

MOMENTUM TRACKING: MOMENTUM ACCELERATION FOR DECENTRALIZED DEEP LEARNING ON HETEROGENEOUS DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

SGD with momentum acceleration is one of the key components for improving the performance of neural networks. For decentralized learning, a straightforward approach using momentum acceleration is Distributed SGD (DSGD) with momentum acceleration (DSGDm). However, DSGDm performs worse than DSGD when the data distributions are statistically heterogeneous. Recently, several studies have addressed this issue and proposed methods with momentum acceleration that are more robust to data heterogeneity than DSGDm, although their convergence rates remain dependent on data heterogeneity and decrease when the data distributions are heterogeneous. In this study, we propose Momentum Tracking, which is a method with momentum acceleration whose convergence rate is proven to be independent of data heterogeneity. More specifically, we analyze the convergence rate of Momentum Tracking in the standard deep learning setting, where the objective function is non-convex and the stochastic gradient is used. Then, we identify that it is independent of data heterogeneity for any momentum coefficient $\beta \in [0, 1)$. Through image classification tasks, we demonstrate that Momentum Tracking is more robust to data heterogeneity than the existing decentralized learning methods with momentum acceleration and can consistently outperform these existing methods when the data distributions are heterogeneous.

1 INTRODUCTION

Neural networks have achieved remarkable success in various fields such as image processing (Simonyan & Zisserman, 2015; Chen et al., 2020) and natural language processing (Devlin et al., 2019). To train neural networks, we need to collect large amounts of training data. However, because of privacy concerns, it is often difficult to collect large amounts of data such as medical images on one server. In such scenarios, decentralized learning has attracted significant attention because it allows us to train neural networks without aggregating all the data onto one server. Recently, decentralized learning has been studied from various perspectives, including data heterogeneity (Tang et al., 2018b; Esfandiari et al., 2021), communication compression (Tang et al., 2018a; Lu & De Sa, 2020; Liu et al., 2021; Takezawa et al., 2022a), and network topologies (Ying et al., 2021).

One of the key components for improving the performance of neural networks is SGD with momentum acceleration (SGDm). Whereas SGD updates the model parameters using a stochastic gradient, SGDm updates the model parameters using the moving average of the stochastic gradient, which is called the momentum. Because SGDm can accelerate convergence and improve generalization performance, SGDm has become an indispensable tool, enabling neural networks to achieve high accuracy (He et al., 2016). Recently, SGDm has been improved in many studies, and methods such as Adam (Kingma & Ba, 2015) and RAdam (Liu et al., 2020a) have been proposed.

In decentralized learning, the straightforward approach to using the momentum is Distributed SGD (DSGD) with momentum acceleration (DSGDm) (Gao & Huang, 2020). When the data distributions held by each node (i.e., the server) are statistically homogeneous, DSGDm works well and can improve the performance as well as SGDm (Lin et al., 2021). However, in real-world decentralized learning settings, the data distributions may be heterogeneous (Hsieh et al., 2020). In such cases, DSGDm performs worse than DSGD (i.e., without momentum acceleration) (Yuan et al., 2021).

Table 1: Comparison of the convergence rates. In the “Data-Heterogeneity” column, “ \checkmark ” indicates that the convergence rate is independent of data heterogeneity, and “ (\checkmark) ” indicates that it is independent, but there is no discussion about data heterogeneity either theoretically or experimentally. In the “Momentum,” “Stochastic,” and “Non-Convex” columns, “ \checkmark ” respectively indicates that the method is accelerated using momentum, the convergence rate is provided when the stochastic gradient is used, and the convergence rate is provided when the objective function is non-convex.

	Data-Heterogeneity	Momentum	Stochastic	Non-Convex
DSGD (Lian et al., 2017)			\checkmark	\checkmark
Gradient Tracking (Koloskova et al., 2021)	\checkmark		\checkmark	\checkmark
DSGDm (Gao & Huang, 2020)		\checkmark	\checkmark	\checkmark
QG-DSGDm (Lin et al., 2021)		\checkmark	\checkmark	\checkmark
DecentLaM (Yuan et al., 2021)		\checkmark	\checkmark	\checkmark
ABm (Xin & Khan, 2020)	(\checkmark)	\checkmark		
GTAdam (Carnevale et al., 2022)	(\checkmark)	\checkmark		
Momentum Tracking (our work)	\checkmark	\checkmark	\checkmark	\checkmark

This is because, when the data distributions are heterogeneous and we use the momentum instead of the stochastic gradient, each model parameter is updated in further different directions and drifts away more easily. As a result, the convergence rate of DSGDm falls below that of DSGD. To address this issue, Lin et al. (2021) and Yuan et al. (2021) modified the update rules of the momentum in DSGDm and proposed methods that are more robust to data heterogeneity than DSGDm. However, their convergence rates remain dependent on data heterogeneity, and our experiments revealed that their performance are degraded when the data distributions are strongly heterogeneous (Sec. 4).

