad+L^2\left(\frac{12\beta_0^2\eta^2\zeta^2}{n\rho(1-Q_1)}+\frac{2\eta^2\sigma^2\left(1+2\alpha^2\beta^2\right)(1-\rho)}{n\left(1-Q_1\right)}\right).\end{aligned}\tag{29}$$

Substituting $Q_1=(1-\rho/4)\,\beta_0^2$, yielding

$$\begin{aligned}\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2&\le\frac{2(1-\beta)}{\eta KT}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)+\frac{(\alpha^2\beta+1-\beta)\eta L\sigma^2}{n(1-\beta)^2}+\frac{\Phi L^2}{1-\beta_0^2+\frac{\rho}{4}}\\&\quad+L^2\left(\frac{12\beta_0^2\eta^2\zeta^2}{n\rho(1-\beta_0^2+\frac{\rho}{4})}+\frac{2\eta^2\sigma^2\left(1+2\alpha^2\beta^2\right)(1-\rho)}{n\left(1-\beta_0^2+\frac{\rho}{4}\right)}\right).\end{aligned}\tag{30}$$

For the first two terms of (30), since β, $\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})$ and L are constants, these two terms can be reduced to $\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta(1+\alpha^2)\sigma^2}{n}\right)$. Due to Lemma 3, Φ can be regarded as 0 when T is large enough. Thus, the third term can be omitted. Omitting constants L, β_0^2 and β^2, the last term can be reduced to $\mathcal{O}\left(\frac{\eta^2\zeta^2}{n\rho^2}\right)+\mathcal{O}\left(\frac{\eta^2\sigma^2(1+\alpha^2)}{n\rho}\right)$. Finally, we obtain the result of Theorem 1:

$$\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2\le\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta(1+\alpha^2)\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\eta^2\zeta^2}{n\rho^2}\right)+\mathcal{O}\left(\frac{\eta^2\sigma^2(1+\alpha^2)}{n\rho}\right).\tag{31}$$

4.2. Proof of Theorem 2

We now state our convergence results for `GT-DSUM` in Algorithm 2. Similar to proof process of Theorem 1, begin with the smoothness of f,

$$\begin{aligned}\mathbb{E}_{t,\tau}f(\bar{\mathbf{z}}^{(t),\tau+1})&\le\mathbb{E}_{t,\tau}f(\bar{\mathbf{z}}^{(t),\tau})+\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{z}}^{(t),\tau}),\bar{\mathbf{z}}^{(t),\tau+1}-\bar{\mathbf{z}}^{(t),\tau}\right\rangle+\frac{L}{2}\mathbb{E}_{t,\tau}\left\|\bar{\mathbf{z}}^{(t),\tau+1}-\bar{\mathbf{z}}^{(t),\tau}\right\|^2\\&\overset{(a)}{=}\mathbb{E}_{t,\tau}f(\bar{\mathbf{z}}^{(t),\tau})\underbrace{-\frac{\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{z}}^{(t),\tau}),\tilde{\mathbf{m}}^{(t),\tau}\right\rangle}_{E_1}+\underbrace{\frac{L\eta^2}{2(1-\beta)^2}\mathbb{E}_{t,\tau}\left\|\tilde{\mathbf{m}}^{(t),\tau}\right\|^2}_{E_2},\end{aligned}\tag{32}$$

where (a) follows because we use the form of (12) with the normalized updating gradient direction $\mathbf{m}_i^{(t),\tau}$. According to the definition of $\mathbf{m}^{(t),\tau}$ (Line 4) in Algorithm 2, we have the following derivation for E_1:

$$\begin{aligned}E_1&=-\frac{\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{z}}^{(t),\tau})-\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\lambda\bar{\mathbf{g}}^{(t),\tau}+(1-\lambda)\bar{\mathbf{y}}^{(t)}\right\rangle-\frac{\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\lambda\bar{\mathbf{g}}^{(t),\tau}+(1-\lambda)\bar{\mathbf{y}}^{(t)}\right\rangle\\&=-\frac{\lambda\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{z}}^{(t),\tau})-\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\bar{\mathbf{g}}^{(t),\tau}\right\rangle-\frac{(1-\lambda)\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{z}}^{(t),\tau})-\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\bar{\mathbf{y}}^{(t)}\right\rangle\\&\quad-\frac{\lambda\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\bar{\mathbf{g}}^{(t),\tau}\right\rangle-\frac{(1-\lambda)\eta}{1-\beta}\mathbb{E}_{t,\tau}\left\langle\nabla f(\bar{\mathbf{x}}^{(t),\tau}),\bar{\mathbf{y}}^{(t)}\right\rangle\\&\overset{(a)}{\le}\frac{(1-\beta)L}{2\beta}\left\|\bar{\mathbf{z}}^{(t),\tau}-\bar{\mathbf{x}}^{(t),\tau}\right\|^2+\frac{\beta L\lambda\eta^2}{2(1-\beta)^3}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2+\frac{\beta L(1-\lambda)\eta^2}{2(1-\beta)^3}\left\|\bar{\mathbf{y}}^{(t)}\right\|^2\\&\quad-\frac{\lambda\eta}{2(1-\beta)}\left(\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2+\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2-\frac{L^2}{n}\sum_{i=1}^{n}\left\|\bar{\mathbf{x}}^{(t),\tau}-\mathbf{x}^{(t),\tau}\right\|^2\right)\end{aligned}\tag{33}$$

$$
\begin{aligned}
&-\frac{(1-\lambda)\eta}{2(1-\beta)}\left(\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2+\left\|\bar{\mathbf{y}}^{(t)}\right\|^2-\frac{2L^2}{n^2}\sum_{i=1}^{n}\left\|\bar{\mathbf{x}}^{(t),\tau}-\mathbf{x}^{(t)}\right\|^2-\frac{2\epsilon^2}{n}\right)\\
\leq&-\frac{\eta}{2(1-\beta)}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2+\frac{(1-\beta)L}{2\beta}\left\|\bar{\mathbf{z}}^{(t),\tau}-\bar{\mathbf{x}}^{(t),\tau}\right\|^2+\frac{(2-\lambda)\eta L^2}{2n(1-\beta)}\sum_{i=1}^{n}\left\|\bar{\mathbf{x}}^{(t),\tau}-\mathbf{x}^{(t),\tau}\right\|^2+\frac{\epsilon^2\eta(1-\lambda)}{n(1-\beta)}\\
&+\left(\frac{\beta L\lambda\eta^2}{2(1-\beta)^3}-\frac{\lambda\eta}{2(1-\beta)}\right)\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2+\left(\frac{\beta L(1-\lambda)\eta^2}{2(1-\beta)^3}-\frac{(1-\lambda)\eta}{2(1-\beta)}\right)\left\|\bar{\mathbf{y}}^{(t)}\right\|^2,
\end{aligned}
$$

where (a) follows from **Fact 2**, **Fact 3** and Proposition 1, which is similar to the derivation process in (16). Based on Lemma 1, we get the following inequality for E_2:

$$
E_2\leq\frac{L\eta^2\lambda^2\sigma^2}{n(1-\beta)^2}+\frac{L\lambda^2\eta^2}{(1-\beta)^2}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2+\frac{L\eta^2(1-\lambda)^2}{(1-\beta)^2}\left\|\bar{\mathbf{y}}^{(t)}\right\|^2. \tag{34}
$$

Substituting (33) and (34) into (32), then rearranging the terms and dividing both sides by $\frac{\eta}{2(1-\beta)}$, gives

$$
\begin{aligned}
\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2\leq&\frac{2(1-\beta)}{\eta}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(t),\tau})-\mathbb{E}f(\bar{\mathbf{z}}^{(t),\tau+1})\right)+\frac{(1-\beta)^2L}{\beta\eta}\left\|\bar{\mathbf{z}}^{(t),\tau}-\bar{\mathbf{x}}^{(t),\tau}\right\|^2+\frac{(2-\lambda)L^2}{n}\sum_{i=1}^{n}\left\|\bar{\mathbf{x}}^{(t),\tau}-\mathbf{x}_i^{(t),\tau}\right\|^2\\
&+\frac{2\epsilon^2(1-\lambda)}{n}+\frac{2L\eta\lambda^2\sigma^2}{n(1-\beta)}+\left(\frac{2L\lambda\eta}{1-\beta}+\frac{\beta L\lambda\eta}{(1-\beta)^2}-\lambda\right)\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2\\
&+\left(\frac{2L\eta(1-\lambda)}{1-\beta}+\frac{\beta L(1-\lambda)\eta}{(1-\beta)^2}-(1-\lambda)\right)\left\|\bar{\mathbf{y}}^{(t)}\right\|^2.
\end{aligned} \tag{35}
$$