Data heterogeneity for decentralized learning has been well studied from both experimental and theoretical perspectives (Hsieh et al., 2020; Koloskova et al., 2020). Subsequently, many methods including Gradient Tracking (Lorenzo & Scutari, 2016; Nedić et al., 2017) have been proposed and it has been shown that their convergence rates do not depend on data heterogeneity (Tang et al., 2018b; Vogels et al., 2021; Koloskova et al., 2021). However, these studies considered only the case where the momentum was not used, and it remains unclear whether these methods are robust to data heterogeneity when the momentum is applied.

In the convex optimization literature, Xin & Khan (2020) and Carnevale et al. (2022) proposed combining Gradient Tracking with momentum or Adam and analyzed the convergence rates. However, they considered only the case where the objective function is strongly convex and the full gradient is used, which does not hold in the standard deep learning setting, where the objective function is non-convex and only the stochastic gradient is accessible. Hence, their convergence rates are still unknown in standard deep learning settings, and it remains unclear whether their convergence rates are independent of data heterogeneity. Furthermore, they did not discuss data heterogeneity, either theoretically or experimentally.

In this work, we propose a decentralized learning method with momentum acceleration, which we call **Momentum Tracking**, whose convergence rate is proven to be independent of data heterogeneity in the standard deep learning setting. More specifically, we provide the convergence rate of Momentum Tracking in a setting in which the objective function is non-convex and the stochastic gradient is used. Then, we identify that the convergence rate of Momentum Tracking is independent of data heterogeneity for any momentum coefficient $\beta \in [0, 1)$. In Table 1, we compare the convergence rate of Momentum Tracking with those of existing methods. To the best of our knowledge, Momentum Tracking is the first decentralized learning method with momentum acceleration whose convergence rate has been proven to be independent of data heterogeneity in the standard deep learning setting. Experimentally, we demonstrate that Momentum Tracking is more robust to data heterogeneity than the existing decentralized learning methods with momentum acceleration and can consistently outperform these existing methods when the data distributions are heterogeneous.

2 PRELIMINARIES AND RELATED WORK

2.1 DECENTRALIZED LEARNING

Let $G = (V, E)$ be an undirected graph that represents the underlying network topology, where V denotes the set of nodes and E denotes the set of edges. Let $N := |V|$ be the number of nodes, and we label each node in V by a set of integers $\{1, 2, \dots, N\}$ for simplicity. We define $\mathcal{N}_i := \{j \in$

$V \mid (i, j) \in E$ as the set of neighbor nodes of node i and define $\mathcal{N}_i^+ := \mathcal{N}_i \cup \{i\}$. In decentralized learning, node i has a local data distribution \mathcal{D}_i and local objective function $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$, and can communicate with node j if and only if $(i, j) \in E$. Then, decentralized learning aims to minimize the average of the local objective functions as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}) \right], \quad f_i(\mathbf{x}) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} [F_i(\mathbf{x}; \xi_i)],$$

where \mathbf{x} is the model parameter, ξ_i is the data sample that follows \mathcal{D}_i , and local objective function $f_i(\mathbf{x})$ is defined as the expectation of $F_i(\mathbf{x}; \xi_i)$ over data sample ξ_i . In the following, $\nabla F_i(\mathbf{x}; \xi_i)$ and $\nabla f_i(\mathbf{x}) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} [\nabla F_i(\mathbf{x}; \xi_i)]$ denote the stochastic and full gradient respectively.