Based on Algorithm 2, we apply multiple local updates with a fixed K. Summing over $\tau\in\{0,1,\dots,K-1\}$, $t\in\{0,1,\dots,T-1\}$, and dividing both sides by KT yields

$$
\begin{aligned}
\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=1}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2\leq&\frac{2(1-\beta)}{\eta KT}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)+\frac{(1-\beta)^2L}{\beta\eta}\frac{1}{KT}\underbrace{\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\bar{\mathbf{z}}^{(t),\tau}-\bar{\mathbf{x}}^{(t),\tau}\right\|^2}_{E_3}\\
&+\frac{(2-\lambda)L^2}{nKT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}_{t,\tau}\left\|\bar{\mathbf{X}}^{(t),\tau}-\mathbf{X}^{(t),\tau}\right\|_F^2+\frac{2\epsilon^2(1-\lambda)}{n}+\frac{2L\eta\lambda^2\sigma^2}{n(1-\beta)}\\
&+\left(\frac{2L\lambda\eta}{1-\beta}+\frac{\beta L\lambda\eta}{(1-\beta)^2}-\lambda\right)\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2\\
&+\left(\frac{2L\eta(1-\lambda)}{1-\beta}+\frac{\beta L(1-\lambda)\eta}{(1-\beta)^2}-(1-\lambda)\right)\frac{1}{T}\sum_{t=0}^{T-1}\left\|\bar{\mathbf{y}}^{(t)}\right\|^2.
\end{aligned} \tag{36}
$$

Recalling the derivation process of (22) and Lemma 1, we have the similar upper bound for E_3

$$
E_3\leq\frac{2\alpha^2\eta^2\beta^2\lambda\sigma^2}{n(1-\beta)^4}+\frac{2\alpha^2\eta^2\beta^2\lambda}{(1-\beta)^4}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2+\frac{2\alpha^2\eta^2\beta^2(1-\lambda)}{(1-\beta)^4}\sum_{t=0}^{T-1}\left\|\bar{\mathbf{y}}^{(t)}\right\|^2. \tag{37}
$$

Substituting the above inequality to (36). Besides, our assumption that the learning rate $\eta\leq\frac{(1-\beta)^2}{2\beta\alpha^2+(2-\beta)L}$ ensures that E_5 and E_6 are smaller than 0. We can proceed as

$$
\begin{aligned}
\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2\leq&\frac{2(1-\beta)}{\eta KT}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)+\underbrace{\frac{(2-\lambda)L^2}{KT}\frac{1}{n}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}_{t,\tau}\left\|\bar{\mathbf{X}}^{(t),\tau}-\mathbf{X}^{(t),\tau}\right\|_F^2}_{E_4}+\frac{2\epsilon^2(1-\lambda)}{n}\\
&+\underbrace{\left(\frac{2\beta\lambda\eta\alpha^2}{(1-\beta)^2}+\frac{2L\lambda\eta}{1-\beta}+\frac{\beta L\lambda\eta}{(1-\beta)^2}-\lambda\right)}_{E_5}\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t),\tau})\right\|^2+\frac{2L\eta\lambda^2\sigma^2}{n(1-\beta)}\\
&+\underbrace{\left(\frac{2\eta\beta(1-\lambda)\alpha^2}{(1-\beta)^2}+\frac{2L\eta(1-\lambda)}{1-\beta}+\frac{\beta L(1-\lambda)\eta}{(1-\beta)^2}-(1-\lambda)\right)}_{E_6}\frac{1}{T}\sum_{t=0}^{T-1}\left\|\bar{\mathbf{y}}^{(t)}\right\|^2+\frac{2\beta\eta\lambda\alpha^2\sigma^2}{n(1-\beta)^2}.
\end{aligned} \tag{38}
$$

We will now try to bound E_4 (*i.e.*, the error between model parameters on each worker and its average across all workers). Starting from the pseudo-code of GT-DSUM, Proposition 2 and proceeding similar to the previous derivation in Theorem 1, we have

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\bar{\mathbf{M}}^{(l)}-\mathbf{M}^{(l)}\right\|_F^2 &= \frac{1}{n}\mathbb{E}\left\|\lambda\left(\bar{\mathbf{G}}^{(l)}-\mathbf{G}^{(l)}\right)+(1-\lambda)\left(\bar{\mathbf{Y}}^{(t)}-\mathbf{Y}^{(t)}\right)\right\|_F^2\\ &\overset{(a)}{=} \frac{1}{n}\mathbb{E}\left\|\lambda\mathbb{E}_l\left[\bar{\mathbf{G}}^{(l)}-\mathbf{G}^{(l)}\right]+(1-\lambda)\left(\bar{\mathbf{Y}}^{(t)}-\mathbf{Y}^{(t)}\right)\right\|_F^2+\frac{\lambda^2}{n}\mathbb{E}\left\|\bar{\mathbf{G}}^{(l)}-\mathbb{E}\left[\bar{\mathbf{G}}^{(l)}\right]-\left(\mathbf{G}^{(l)}-\mathbb{E}\left[\mathbf{G}^{(l)}\right]\right)\right\|_F^2\\ &\overset{(b)}{\le} \frac{2\lambda^2}{n}\mathbb{E}\left\|\mathbb{E}_l\left[\mathbf{G}^{(l)}\right]\pm\nabla f\left(\bar{\mathbf{X}}^{(l)}\right)-\bar{\mathbf{G}}^{(l)}\right\|_F^2+\frac{2(1-\lambda)^2}{n}\mathbb{E}\left\|\bar{\mathbf{Y}}^{(t)}-\mathbf{Y}^{(t)}\right\|_F^2+4\lambda^2\sigma^2\\ &\overset{(c)}{\le} 8\zeta^2+\frac{4L^2}{n}\mathbb{E}\left\|\bar{\mathbf{X}}^{(l)}-\mathbf{X}^{(l)}\right\|_F^2+\frac{2}{n}\mathbb{E}\left\|\bar{\mathbf{Y}}^{(t)}-\mathbf{Y}^{(t)}\right\|_F^2+4\sigma^2,\end{aligned} \tag{39}$$

where (a) follows from **Fact 1**; (b) follows because of **Fact 4** and Assumption 2; we scale the inequality since $\lambda\le 1$ and applying Assumption 1 and 2 in (c). Here we construct the matrix form of Algorithm 2 as follows:

$$\begin{aligned}\mathbf{M}^{(l)} &= \lambda\mathbf{G}^{(l)}+(1-\lambda)\mathbf{Y}^{(t)}\\ \mathbf{X}^{(l+1)} &= \mathbf{W}\left((1+\beta)\mathbf{X}^{(l)}-\beta\mathbf{W}\mathbf{X}^{(l-1)}-(1+\alpha\beta)\eta\mathbf{M}^{(l)}+\alpha\beta\eta\mathbf{W}\mathbf{M}^{(l-1)}\right)\\ \mathbf{Y}^{(t+1)} &= \mathbf{W}\left(\mathbf{Y}^{(t)}+\frac{2\mathbf{X}^{(t)}-\mathbf{X}^{(t+1)}-\mathbf{X}^{(t-1)}}{\eta}\right).\end{aligned} \tag{40}$$

Based on the updated rule in (40), we have

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{(l+1)}-\bar{\mathbf{X}}^{(l+1)}\right\|_F^2 &\overset{(a)}{\le} \frac{1-\rho}{n}\mathbb{E}\left\|(1+\beta)\left(\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right)-\beta\left(\mathbf{W}\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right)-(1+\alpha\beta)\eta\left(\mathbf{M}^{(l)}-\bar{\mathbf{M}}^{(l)}\right)\right.\\ &\quad\left.+\alpha\beta\eta\left(\mathbf{W}\mathbf{M}^{(l-1)}-\bar{\mathbf{M}}^{(l-1)}\right)\right\|_F^2\\ &\overset{(b)}{\le} \frac{2(1-\rho)}{n}\mathbb{E}\left\|(1+\beta)\left(\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right)-\beta\left(\mathbf{W}\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right)\right\|_F^2\\ &\quad+\frac{2(1-\rho)}{n}\mathbb{E}\left\|(1+\alpha\beta)\eta\left(\mathbf{M}^{(l)}-\bar{\mathbf{M}}^{(l)}\right)-\alpha\beta\eta\left(\mathbf{W}\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right)\right\|_F^2\\ &\overset{(c)}{\le} \frac{4(1-\rho)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2+\frac{4(1-\rho)\beta_0^2}{n}\mathbb{E}\left\|\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right\|_F^2\\ &\quad+\frac{4(1-\rho)\beta_0^2\eta^2}{n}\mathbb{E}\left\|\mathbf{M}^{(l)}-\bar{\mathbf{M}}^{(l)}\right\|_F^2+\frac{4(1-\rho)\beta_0^2\eta^2}{n}\mathbb{E}\left\|\mathbf{M}^{(l-1)}-\bar{\mathbf{M}}^{(l-1)}\right\|_F^2\\ &\overset{(d)}{\le} \frac{8(1-\rho)\beta_0^2}{n}\left(\mathbb{E}\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2+\mathbb{E}\left\|\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right\|_F^2\right)\\ &\quad+\frac{2(1-\rho)\beta_0^2}{nL^2}\left(\mathbb{E}\left\|\mathbf{Y}^{(t)}-\bar{\mathbf{Y}}^{(t)}\right\|_F^2+\mathbb{E}\left\|\mathbf{Y}^{(t-1)}-\bar{\mathbf{Y}}^{(t-1)}\right\|_F^2\right)\\ &\quad+8\beta_0^2\zeta^2L^{-2}+8\beta_0^2\sigma^2L^{-2},\end{aligned} \tag{41}$$

where (a) follows because of Assumption 3; (b), (c) follows from **Fact 4** where $a=1$. We scale $1+\beta$, β, $1+\alpha\beta$, $\alpha\beta$ to β_0 since we define $\beta_0=\max\{1+\alpha\beta,1+\beta\}$; we apply the last inequality of (39) into the (d), and it holds because we assume that $\eta\le 1/2L$.

Next, we can describe the gradient tracking progress in consensus made each round. Note that $\mathbf{W}\prec\mathbf{I}$, we have $\mathbf{I}-(1+\beta)\mathbf{W}\prec 2\mathbf{I}$ and $-\mathbf{W}\prec-\mathbf{I}$. It is noted that

$$2\mathbf{X}^{(l)}-\mathbf{X}^{(l+1)}-\mathbf{X}^{(l-1)}\le 3\mathbf{X}^{(l)}+(\beta-1)\mathbf{X}^{(l-1)}+(1+\alpha\beta)\eta\mathbf{M}^{(l)}-\alpha\beta\eta\mathbf{M}^{(l-1)}. \tag{42}$$

Begin with the updated rule in (40), we can derive further by the same process in (41):

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{Y}^{(t+1)}-\bar{\mathbf{Y}}^{(t+1)}\right\|_F^2 &\le \frac{64\beta_0^2L^2(1-\rho)}{n}\mathbb{E}\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2+\frac{64\beta_0^2L^2(1-\rho)}{n}\mathbb{E}\left\|\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right\|_F^2\\ &\quad+\frac{18\beta_0^2(1-\rho)}{n}\mathbb{E}\left\|\mathbf{Y}^{(t)}-\bar{\mathbf{Y}}^{(t)}\right\|_F^2+\frac{18\beta_0^2(1-\rho)}{n}\mathbb{E}\left\|\mathbf{Y}^{(t-1)}-\bar{\mathbf{Y}}^{(t-1)}\right\|_F^2\\ &\quad+128\beta_0^2\zeta^2+64\beta_0^2\sigma^2,\end{aligned} \tag{43}$$

the above inequality holds when learning rate $\frac{3}{2\beta_0L}\le\eta$. Simply adding the results of (41) and (43), yielding

$$\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{(l+1)}-\bar{\mathbf{X}}^{(l+1)}\right\|_F^2+\frac{1}{n}\mathbb{E}\left\|\mathbf{Y}^{(t+1)}-\bar{\mathbf{Y}}^{(t+1)}\right\|_F^2$$

$$\leq \left(\frac{64\beta_0^2L^2(1-\rho)}{n}+\frac{8\beta_0^2(1-\rho)}{n}\right)\left(\mathbb{E}\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2+\mathbb{E}\left\|\mathbf{X}^{(l-1)}-\bar{\mathbf{X}}^{(l-1)}\right\|_F^2\right)$$
$$+\left(\frac{18\beta_0^2(1-\rho)}{n}+\frac{2(1-\rho)\beta_0^2}{nL^2}\right)\left(\mathbb{E}\left\|\mathbf{Y}^{(t)}-\bar{\mathbf{Y}}^{(t)}\right\|_F^2+\mathbb{E}\left\|\mathbf{Y}^{(t-1)}-\bar{\mathbf{Y}}^{(t-1)}\right\|_F^2\right) \tag{44}$$
$$+128\beta_0^2\zeta^2+64\beta_0^2\sigma^2+8\beta_0^2\zeta^2L^{-2}+8\beta_0^2\sigma^2L^{-2}.$$

By Proposition 3, we can get the upper bound of E_4. Before that, we have $\mathbf{x}_i^{(0)}=\bar{\mathbf{x}}^{(0)}=\mathbf{x}_0$ and $\mathbf{y}_i^{(0)}=\mathbf{g}_i^{(0),0}=\nabla F_i(\mathbf{x}_i^{(0),0},\xi_i^{(0),0})$ at the initial stage, and assuming that $\mathbf{x}_i^{(-1)}=\mathbf{y}_i^{(-1)}=\mathbf{0}$. We denote $Q_2=\max\left\{8\beta_0^2(1-\rho)(8L^2+1),2\beta_0^2(1-\rho)\left(9+\frac{1}{L^2}\right)\right\}$ and easily obtain that

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{(1)}-\bar{\mathbf{X}}^{(1)}\right\|_F^2&\leq\frac{Q_2}{n}\mathbb{E}\left\|\mathbf{X}^{(0)}-\bar{\mathbf{X}}^{(0)}\right\|_F^2+\frac{Q_2}{n}\mathbb{E}\left\|\mathbf{X}^{(-1)}-\bar{\mathbf{X}}^{(-1)}\right\|_F^2+8\beta_0^2\zeta^2L^{-2}+8\beta_0^2\sigma^2L^{-2}\\&=8\beta_0^2\zeta^2L^{-2}+8\beta_0^2\sigma^2L^{-2},\end{aligned} \tag{45}$$

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{Y}^{(1)}-\bar{\mathbf{Y}}^{(1)}\right\|_F^2&\leq\frac{Q_2}{n}\mathbb{E}\left\|\mathbf{Y}^{(0)}-\bar{\mathbf{Y}}^{(0)}\right\|_F^2+128\beta_0^2\zeta^2+64\beta_0^2\sigma^2\\&=\frac{Q_2}{n}\mathbb{E}\left\|\mathbf{G}^{(0)}-\bar{\mathbf{G}}^{(0)}\right\|_F^2+128\beta_0^2\zeta^2+64\beta_0^2\sigma^2+8\beta_0^2\zeta^2L^{-2}\\&\leq Q_2(8\zeta^2+4\sigma^2)+128\beta_0^2\zeta^2+64\beta_0^2\sigma^2.\end{aligned} \tag{46}$$

We denote $\Psi_1=\max_{0\leq l\leq KT-2}\left\{\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2\right\}$ and $\Psi_2=\max_{0\leq t\leq T-2}\left\{\frac{1}{n}\mathbb{E}\left\|\mathbf{Y}^{(t)}-\bar{\mathbf{Y}}^{(t)}\right\|_F^2\right\}$. For E_4, it gets

$$\begin{aligned}E_4&\leq\frac{1}{n}\sum_{l=0}^{KT-1}\mathbb{E}_l\left\|\mathbf{X}^{(l)}-\bar{\mathbf{X}}^{(l)}\right\|_F^2+\frac{1}{n}\sum_{t=0}^{T-1}\mathbb{E}_t\left\|\mathbf{Y}^{(t)}-\bar{\mathbf{Y}}^{(t)}\right\|_F^2\\&\leq\frac{Q_2(8\zeta^2+4\sigma^2)}{n(1-Q_2)}+\frac{KT\Psi_1}{1-Q_2}+\frac{T\Psi_2}{1-Q_2}+KT\frac{128\beta_0^2\zeta^2+64\beta_0^2\sigma^2+8\beta_0^2\zeta^2L^{-2}+8\beta_0^2\sigma^2L^{-2}}{n(1-Q_2)},\end{aligned} \tag{47}$$

where the last inequality holds by assume that $\max\left\{1-\frac{1}{8\beta_0^2(1+8L^2)},1-\frac{L^2}{2\beta_0^2(1+9L^2)}\right\}<\rho\leq 1$ to make sure that $Q_2<1$. Substitute (47) into (38), we have

$$\begin{aligned}\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2&\leq\frac{2(1-\beta)}{\eta KT}\left(\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})\right)+\frac{2\beta\eta\lambda\alpha^2\sigma^2}{n(1-\beta)^2}+\frac{L^2(2-\lambda)(\Psi_1+\Psi_2)}{1-Q_2}\\&+\frac{Q_2L^2(2-\lambda)(8\zeta^2+4\sigma^2)}{nKT(1-Q_2)}+\frac{2L\eta\lambda^2\sigma^2}{n(1-\beta)}\\&+(2-\lambda)\frac{128\beta_0^2\zeta^2L^2+64\beta_0^2\sigma^2L^2+8\beta_0^2\zeta^2+8\beta_0^2\sigma^2}{n(1-Q_2)}+\frac{2\epsilon^2(1-\lambda)}{n}.\end{aligned} \tag{48}$$