Distributed SGD (DSGD) (Lian et al., 2017) is one of the most well-known algorithms for decentralized learning. Formally, the update rules of DSGD are defined as follows:

$$\mathbf{x}_i^{(r+1)} = \sum_{j \in \mathcal{N}_i^+} W_{ij} \left(\mathbf{x}_j^{(r)} - \eta \nabla F_j(\mathbf{x}_j^{(r)}; \xi_j^{(r)}) \right), \quad (1)$$

where $\eta > 0$ is the step size and $W_{ij} \in [0, 1]$ is the weight of edge (i, j) . Let $\mathbf{W} \in [0, 1]^{N \times N}$ be the matrix whose (i, j) -element is W_{ij} if $(i, j) \in E$ and 0 otherwise. In general, a mixing matrix is used for \mathbf{W} (i.e., $\mathbf{W} = \mathbf{W}^\top$, $\mathbf{W}\mathbf{1} = \mathbf{1}$, and $\mathbf{W}^\top \mathbf{1} = \mathbf{1}$). Lian et al. (2018) extended DSGD in the case where each node communicates asynchronously and analyzed the convergence rate. Koloskova et al. (2020) analyzed the convergence rate of DSGD when the network topology changes over time. These results revealed that the convergence rate of DSGD decreases and the performance is degraded when the data distributions held by each node are statistically heterogeneous. This is because the local gradients ∇f_i are different across nodes and each model parameter \mathbf{x}_i tends to drift away when the data distributions are heterogeneous. To address this issue, D^2 (Tang et al., 2018b), Gradient Tracking (Lorenzo & Scutari, 2016; Nedić et al., 2017), and primal-dual algorithms (Niwa et al., 2020; 2021; Takezawa et al., 2022b) were proposed to correct the local gradient ∇f_i to the global gradient ∇f . As a different approach, Vogels et al. (2021) proposed a novel averaging method to prevent each model parameter \mathbf{x}_i from drifting away. It has been shown that the convergence rates of these methods do not depend on data heterogeneity and do not decrease, even when the data distributions held by each node are statistically heterogeneous. However, these methods do not consider the case in which momentum is used.

2.2 MOMENTUM ACCELERATION

The methods with momentum acceleration were originally proposed by Polyak (1964), and SGD with momentum acceleration (SGDm) has achieved successful results in training neural networks (Simonyan & Zisserman, 2015; He et al., 2016; Wang et al., 2020b). In decentralized learning, a straightforward approach to using the momentum is DSGD with momentum acceleration (DSGDm) (Gao & Huang, 2020). The update rules of DSGDm are defined as follows:

$$\mathbf{u}_i^{(r+1)} = \beta \mathbf{u}_i^{(r)} + \nabla F_i(\mathbf{x}_i^{(r)}; \xi_i^{(r)}), \quad (2)$$

$$\mathbf{x}_i^{(r+1)} = \sum_{j \in \mathcal{N}_i^+} W_{ij} \left(\mathbf{x}_j^{(r)} - \eta \mathbf{u}_j^{(r+1)} \right), \quad (3)$$

where \mathbf{u}_i is the local momentum of node i and $\beta \in [0, 1)$ is a momentum coefficient. In addition, several variants of DSGDm were studied by Yu et al. (2019); Assran et al. (2019); Wang et al. (2020a); Singh et al. (2021). When the data distributions held by each node are statistically homogeneous, DSGDm works well and can improve the performance as well as SGDm. However, when the data distributions are statistically heterogeneous, DSGDm leads to poorer performance than DSGD. This is because when the data distributions held by each node are statistically heterogeneous (i.e., ∇f_i varies significantly across nodes), the difference in the updated value of the model parameter across the nodes (i.e., $\eta \mathbf{u}_i$) is amplified by the momentum (Lin et al., 2021).

To address this issue, Yuan et al. (2021) and Lin et al. (2021) proposed methods to modify the update rules of the momentum in DSGDm, called DecentLaM and QG-DSGDm, respectively. They further experimentally demonstrated that these methods are more robust to data heterogeneity than DSGDm. However, their convergence rates have been shown to still depend on data heterogeneity and decrease when the data distributions are heterogeneous.

2.3 GRADIENT TRACKING

One of the most well-known methods whose convergence rate does not depend on data heterogeneity is Gradient Tracking (Lorenzo & Scutari, 2016). Whereas DSGD exchanges only the model parameter \mathbf{x}_i , Gradient Tracking exchanges the model parameter \mathbf{x}_i and local (stochastic) gradient ∇f_i and then updates the model parameters while estimating global gradient ∇f . Nedić et al. (2017) and Qu & Li (2018) analyzed the convergence rate of Gradient Tracking when the objective function is (strongly) convex and the full gradient is used. Pu & Nedic (2021) analyzed the convergence rate when the objective function is strongly convex and the stochastic gradient is used. Recently, Koloskova et al. (2021) analyzed the convergence rates of Gradient Tracking in a standard deep learning setting, where the objective function is non-convex and the stochastic gradient is used. There is also a line of research to combine Gradient Tracking with variance reduction methods (Xin et al., 2022). They showed that the convergence rate of Gradient Tracking does not depend on data heterogeneity. However, these studies only consider the case without momentum acceleration, and the convergence analysis for Gradient Tracking with momentum acceleration has not been explored thus far in the aforementioned studies.

In the convex optimization literature, Xin & Khan (2020) and Carnevale et al. (2022) proposed a combination of Gradient Tracking and the momentum or Adam (Kingma & Ba, 2015). However, they only considered the case where the objective function is strongly convex and the full gradient is used. The convergence rate is still unclear in the standard deep learning setting, where the objective function is non-convex and the stochastic gradient is used. Furthermore, there is no discussion about data heterogeneity in these studies, either theoretically or experimentally.

3 PROPOSED METHOD

In this section, we propose **Momentum Tracking**, which is a decentralized learning method with momentum acceleration whose convergence rate is proven to be independent of the data heterogeneity in the standard deep learning setting.

3.1 SETUP

We assume that the following four standard assumptions hold:

Assumption 1. *There exists a constant $f^* > -\infty$ that satisfies $f(\mathbf{x}) \geq f^*$ for all $\mathbf{x} \in \mathbb{R}^d$.*

Assumption 2. *There exists a constant $p \in (0, 1]$ that satisfies for all $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^d$,*

$$\|\mathbf{X}\mathbf{W} - \bar{\mathbf{X}}\|_F^2 \leq (1-p)\|\mathbf{X} - \bar{\mathbf{X}}\|_F^2, \quad (4)$$

where $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{d \times N}$ and $\bar{\mathbf{X}} := \frac{1}{N}\mathbf{X}\mathbf{1}\mathbf{1}^\top$.

Assumption 3. *There exists a constant $L > 0$ that satisfies for all $i \in V$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \quad (5)$$

Assumption 4. *There exists a constant σ^2 that satisfies for all $i \in V$ and $\mathbf{x}_i \in \mathbb{R}^d$,*

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \|\nabla F_i(\mathbf{x}_i; \xi_i) - \nabla f_i(\mathbf{x}_i)\|^2 \leq \sigma^2. \quad (6)$$

Assumptions 1, 2, 3, and 4 are commonly used for decentralized learning algorithms (Lian et al., 2017; Yu et al., 2019; Koloskova et al., 2021; Lin et al., 2021). Additionally, the following assumption, which represents data heterogeneity, is commonly used in the convergence analysis of decentralized learning algorithms (Lian et al., 2017; Yu et al., 2019; Lin et al., 2021).

Assumption 5. *There exists a constant ζ^2 that satisfies for all $\mathbf{x} \in \mathbb{R}^d$,*

$$\frac{1}{N} \sum_{i=1}^N \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \zeta^2.$$

Under Assumption 5, the convergence rates of DSGD (Lian et al., 2017), DSGDm (Gao & Huang, 2020; Yuan et al., 2021), QG-DSGDm (Lin et al., 2021), and DecentLaM (Yuan et al., 2021) were

shown to be dependent on data heterogeneity ζ^2 and decrease as ζ^2 increases. By contrast, in Sec. 3.3, we prove that Momentum Tracking converges without Assumption 5 and the convergence rate is independent of data heterogeneity ζ^2 . In addition, we do not assume the convexity of the objective functions $f(\mathbf{x})$ and $f_i(\mathbf{x})$. Therefore, $f(\mathbf{x})$ and $f_i(\mathbf{x})$ are potentially non-convex functions (e.g., the loss functions of neural networks).

3.2 MOMENTUM TRACKING

In this section, we propose **Momentum Tracking**, which is robust to data heterogeneity and accelerated by the momentum. The update rules of Momentum Tracking are defined as follows:

$$\mathbf{u}_i^{(r+1)} = \beta \mathbf{u}_i^{(r)} + \nabla F_i(\mathbf{x}_i^{(r)}; \xi_i^{(r)}), \quad (7)$$

$$\mathbf{x}_i^{(r+1)} = \sum_{j \in \mathcal{N}_i^+} W_{ij} \mathbf{x}_j^{(r)} - \eta \left(\mathbf{u}_i^{(r+1)} - \mathbf{c}_i^{(r)} \right), \quad (8)$$

$$\mathbf{c}_i^{(r+1)} = \sum_{j \in \mathcal{N}_i^+} W_{ij} \left(\mathbf{c}_j^{(r)} - \mathbf{u}_j^{(r+1)} \right) + \mathbf{u}_i^{(r+1)}, \quad (9)$$

where $\beta \in [0, 1)$ is a momentum coefficient. The pseudo-code for Momentum Tracking is presented in Sec. A. In Momentum Tracking, \mathbf{c}_i corrects the local momentum \mathbf{u}_i to the global momentum $\frac{1}{N} \sum_j \mathbf{u}_j$ and prevents each model parameter \mathbf{x}_i from drifting, even when the data distributions are statistically heterogeneous (i.e., the local momentum \mathbf{u}_i varies significantly across nodes).