Since $\beta\in[0,1)$, $\lambda\in[0,1]$ and $\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})$ is constant, the first two terms of (48) can be reduced to $\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta\alpha^2\sigma^2}{n}\right)$ after omitting $1-\beta$, $\mathbb{E}f(\bar{\mathbf{z}}^{(0),0})-\mathbb{E}f(\bar{\mathbf{z}}^{(T),0})$ and λ. Based on Lemma 3, ψ_1 and ψ_2 can be regarded as 0 when T is large enough. Therefore, the third term can be omitted. Since $Q_2=\max\left\{8\beta_0^2(1-\rho)(8L^2+1),2\beta_0^2(1-\rho)\left(9+\frac{1}{L^2}\right)\right\}$ and λ, L, β and β_0 are constants, the 4th to 6th terms can be reduced to $\mathcal{O}\left(\frac{\zeta^2+\sigma^2}{nKT\rho}\right)+\mathcal{O}\left(\frac{\eta\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\zeta^2+\sigma^2}{n\rho}\right)$. We omit the smallest in the three terms (*i.e.*, $\mathcal{O}\left(\frac{\zeta^2+\sigma^2}{nKT\rho}\right)$). By omitting constant $(1-\lambda)$, the last term can be reduced to $\mathcal{O}\left(\frac{\epsilon}{n}\right)$. Finally, we can get the convergence result of GT-DSUM:

$$\frac{1}{KT}\sum_{t=0}^{T-1}\sum_{\tau=0}^{K-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t),\tau})\right\|^2\leq\mathcal{O}\left(\frac{1}{\eta KT}+\frac{\eta(1+\alpha^2)\sigma^2}{n}\right)+\mathcal{O}\left(\frac{\zeta^2+\sigma^2}{n\rho}\right)+\mathcal{O}\left(\frac{\epsilon}{n}\right). \tag{49}$$

The convergence results indicate that parameter α has a significant impact on the convergence rates of the two proposed algorithms. In Section 5, we conduct hyperparameters tuning to examine the influence of α on model performance. However, it is worth noting that the convergence results merely provide upper bounds. Our extensive experiments demonstrate that our proposed algorithms outperform compared baselines with better testing accuracy and faster convergence rate, particularly when an appropriate α is selected.

5. Evaluation

Our main evaluation results demonstrate that D-SUM outperforms other methods in terms of model accuracy, and GT-DSUM achieves a higher performance under different levels of non-IID. All experiments are executed in a CPU/GPU cluster, equipped with

Table 2
Datasets and Models.

Dataset	Task	Training samples	Testing samples	Classes	Model
MNIST [25]	Handwritten character recognition (CV)	60,000	10,000	10	LeNet described in Table 3
EMNIST [3]	Handwritten character recognition (CV)	731,668	82,587	62	CNN described in Table 4
CIFAR10 [24]	Image classification (CV)	50,000	10,000	10	LeNet described in Table 3
AG NEWS [65]	Text classification (NLP)	120,000	7,600	4	RNN described in Table 6
SVHN [36]	Digit classification (CV)	630,420	26,032	10	VGG16 described in Table 5

Table 3
LeNet model on MNIST and CIFAR10.

Layer	Output Shape	Hyperparameters	Activation
Conv2D	(28, 28, 6)	kernel size = 5	ReLU
MaxPool2D	(14, 14, 6)	pool size = 2	
Conv2D	(10, 10, 16)	kernel size = 5	ReLU
MaxPool2D	(5, 5, 16)	pool size = 2	
Flatten	400		
Dense	120		
Dense	84		
Dropout	84	$p = 0.5$	
Dense	10		

Inter(R) Xeon(R) Gold 6126, 4 GTX 2080Ti cards, and 12 Tesla T4 cards. We used Pytorch and Ray [34] to implement and train our models.

5.1. Experiment methodology

Baselines. We consider the following three decentralized methods with momentum, which are described as follows:

- *Local SGD* [43] periodically averages model parameters among all worker nodes. Compared with the vanilla SGD, each node independently runs the single-node SGD with Heavy Ball momentum.
- *QG-DSGDm* [31] mimics the global optimization direction and integrates the quasi-global momentum into local stochastic gradients without causing extra communication costs. It empirically mitigates the impact on data heterogeneity.
- *SlowMo* [54] performs a slow, periodical momentum update through an All-Reduce pattern (model averaging) after multiple SGD steps. For simplicity, we use the common mini-batch gradient as the local update direction.

Datasets and models. We study the decentralized behaviors on both computer vision (CV) and natural language processing (NLP) tasks, including MNIST, EMNIST, CIFAR10, SVHN, and AG NEWS. An overall description is given in Table 2. Concretely, MNIST is a 10-class handwritten digits image classification dataset with $70,000$ 28×28 examples, $60,000$ of which are training datasets, and the remaining $10,000$ are test datasets. Its extended version, EMNIST consists of images of digits and upper and lower case English characters, which includes 62 total classes. CIFAR10 is labeled subsets of the 80 million images dataset, sharing the same $60,000$ input images with 10 unique labels. SVHN is a 10-class digit classification benchmark dataset that contains more than $600,000$ images of printed digits cropped from pictures of house number plates. For all CV tasks, we train different CNN architectures, which are described in Table 3, 4 and 5. For NLP, AG NEWS is a 4-class classification dataset on categorized news articles, containing $120,000$ training samples and $7,600$ testing samples. We train a simple RNN, which includes an embedding layer, a dropout layer, followed by a dense layer. A full description is in Table 6.

Hyperparameters. For all algorithms with different benchmarks, the cross-device setting deploys 10 workers training by default. In our experiments, we set the local mini-batch size as 256 for CIFAR10, and 128 for the rest, and the number of local updates is set as $K = 10$. To illustrate the challenge of data heterogeneity in decentralized deep training, we adopt the Dirichlet distribution value [31] to control different levels of non-IID degree, for the case with non-IID $= 0.1, 1, 10$; the smaller the value is, the more likely the workers hold samples from only one class of labels (*i.e.*, non-IID $= 0.1$ can be viewed as an extreme data skewness case). Among choices of $\mathbf{W}$ considered in practice, we pre-construct a dynamic topology changing sequence varying from full-mesh to ring by the popular Metropolis-Hastings rule [21] *i.e.*, $w_{ij} = w_{ji} = \min\left\{\frac{1}{\deg(i)+1}, \frac{1}{\deg(j)+1}\right\}$ for any i, j, $w_{ii} = 1 - \sum_{j=1}^{n} w_{ij}$. The learning rate η is fine-tuned via a grid search on the set $\left\{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}\right\}$ for each algorithm and dataset. Besides, we set the scalar α, momentum β, normalized parameter λ as 2, 0.9, and 0.8 respectively by default.

Table 4
CNN model on EMNIST.

Layer	Output Shape	Hyperparameters	Activation
Conv2D	(26, 26, 32)	kernel size = 3, strides = 1	
Conv2D	(24, 24, 64)	kernel size = 3, strides = 1	ReLU
MaxPool2D	(12, 12, 64)	pool size = 2	
Dropout	(12, 12, 64)	$p = 0.25$	
Flatten	9,216		
Dense	128		
Dropout	128	$p = 0.5$	
Dense	62		softmax

Table 5
VGG16 model on SVHN.

Layer	Output Shape	Hyperparameters	Activation
2xConv2D	(32, 32, 64)	kernel size = 3, strides = 1	
MaxPool2D	(16, 16, 64)	pool size = 2	ReLU
2xConv2D	(16, 16, 128)	kernel size = 3, strides = 1	
MaxPool2D	(8, 8, 128)	pool size = 2	ReLU
3xConv2D	(8, 8, 256)	kernel size = 3, strides = 1	
MaxPool2D	(4, 4, 256)	pool size = 2	ReLU
3xConv2D	(4, 4, 512)	kernel size = 3, strides = 1	
MaxPool2D	(2, 2, 512)	pool size = 2	ReLU
3xConv2D	(2, 2, 512)	kernel size = 3, strides = 1	
MaxPool2D	(1, 1, 512)	pool size = 2	ReLU
Flatten	512		
Dropout	512	$p = 0.25$	
Dense	512		
Dropout	512	$p = 0.25$	
Dense	256		
Dense	10		softmax

Table 6
RNN model on AG NEWS.