Because Momentum Tracking is equivalent to Gradient Tracking when $\beta = 0$, Momentum Tracking is a simple extension of Gradient Tracking. Hence, when $\beta = 0$, it has been shown that the convergence rate of Momentum Tracking is independent of data heterogeneity ζ^2 (Koloskova et al., 2021). However, because data heterogeneity is amplified when the momentum is used instead of the stochastic gradient (i.e., $\beta > 0$) (Lin et al., 2021; Yuan et al., 2021), it is unclear whether the convergence rate of Momentum Tracking is independent of data heterogeneity ζ^2 for any $\beta \in [0, 1)$ or for only a restricted range of β . In Sec. 3.3, we provide the convergence rate of Momentum Tracking and prove that it is independent of data heterogeneity ζ^2 for any $\beta \in [0, 1)$.

3.3 CONVERGENCE ANALYSIS

Under Assumptions 1, 2, 3, and 4, Theorem 1 provides the convergence rate of Momentum Tracking in the standard deep learning setting. All proofs are presented in Sec. D.

Theorem 1 (Convergence Rate in Non-Convex Setting). *Suppose that Assumptions 1, 2, 3, and 4 hold, each model parameter \mathbf{x}_i is initialized with the same parameters, and both \mathbf{u}_i and \mathbf{c}_i are initialized as $\frac{1}{1-\beta} (\nabla F_i(\mathbf{x}_i^{(0)}; \xi_i^{(0)}) - \frac{1}{N} \sum_{j=1}^N \nabla F_j(\mathbf{x}_j^{(0)}; \xi_j^{(0)}))$. Then, for any $\beta \in [0, 1)$ and $R \geq 1$, there exists a step size η such that the average parameter $\bar{\mathbf{x}} := \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ generated by Eqs. (7-9) satisfies*

$$\begin{aligned} & \frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left\| \nabla f(\bar{\mathbf{x}}^{(r)}) \right\|^2 \\ & \leq \mathcal{O} \left(\sqrt{\frac{r_0 \sigma^2 L}{NR}} + \left(\frac{r_0^2 \sigma^2 L^2}{p^4 R^2 (1-\beta)} \left(1 + \frac{p\beta^2}{1-\beta} \right) \right)^{\frac{1}{3}} + \frac{Lr_0}{(1-\beta)p^2 R} \sqrt{1 + \frac{\beta^2}{(1-\beta)^3 p}} \right), \end{aligned} \quad (10)$$

where $r_0 := f(\bar{\mathbf{x}}^{(0)}) - f^*$.

Remark 1. *Combinations of Gradient Tracking with the momentum or Adam have also been proposed by Xin & Khan (2020) and Carnevale et al. (2022). However, they considered only the setting in which the objective function is strongly convex and the full gradient is used. By contrast, our study focuses on the deep learning setting. Hence, our proof strategies are completely different from those in these previous studies, and Theorem 1 provides the convergence rate in the setting where the objective function is non-convex and the stochastic gradient is used.*

Remark 2. *The convergence rate of Gradient Tracking in the standard deep learning setting was provided by Koloskova et al. (2020). However, they did not consider the case where the momentum is used, and it is not trivial to provide the convergence rate of Momentum Tracking from the results in this previous work.*

3.4 DISCUSSION

Comparison with Gradient Tracking: Theorem 1 indicates that the convergence rate of Momentum Tracking does not depend on data heterogeneity ζ^2 for any $\beta \in [0, 1)$ and does not decrease even when the data distributions are statistically heterogeneous (i.e., $\zeta^2 > 0$). Therefore, Theorem 1 indicates that Momentum Tracking is theoretically robust to data heterogeneity for any $\beta \in [0, 1)$. Although Momentum Tracking is a simple extension of Gradient Tracking, our work is the first to identify that the combination of Gradient Tracking and the momentum converges without being affected by data heterogeneity ζ^2 for any $\beta \in [0, 1)$ in the standard deep learning setting.