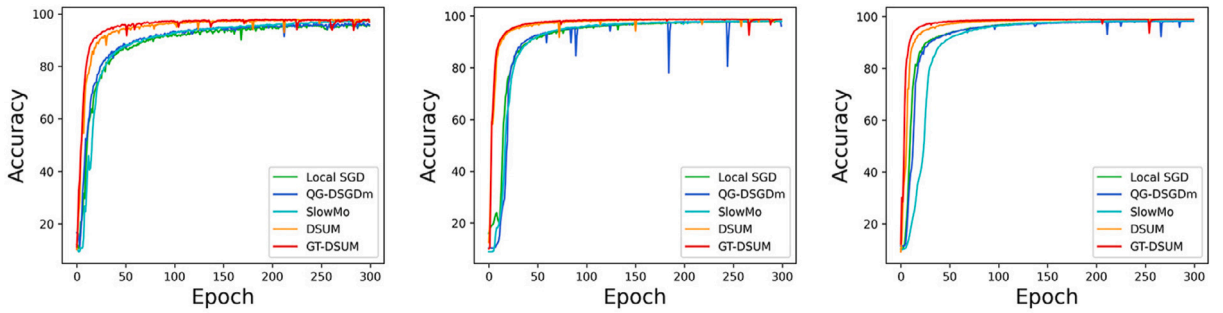
Layer	Hyperparameters
EmbeddingBag	embeddings = 95,812, dimension = 64
Dense	in_features = 64, out_features = 4
Dropout	$p = 0.5$

Performance Metrics. We examine the effects of different momentum variants on decentralized deep learning including

- *Model generalization* is measured by the proportion between the amount of the correct data by the model and that of all data in the test dataset. We report the averaged model performance of local models over test samples.
- *Effect of different hyperparameters* is explored by tuning their values to study the properties of `D-SUM` and `GT-DSUM`. In our evaluation, we show the impact of the following factors:
  - The number of local updates K is one of the most important parameters since it influences the final model generalization and training time. In usual, the number of K is set less than 20 [40,39]. Hence, we present the comparison with $K \in \{1, 5, 10, 15, 20\}$.
  - The momentum term β is further investigated via grid search on the set $\{0, 01, 0.1, 0.5, 0.9, 0.99\}$ as different strategies for handling algorithms.
  - One of the most important parameters α is varying from 0 to 15 to analyze its influence on the convergence performance.
- *Scalability* is a crucial property to have while handling tasks in a distributed situation. We evaluate this by extending the framework scale by adjusting the number of devices n training on 4, 10, 16, and 32 workers.

Table 7
The testing accuracy with different algorithms on various training benchmarks and different degrees of non-IID.

Datasets	Algorithms	Testing Accuracy (%)		
		non-IID = 0.1	non-IID = 1	non-IID = 10
MNIST [25]	`Local SGD w/ momentum`	95.66 ± 0.21	97.99 ± 0.03	98.39 ± 0.03
	`QG-DSGDm`	96.02 ± 0.19	97.46 ± 1.36	98.21 ± 0.04
	`SlowMo`	97.32 ± 0.02	97.93 ± 0.07	98.34 ± 0.06
	`D-SUM` (ours)	**97.89** ± 0.21	**98.77** ± 0.04	**98.94** ± 0.01
	`GT-DSUM` (ours)	97.51 ± 0.61	98.70 ± 0.01	98.82 ± 0.03
EMNIST [3]	`Local SGD w/ momentum`	45.90 ± 1.21	36.77 ± 0.13	38.29 ± 0.03
	`QG-DSGDm`	46.03 ± 0.6	46.02 ± 0.12	36.72 ± 0.02
	`SlowMo`	45.52 ± 0.03	37.11 ± 0.01	37.50 ± 0.0
	`D-SUM` (ours)	49.68 ± 0.43	49.75 ± 0.05	42.50 ± 0.01
	`GT-DSUM` (ours)	**50.49** ± 0.82	**50.25** ± 0.07	**51.87** ± 0.02
CIFAR10 [24]	`Local SGD w/ momentum`	22.94 ± 1.11	42.93 ± 0.85	52.82 ± 0.01
	`QG-DSGDm`	26.34 ± 1.42	49.12 ± 038	54.03 ± 0.24
	`SlowMo`	31.06 ± 1.27	50.46 ± 0.04	55.50 ± 0.10
	`DSUM` (ours)	31.16 ± 1.27	54.34 ± 0.11	57.59 ± 1.05
	`GT-DSUM` (ours)	**36.16** ± 0.74	**56.95** ± 1.56	**59.34** ± 1.55
AG NEWS [65]	`Local SGD w/ momentum`	75.51 ± 0.44	77.98 ± 0.39	80.66 ± 0.02
	`QG-DSGDm`	78.82 ± 0.31	79.33 ± 0.38	82.24 ± 0.02
	`SlowMo`	82.57 ± 0.03	83.17 ± 0.01	83.79 ± 0.01
	`DSUM` (ours)	84.13 ± 0.55	85.46 ± 0.31	87.52 ± 0.04
	`GT-DSUM` (ours)	**84.29** ± 0.37	**87.59** ± 0.18	**89.07** ± 0.04

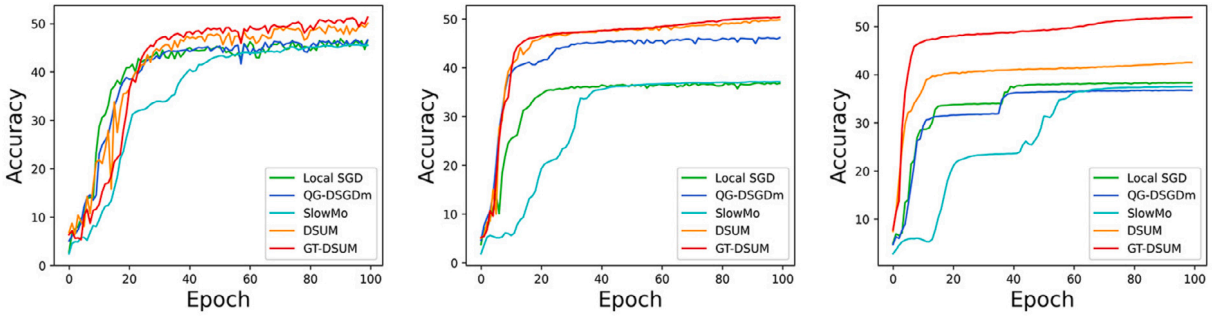


(a) non-IID = 0.1 (b) non-IID = 1 (c) non-IID = 10

Fig. 1. Testing accuracy for various tasks training on **LeNet** over **MNIST**. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

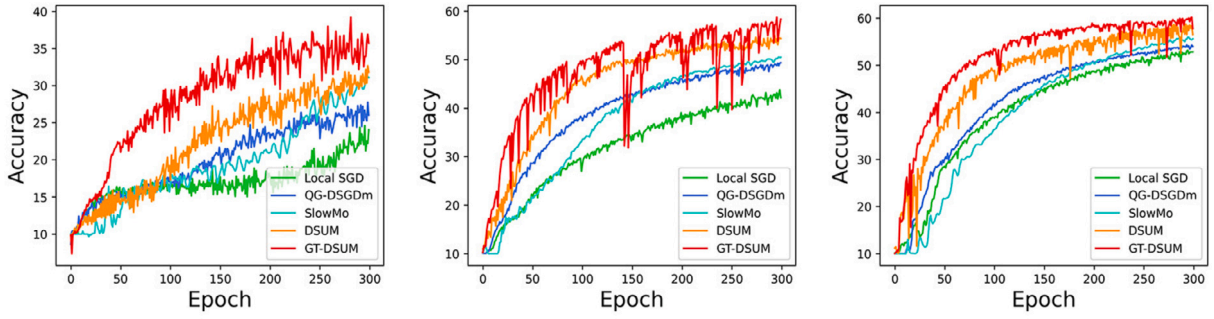
5.2. Evaluation results

Performance with compared baselines. Table 7 and Fig. 1 to 4 examine the effect of our proposed algorithms with compared baselines on different datasets. In Table 7, we can see that our proposed algorithms outperform all other baselines across different levels of data skewness. For training LeNet over MNIST, although all methods achieve perfect model performance, `D-SUM` and `GT-DSUM` still significantly improve the model accuracy compared with the other three baselines. Comparing `D-SUM` and `GT-DSUM`, the latter is inferior to the former one, but we observe that the model's precision could be negligible. On the EMNIST dataset, we observe that `GT-DSUM` has achieves the best performance under different non-IID levels, improving the accuracy results 10.0%, 36.7% and 35.5% compared with `Local SGD w/ momentum`. For CIFAR10 and AG NEWS, the performance of our algorithms and benchmarks: `GT-DSUM` > `D-SUM` > `SlowMo` > `QG-DSGDm` > `Local SGD w/ momentum`, noting that our proposed algorithms outperform other benchmarks both on accuracy and convergence perspectives and demonstrate that the GT technique shows its effectiveness in mitigating the negative impact caused by data heterogeneity. `D-SUM` improves accuracy up to 35.8% when non-IID = 1. Moreover, as the non-IID level increases, leading to model deterioration and poor convergence performance, `GT-DSUM` still achieves the accuracy of 59.34%, 56.95%, 36.16%, which is higher than `Local SGD w/ momentum` 12.3%, 32.7%, and 57.6% on CIFAR10, respectively. For NLP tasks, the evaluation results indicate that our proposed algorithms benefit. For example, `D-SUM` achieves 84.13%, 85.46%, and 87.52% on testing accuracy compared with other baselines which perform a poor generalization under three different degrees of data heterogeneity, changing from non-IID to IID. Moreover, `GT-DSUM` further improves the accuracy results 11.6%, 12.3% and 10.4% better than `Local SGD w/ momentum`. Fig. 1, 2, 3 and 4 present the experimental results on the model training process



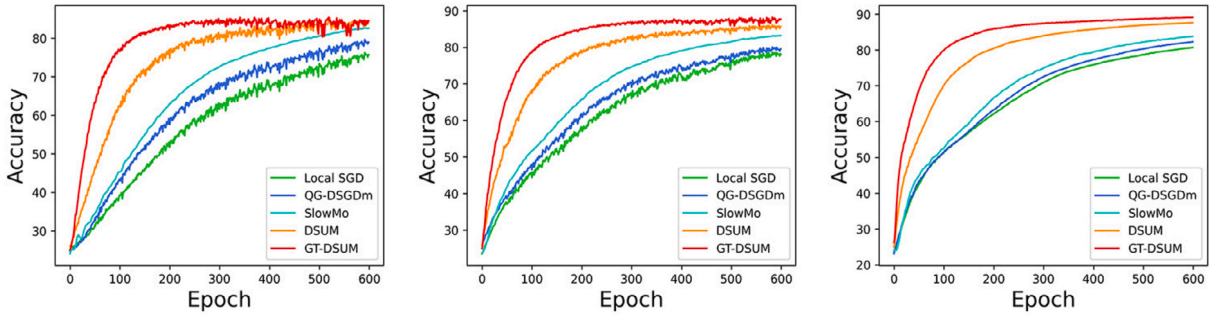
(a) non-IID = 0.1 (b) non-IID = 1 (c) non-IID = 10

Fig. 2. Testing accuracy for various tasks training on **CNN** over **EMNIST**.



(a) non-IID = 0.1 (b) non-IID = 1 (c) non-IID = 10

Fig. 3. Testing accuracy for various tasks training on **LeNet** over **CIFAR10**.



(a) non-IID = 0.1 (b) non-IID = 1 (c) non-IID = 10

Fig. 4. Testing accuracy for various tasks training on **RNN** over **AG NEWS**.

for different benchmarks. We note that the superiority of our proposed methods is better reflected in the convergence acceleration. For example, both `D-SUM` and `GT-DSUM` require about 50 epochs to reach convergence for LeNet over MNIST, which reduces the number of training epochs by 40%. Switching to large datasets, *i.e.*, CIFAR10 and AG NEWS, our proposed algorithms converge faster than other baselines with respect to the training epochs. In summary, our proposed algorithms outperform other compared baselines with better testing accuracy and faster convergence rate. Besides, `GT-DSUM` also achieves a higher training performance under different non-IID levels for those datasets with a relatively larger scale. Moreover, we also show algorithms' performance in terms of convergence rate on the test loss of VGG16 trained on the SVHN benchmark. In Fig. 5, we can see that `D-SUM` achieves the fastest convergence speed when noniid = 10. `GT-DSUM` seems to have a slow convergence speed at first. However, when epoch = 30, there is a phase of acceleration and `GT-DSUM` ultimately retches the lowest loss value. This indicates that our proposed algorithms still perform well on complex neural network architectures.

Effect of local update. We also evaluate how the number of local updates K influences the training performance, showing the results over four datasets on `Local SGD w/ momentum` and two proposed algorithms when non-IID = 1 in Fig. 6. We make two observations from the results. Firstly, our algorithms have better performance than `Local SGD w/ momentum` regardless K. Train-

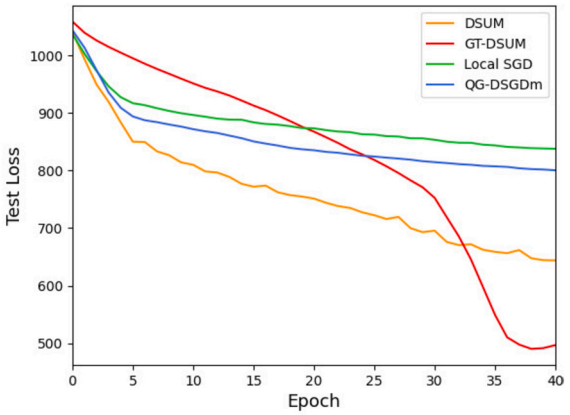
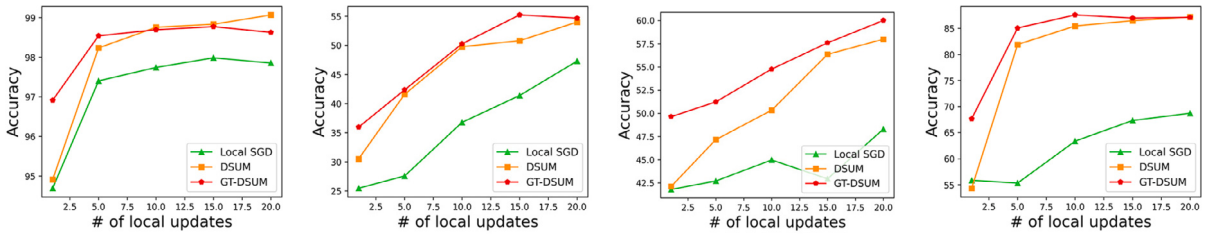
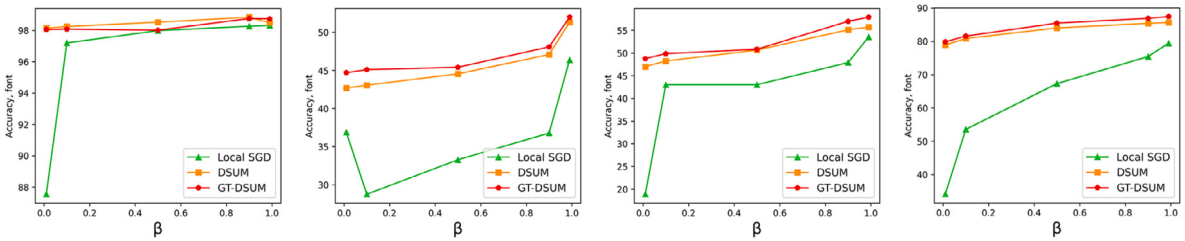


Fig. 5. Test loss for the task training on **VGG16** over **SVHN** when noniid = 10.



(a) LeNet over MNIST (b) CNN over EMNIST (c) LeNet over CIFAR10 (d) RNN over AG NEWS

Fig. 6. Impact on the number of local updates K on the convergence when momentum $\beta = 0.9$ under the non-IID = 1 case.



(a) LeNet over MNIST (b) CNN over EMNIST (c) LeNet over CIFAR10 (d) RNN over AG NEWS

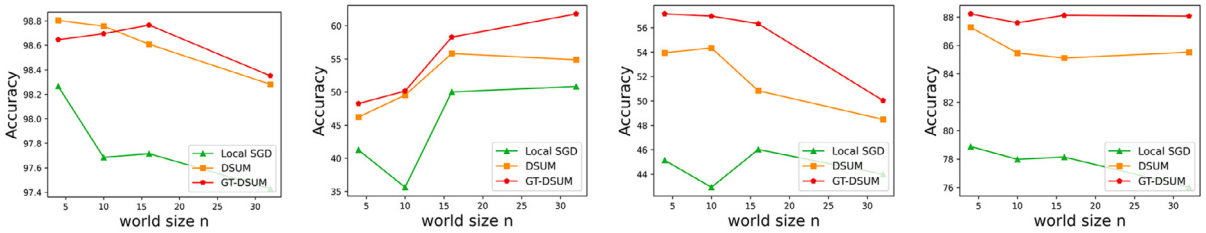
Fig. 7. Impact on the momentum β on the convergence when the number of local update $K = 10$ under the non-IID = 1 case.

ing LeNet over MNIST in Fig. 6(a), `D-SUM` achieves a higher testing accuracy 99.07% when $K = 20$. For CIFAR10 and AG NEWS, shown in Fig. 6(c), 6(d), we observe a similar phenomenon that `GT-DSUM` always keep competitive when the number of local updates increases. Among them, it reaches 57.58% and 85.07%, improving 34.1% and 53.8% than `Local SGD w/ momentum` when $K = 15$ and $K = 5$, respectively. Secondly, we can find see that the workers may not guarantee to improve the model generalization substantially by increasing the number of local updates K, *e.g.*, test precision does not monotonically increase with K. For example, we find that $K = 10$ can obtain the best performance over AG NEWS, and $K = 15$ over MNIST and EMNIST for `GT-DSUM`. Besides, all benchmarks perform worst when $K = 1$, while when K is too large performance may be degraded because workers' local models drift too far apart in distributed optimizations [39,54].