Because the convergence rate of Momentum Tracking Eq. (10) is optimal when $\beta = 0$, Theorem 1 does not show that the convergence rate is improved by using the momentum. However, the convergence rates of DSGDm and QG-DSGDm provided by Gao & Huang (2020) and Lin et al. (2021) are also optimal when $\beta = 0$. Moreover, they do not provide theoretical results that are consistent with the experimental results that the convergence rates are improved when $\beta > 0$. As in these studies, we experimentally demonstrate that convergence is accelerated when $\beta > 0$ in Sec. 4 and leave for future work to show the theoretical benefits of using $\beta > 0$.

Comparison with Existing Algorithms with Momentum Acceleration: Next, we compare the convergence rate of Momentum Tracking with those of existing decentralized learning algorithms with momentum acceleration: DSGDm (Gao & Huang, 2020), DecentLaM (Yuan et al., 2021), and QG-DSGDm (Lin et al., 2021). Here, we only show the convergence rate of QG-DSGDm, but the same discussion holds for the other methods. The convergence rate of QG-DSGDm is as follows:

Theorem 2 ((Lin et al., 2021)). *Suppose that Assumptions 1, 2, 3, and 4 hold, and Assumption 5 also holds. Then, for any $\beta \in [0, \frac{p}{21+p}]$ and $R \geq 1$, there exists a step size η such that the average parameter $\bar{\mathbf{x}} := \frac{1}{N} \sum_i \mathbf{x}_i$ generated by QG-DSGDm satisfies¹*

$$\frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left\| \nabla f(\bar{\mathbf{x}}^{(r)}) \right\|^2 \leq \mathcal{O} \left(\sqrt{\frac{r_0 \sigma^2 L}{NR}} + \left(\frac{r_0^2 L^2 (\zeta^2 + \sigma^2)}{p^2 R^2} \right)^{\frac{1}{3}} + \frac{Lr_0}{R} \left(\frac{1}{p} + \frac{1}{1-\beta} + \frac{\beta}{(1-\beta)^3} \right) \right),$$

where $r_0 := f(\bar{\mathbf{x}}^{(0)}) - f^*$.

Data heterogeneity ζ^2 appears in the second term, and the convergence rate of QG-DSGDm depends on data heterogeneity ζ^2 . Therefore, the convergence rate of QG-DSGDm decreases when the data distributions held by each node are statistically heterogeneous. By contrast, the convergence rate of Momentum Tracking Eq. (10) does not depend on data heterogeneity ζ^2 . Therefore, Momentum Tracking is more robust to data heterogeneity than QG-DSGDm. Because the convergence rates of DSGDm and DecentLaM also depend on data heterogeneity ζ^2 , the same discussion holds for DSGDm and DecentLaM. Hence, Momentum Tracking is more robust to data heterogeneity than these methods. To the best of our knowledge, Momentum Tracking is the first decentralized learning method with momentum acceleration whose convergence rate has been proven to be independent of data heterogeneity ζ^2 in the standard deep learning setting.

Next, we discuss the range of β . The convergence rates of QG-DSGDm and DecentLaM provided by Lin et al. (2021) and Yuan et al. (2021) hold only when the range of β is restricted. For instance, Theorem 2 assumes that $\beta \leq \frac{p}{21+p} (< 0.05)$. However, these restrictions on the range of β do not hold in practice. (Typically, β is set to 0.9.) Therefore, the convergence rates of QG-DSGDm and DecentLaM are unclear in such practical cases. By contrast, Theorem 1 can provide the convergence rate of Momentum Tracking that holds for any $\beta \in [0, 1)$.

Comparison with SGDm: Next, we compare the convergence rate of Momentum Tracking with that of SGDm. In a setting where the objective function is non-convex and the stochastic gradient is used, SGDm has been proven to converge to the stationary point with $\mathcal{O}(1/\sqrt{R})$ (Yan et al., 2018; Liu et al., 2020b). By contrast, Theorem 1 indicates that if the number of rounds R is sufficiently large, Momentum Tracking converges with $\mathcal{O}(1/\sqrt{NR})$. Therefore, Momentum Tracking can achieve a linear speedup with respect to the number of nodes N , which is a common and important property in decentralized learning methods (Lian et al., 2018; Koloskova et al., 2020).

¹For simplicity, we set the additional hyperparameter μ for QG-DSGDm to β .