Effect of β. The third set of simulations evaluates the performance of model accuracy on different β, which are depicted in Fig. 7. We have a key observation from those results. Regardless of datasets or models, evaluation results with a greater β (*e.g.*, 0.9, 0.99) trend to outperform with a smaller one (*e.g.*, 0.01, 0.1). In addition, it can be observed that the testing accuracy monotonically increases with β on CIFAR10 and AG NEWS. For EMNIST, CIFAR10 and AG NEWS, as depicted in Fig. 7(b), 7(c), 7(d), we note that `GT-DSUM` reaches a higher model accuracy compared with `Local SGD w/ momentum` when β is creasing. For example, the accuracy achieves 45.09%, 49.86%, 81.57% when $\beta = 0.1$ for `GT-DSUM`, improving the accuracy results 56.8%, 16.0%, 51.9% better than `Local SGD w/ momentum`.

Table 8
The impact of α for D-SUM and GT-DSUM on the test accuracy with non-IID = 1. "$\star$" indicates non-convergence.

Datasets	Methods	The test accuracy (%) evaluated on different α under the non-IID = 1 case									
		$\alpha=0$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=3$	$\alpha=4$	$\alpha=5$	$\alpha=8$	$\alpha=10$	$\alpha=15$
MNIST	D-SUM	98.05	98.40	98.57	98.76	98.85	98.76	98.85	**99.09**	98.87	92.86
	GT-DSUM	98.18	98.50	98.70	98.70	**98.80**	$\star$	$\star$	$\star$	$\star$	$\star$
EMNIST	D-SUM	37.1	43.58	35.58	49.75	**55.32**	49.80	43.00	48.47	53.04	$\star$
	GT-DSUM	33.72	46.06	47.10	**50.25**	39.84	$\star$	$\star$	$\star$	$\star$	$\star$
CIFAR10	D-SUM	47.10	50.85	51.68	50.32	**54.83**	51.10	51.53	50.68	$\star$	$\star$
	GT-DSUM	45.98	49.80	50.55	54.77	**57.58**	54.43	$\star$	$\star$	$\star$	$\star$
AG NEWS	D-SUM	79.16	81.78	83.72	85.46	86.38	86.36	88.20	87.07	**88.82**	88.56
	GT-DSUM	78.90	83.12	85.34	**87.59**	86.89	77.67	$\star$	$\star$	$\star$	$\star$



(a) LeNet over MNIST (b) CNN over EMNIST (c) LeNet over CIFAR10 (d) RNN over AG NEWS

Fig. 8. Impact on the momentum world size on the convergence when the number of local updates $K = 10$ under the non-IID = 1 case.

Sensitivity of α. We conduct the hyperparameter tuning to examine the influence of parameter α on model performance, which adopted the default parameter settings for K, β and λ. The results are shown in Table 8. Two observations are as follows. Firstly, we note that different optimal values of α are always found when D-SUM is evaluated on various datasets with the same level of data heterogeneity (*e.g.*, non-IID = 1). For instance, D-SUM achieves an accuracy of 99.09% with LeNet over MNIST when it takes the optimal value of $\alpha = 8$, and 88.82% for AG NEWS with $\alpha = 10$, respectively. It is hard and time-consuming to determine the optimum α due to different characteristics of datasets and models, however, there always exists an optimal α between 0 and 15 under different benchmarks in general. Secondly, as α increases, both D-SUM and GT-DSUM make a significant degradation in model performance. This phenomenon verifies the analysis detailed in Remark 1 and 2, which indicates that due to the higher order, α largely affects the generalization of global models optimized by the proposed two algorithms. Besides, GT-DSUM requires a stricter constraint on α than D-SUM to ensure the model's validity. From Table 8, we can see that GT-DSUM leads to non-convergence when $\alpha \in (3, 5)$. Especially, the accuracy first decreases when α is about 3 when training CNN over EMNIST as well as RNN over AG NEWS. Empirically, there is still a gap between the vulnerable property of α to momentum-based optimizer and the robustness endowing with superior performance. Furthermore, the effectiveness and robustness of UMP should be further investigated by extensive experiments.

Scalability. To further study the scaling properties of D-SUM and GT-DSUM, we final train on different numbers of workers compared with Local SGD w/ momentum when non-IID = 1. For simplicity, model parameters are transmitted between each other according to the Ring-Reduce pattern in this section. Results are shown in Fig. 8. When the number of participating workers increases, the advantage of our schemes is readily apparent since our method GT-DSUM consistently reaches a higher model accuracy compared to the Local SGD w/ momentum in this non-IID case. For example, the accuracy achieves 98.77%, 56.33% and 88.12% for EMNIST, CIFAR10 and AG NEWS when $n = 16$ for GT-DSUM, having an improvement 1.1%, 10.8% and 12.8% higher than that of Local SGD w/ momentum, respectively.

6. Conclusion

In this paper, we propose a unified momentum-based paradigm UMP with two algorithms D-SUM and GT-DSUM. The former obtains good model generalization, dealing with the validity under non-convex cases, while the latter is further developed by applying the GT technique to eliminate the negative impact of heterogeneous data. A range of algorithms incorporating momentum techniques can be obtained by specifying the parameters of our proposed algorithms. By deriving the convergence of general non-convex settings, these algorithms achieve competitive performance closely related to a critical parameter α. Extensive experimental results show our UMP leads to at most 57.6% increase in improvement of accuracy.

CRediT authorship contribution statement

Haizhou Du: Conceptualization, Formal analysis, Project administration, Supervision, Writing – original draft, Writing – review & editing. **Chaoqian Cheng:** Data curation, Formal analysis, Investigation, Writing – review & editing. **Chengdong Ni:** Formal analysis, Methodology, Visualization, Writing – original draft.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "A Unified Momentum-based Paradigm of Decentralized SGD for Non-Convex Models and Heterogeneous Data".

Data availability

No data was used for the research described in the article.

References

[1] S. Arora, Z. Li, A. Panigrahi, Understanding gradient descent on the edge of stability in deep learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 948–1024.
[2] A. Balu, Z. Jiang, S.Y. Tan, C. Hedge, Y.M. Lee, S. Sarkar, Decentralized deep learning using momentum-accelerated consensus, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3675–3679.
[3] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, EMNIST: extending mnist to handwritten letters, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2921–2926.
[4] A. Cutkosky, H. Mehta, Momentum improves normalized SGD, in: International Conference on Machine Learning, 2020, pp. 2260–2268.
[5] A. Defazio, Understanding the role of momentum in non-convex optimization: practical insights from a Lyapunov analysis, arXiv preprint arXiv:2010.00406, 2020.
[6] Q. Deng, W. Gao, Minibatch and momentum model-based methods for stochastic weakly convex optimization, Adv. Neural Inf. Process. Syst. 34 (2021).
[7] P. Di Lorenzo, G. Scutari, Next: in-network nonconvex optimization, IEEE Trans. Signal Inf. Process. Netw. 2 (2016) 120–136.
[8] Y. Esfandiari, S.Y. Tan, Z. Jiang, A. Balu, E. Herron, C. Hegde, S. Sarkar, Cross-gradient aggregation for decentralized learning from non-IID data, in: International Conference on Machine Learning, 2021, pp. 3036–3046.
[9] C. Fang, C.J. Li, Z. Lin, T. Zhang, Spider: near-optimal non-convex optimization via stochastic path-integrated differential estimator, Adv. Neural Inf. Process. Syst. 31 (2018).
[10] H. Gao, H. Huang, Periodic stochastic gradient descent with momentum for decentralized training, arXiv preprint arXiv:2008.10435, 2020.
[11] H. Gao, J. Li, H. Huang, On the convergence of local stochastic compositional gradient descent with momentum, in: Proceedings of the 39th International Conference on Machine Learning, PMLR, 2022, pp. 7017–7035.
[12] E. Ghadimi, H.R. Feyzmahdavian, M. Johansson, Global convergence of the heavy-ball method for convex optimization, in: 2015 European Control Conference (ECC), 2015, pp. 310–315.
[13] A. Ghosh, H. Lyu, X. Zhang, R. Wang, Implicit regularization in heavy-ball momentum accelerated stochastic gradient descent, in: International Conference on Learning Representations, 2023.
[14] I. Gitman, H. Lang, P. Zhang, L. Xiao, Understanding the role of momentum in stochastic gradient methods, Adv. Neural Inf. Process. Syst. 32 (2019).
[15] A. Han, J. Gao, Riemannian stochastic recursive momentum method for non-convex optimization, in: International Joint Conference on Artificial Intelligence, 2021.
[16] K. Hsieh, A. Phanishayee, O. Mutlu, P. Gibbons, The non-IID data quagmire of decentralized machine learning, in: International Conference on Machine Learning, 2020, pp. 4387–4398.
[17] Y. Huang, Y. Sun, Z. Zhu, C. Yan, J. Xu, Tackling data heterogeneity: a new unified framework for decentralized sgd with sample-induced topology, in: International Conference on Machine Learning, PMLR, 2022, pp. 9310–9345.
[18] S. Jelassi, Y. Li, Towards understanding how momentum improves generalization in deep learning, in: Proceedings of the 39th International Conference on Machine Learning, PMLR, 2022, pp. 9965–10040.
[19] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, SCAFFOLD: stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, 2020, pp. 5132–5143.
[20] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2015.
[21] A. Koloskova, T. Lin, S.U. Stich, An improved analysis of gradient tracking for decentralized machine learning, Adv. Neural Inf. Process. Syst. 34 (2021).
[22] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, S. Stich, A unified theory of decentralized SGD with changing topology and local updates, in: International Conference on Machine Learning, 2020, pp. 5381–5393.
[23] A. Koloskova, S. Stich, M. Jaggi, Decentralized stochastic optimization and gossip algorithms with compressed communication, in: International Conference on Machine Learning, 2019, pp. 3478–3487.
[24] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, 2009.
[25] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.
[26] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.
[27] X. Li, W. Yang, S. Wang, Z. Zhang, Communication-efficient local decentralized SGD methods, arXiv preprint arXiv:1910.09126, 2019.
[28] Y. Li, C. Wei, T. Ma, Towards explaining the regularization effect of initial large learning rate in training neural networks, Adv. Neural Inf. Process. Syst. 32 (2019).
[29] Z. Li, X. Shang, R. He, T. Lin, C. Wu, No fear of classifier biases: neural collapse inspired federated learning with synthetic and fixed classifier, arXiv preprint arXiv:2303.10058, 2023.
[30] X. Lian, C. Zhang, H. Zhang, C.J. Hsieh, W. Zhang, J. Liu, Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent, Adv. Neural Inf. Process. Syst. 30 (2017).
[31] T. Lin, S.P. Karimireddy, S.U. Stich, M. Jaggi, Quasi-global momentum: accelerating decentralized deep learning on heterogeneous data, in: International Conference on Machine Learning, 2021, pp. 6654–6665.

[32] Y. Liu, T. Lin, A. Koloskova, S.U. Stich, Decentralized gradient tracking with local steps, arXiv preprint arXiv:2301.01313, 2023.
[33] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, J. Feng, No fear of heterogeneity: classifier calibration for federated learning with non-IID data, Adv. Neural Inf. Process. Syst. 34 (2021) 5972–5984.
[34] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M.I. Jordan, I. Stoica, Ray: a distributed framework for emerging ai applications, in: Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation, 2018, pp. 561–577.
[35] Y.E. Nesterov, A method for solving the convex programming problem with convergence rate $o(1/k^2)$, Dokl. Akad. Nauk SSSR 269 (1983) 543–547.
[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading Digits in Natural Images with Unsupervised Feature Learning, 2011.
[37] B.T. Polyak, Some methods of speeding up the convergence of iteration methods, USSR Comput. Math. Math. Phys. 4 (1964) 1–17.
[38] S. Pu, A. Nedić, Distributed stochastic gradient tracking methods, Math. Program. 187 (2021) 409–457.
[39] Z. Qu, X. Li, Rui L. Duan, Y. Liu, B. Tang, Generalized federated learning via sharpness aware minimization, in: International Conference on Machine Learning, 2022.
[40] S.J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, H.B. McMahan, Adaptive federated optimization, in: International Conference on Learning Representations, 2020.
[41] X. Shang, Y. Lu, G. Huang, H. Wang, Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features, in: International Joint Conference on Artificial Intelligence, 2022, pp. 2218–2224.
[42] P. Sharma, R. Panda, G. Joshi, P. Varshney, Federated minimax optimization: improved convergence analyses and algorithms, in: International Conference on Machine Learning, PMLR, 2022, pp. 19683–19730.
[43] S.U. Stich, Local SGD converges fast and communicates little, in: International Conference on Learning Representations, 2018.
[44] T. Sun, D. Li, B. Wang, Decentralized federated averaging, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[45] I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, in: International Conference on Machine Learning, 2013, pp. 1139–1147.
[46] B. Swenson, R. Murray, H.V. Poor, S. Kar, Distributed stochastic gradient descent: nonconvexity, non-smoothness, and convergence to local minima, J. Mach. Learn. Res. 23 (2022) 14751–14812.
[47] Y. Takezawa, H. Bao, K. Niwa, R. Sato, M. Yamada, Momentum tracking: momentum acceleration for decentralized deep learning on heterogeneous data, arXiv preprint arXiv:2209.15505, 2022.
[48] H. Tang, X. Lian, M. Yan, C. Zhang, J. Liu, D^2: decentralized training over decentralized data, in: International Conference on Machine Learning, 2018, pp. 4848–4856.
[49] W. Tao, S. Long, G. Wu, Q. Tao, The role of momentum parameters in the optimal convergence of adaptive Polyak's Heavy-ball methods, in: 9th International Conference on Learning Representations, ICLR, 2021.
[50] Y. Tao, Y. Wu, X. Cheng, D. Wang, Private stochastic convex optimization and sparse learning with heavy-tailed data revisited, in: International Joint Conference on Artificial Intelligence, 2022.
[51] H. Tran, A. Cutkosky, Better SGD using second-order momentum, Adv. Neural Inf. Process. Syst. 35 (2022) 3530–3541.
[52] T. Vogels, L. He, A. Koloskova, S.P. Karimireddy, T. Lin, S.U. Stich, M. Jaggi, Relaysum for decentralized deep learning on heterogeneous data, Adv. Neural Inf. Process. Syst. 34 (2021).
[53] W. Wan, S. Hu, J. Lu, L.Y. Zhang, H. Jin, Y. He, Shielding federated learning: robust aggregation with adaptive client selection, in: International Joint Conference on Artificial Intelligence, 2022.
[54] J. Wang, V. Tantia, N. Ballas, M.G. Rabbat, Slowmo: improving communication-efficient distributed SGD with slow momentum, in: 8th International Conference on Learning Representations, ICLR, 2020.
[55] Z. Wang, K. Ji, Y. Zhou, Y. Liang, V. Tarokh, Spiderboost and momentum: faster variance reduction algorithms, Adv. Neural Inf. Process. Syst. 32 (2019).
[56] R. Xin, U. Khan, S. Kar, A hybrid variance-reduced method for decentralized stochastic non-convex optimization, in: International Conference on Machine Learning, 2021, pp. 11459–11469.
[57] R. Xin, U.A. Khan, S. Kar, An improved convergence analysis for decentralized online stochastic non-convex optimization, IEEE Trans. Signal Process. 69 (2021) 1842–1858.
[58] H. Xu, C.Y. Ho, A.M. Abdelmoniem, A. Dutta, E. Bergou, K. Karatsenidis, M. Canini, P. Kalnis, GRACE: a compressed communication framework for distributed machine learning, in: Proc. of 41st IEEE Int. Conf. Distributed Computing Systems (ICDCS), 2021.
[59] J. Xu, X. Tong, S.L. Huang, Personalized federated learning with feature alignment and classifier collaboration, in: The Eleventh International Conference on Learning Representations, 2023.
[60] Y. Yan, T. Yang, Z. Li, Q. Lin, Y. Yang, A unified analysis of stochastic momentum methods for deep learning, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 2955–2961.
[61] H. Yu, R. Jin, S. Yang, On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization, in: International Conference on Machine Learning, 2019, pp. 7184–7193.
[62] K. Yuan, S.A. Alghunaim, X. Huang, Removing data heterogeneity influence enhances network topology dependence of decentralized sgd, arXiv preprint arXiv:2105.08023, 2021.
[63] K. Yuan, Y. Chen, X. Huang, Y. Zhang, P. Pan, Y. Xu, W. Yin, Decentlam: decentralized momentum sgd for large-batch deep training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3029–3039.
[64] H. Zhang, J. Liu, J. Jia, Y. Zhou, H. Dai, D. Dou, FedDUAP: federated learning with dynamic update and adaptive pruning using shared data on the server, in: International Joint Conference on Artificial Intelligence, 2022.
[65] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28 (2015).