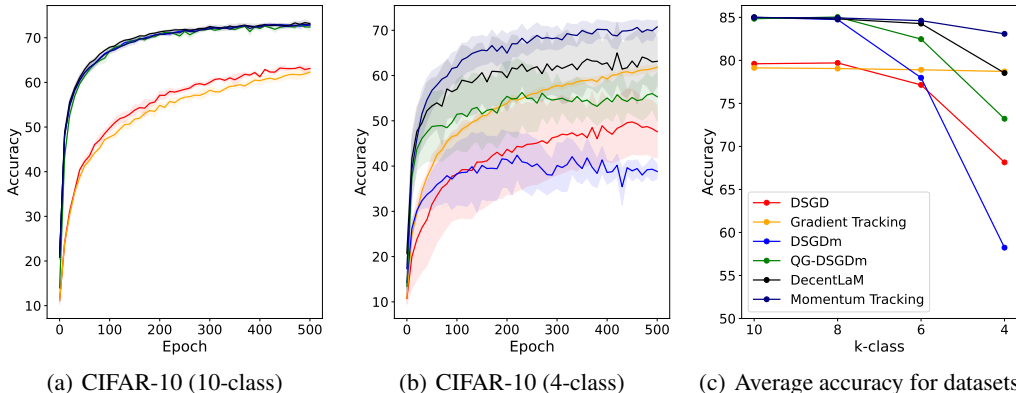


Figure 1: (a) Learning curve on CIFAR-10 with LeNet in the 10-class (i.e., homogeneous) setting. We evaluated the test accuracy per 10 epochs. (b) Learning curve in the 4-class (i.e., heterogeneous) setting. (c) Average test accuracy for all datasets (i.e., FashionMNIST, SVHN, and CIFAR-10).

4 EXPERIMENT

In this section, we present the results of an experimental evaluation of Momentum Tracking and demonstrate that Momentum Tracking is more robust to data heterogeneity than the existing decentralized learning methods with momentum acceleration. In this section, we focus on test accuracy, and more detailed evaluation about the convergence rate is presented in Sec. C.4.

4.1 SETUP

Comparison Methods: (1) DSGD (Lian et al., 2017): the method described in Sec. 2.1; (2) DSGDm (Gao & Huang, 2020): the method described in Sec. 2.2; (3) QG-DSGDm (Lin et al., 2021): a method in which the update rule of the momentum in DSGDm is modified to be more robust to data heterogeneity than DSGDm; (4) DecentLaM (Yuan et al., 2021): a method in which the update rule of the momentum in DSGDm is modified to be more robust to data heterogeneity; (5) Gradient Tracking (Nedić et al., 2017): a method without momentum acceleration that is robust to data heterogeneity; (6) Momentum Tracking: the proposed method described in Sec. 3.

Dataset and Model: We evaluated Momentum Tracking using three 10-class image classification tasks: FashionMNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10 (Krizhevsky, 2009). Following the previous work (Niwa et al., 2020), we distributed the data to nodes such that each node was given data of randomly selected k classes. When $k = 10$, the data distributions held by each node can be regarded as statistically homogeneous. When $k < 10$, the data distributions are regarded as statistically heterogeneous. We evaluated the comparison methods by setting k to $\{4, 6, 8, 10\}$ and changing data heterogeneity. Note that a smaller k indicates that the data distributions are more heterogeneous. For the neural network architecture, we used LeNet (LeCun et al., 1998) with group normalization (Wu & He, 2018) in Sec. 4.2. In Sec. 4.3, we present more detailed evaluation by varying the neural network architecture (e.g., VGG-11 (Simonyan & Zisserman, 2015) and ResNet-34 (He et al., 2016)). For each comparison method, we used 10% of the training data for validation and individually tuned the step size. For DSGDm, QG-DSGDm, DecentLaM, and Momentum Tracking, we set β to 0.9. All experiments were repeated using three different seed values, and we report their averages. More detailed hyperparameter settings are presented in Sec E.

Network Topology and Implementation: In Secs. 4.2 and 4.3, we present the results of setting the underlying network topology to a ring consisting of eight nodes (i.e., $N = 8$). In Sec. C.1, we present more detailed evaluation by varying the network topology. All comparison methods were implemented using PyTorch and run on eight GPUs (NVIDIA RTX 3090).

4.2 EXPERIMENTAL RESULTS

Table 2 lists the test accuracy for FashionMNIST, SVHN, and CIFAR-10. Fig. 1 (a) and (b) present the learning curves for CIFAR-10 and Fig. 1 (c) presents the average test accuracy for all datasets.

Table 2: Test accuracy on FashionMNIST, SVHN, and CIFAR-10 with LeNet. “ k -class” means that each node has only the data of randomly selected k classes. Bold font means the highest accuracy.

FashionMNIST				
	10-class	8-class	6-class	4-class
DSGD	85.6 \pm 0.49	85.6 \pm 0.41	82.7 \pm 1.12	78.1 \pm 1.56
Gradient Tracking	85.0 \pm 0.49	85.4 \pm 0.26	85.0 \pm 0.37	84.9 \pm 0.22
DSGDm	89.5 \pm 0.15	89.3 \pm 0.21	82.1 \pm 3.23	68.7 \pm 5.02
QG-DSGDm	89.6 \pm 0.10	89.5 \pm 0.47	86.9 \pm 1.59	80.8 \pm 2.94
DecentLaM	89.5 \pm 0.14	89.3 \pm 0.36	89.2 \pm 0.41	84.3 \pm 3.05
Momentum Tracking	89.5 \pm 0.36	89.4 \pm 0.05	88.9 \pm 0.47	86.8 \pm 1.56

SVHN				
	10-class	8-class	6-class	4-class
DSGD	90.1 \pm 0.17	89.5 \pm 0.61	87.6 \pm 1.94	78.8 \pm 8.55
Gradient Tracking	90.1 \pm 0.30	89.8 \pm 0.38	89.8 \pm 0.39	89.4 \pm 0.47
DSGDm	92.6 \pm 0.35	92.4 \pm 0.19	88.1 \pm 4.38	67.2 \pm 9.69
QG-DSGDm	92.5 \pm 0.22	92.5 \pm 0.17	90.9 \pm 1.67	83.5 \pm 7.14
DecentLaM	92.4 \pm 0.21	92.2 \pm 0.39	92.0 \pm 0.48	88.2 \pm 4.75
Momentum Tracking	92.6 \pm 0.32	92.4 \pm 0.40	92.3 \pm 0.23	91.7 \pm 0.53

CIFAR-10				
	10-class	8-class	6-class	4-class
DSGD	63.1 \pm 0.60	64.1 \pm 0.52	61.2 \pm 1.16	47.6 \pm 5.77
Gradient Tracking	62.3 \pm 0.73	62.0 \pm 0.80	61.9 \pm 0.58	61.8 \pm 0.82
DSGDm	72.9 \pm 0.41	72.5 \pm 0.20	63.8 \pm 6.24	38.8 \pm 1.61
QG-DSGDm	72.4 \pm 0.87	73.1 \pm 0.16	69.6 \pm 2.42	55.3 \pm 5.30
DecentLaM	73.2 \pm 0.36	72.9 \pm 0.14	71.7 \pm 1.10	63.1 \pm 5.43
Momentum Tracking	72.9 \pm 0.59	73.0 \pm 0.49	72.6 \pm 0.41	70.7 \pm 1.38

Comparison of Momentum Tracking and Gradient Tracking: First, we discuss the results of Momentum Tracking and Gradient Tracking. Table 2 and Fig. 1 indicate that Momentum Tracking achieves a higher accuracy faster than Gradient Tracking and outperforms Gradient Tracking in all settings. When the data distributions are homogeneous (i.e., 10-class), Momentum Tracking outperforms Gradient Tracking by 5.8% on average. When the data distributions are heterogeneous (e.g., 4-class), Momentum Tracking outperforms Gradient Tracking by 4.4% on average. Therefore, the results show that Momentum Tracking can consistently outperform Gradient Tracking regardless of data heterogeneity.

Comparison of Momentum Tracking and DSGDm: Next, we discuss the results of Momentum Tracking and DSGDm. The results show that when the data distributions are homogeneous (i.e., 10-class), Momentum Tracking and DSGDm are comparable and outperform DSGD and Gradient Tracking. However, when the data distributions are heterogeneous (e.g., 4-class), the test accuracy of DSGDm decreases even more than that of DSGD, and DSGDm underperforms DSGD by 9.9% on average. By contrast, the results indicate that Momentum Tracking consistently outperforms DSGD and Gradient Tracking by 14.9% and 4.4% respectively when the data distributions are heterogeneous. The results indicate that Momentum Tracking is more robust to data heterogeneity than DSGDm and outperforms DSGDm by 24.9% on average.

Comparison of Momentum Tracking, QG-DSGDm, and DecentLaM: When the data distributions are homogeneous (i.e., 10-class), Momentum Tracking, QG-DSGDm, and DecentLaM are comparable and outperform DSGD and Gradient Tracking. By contrast, when the data distributions are heterogeneous (e.g., 4-class), Momentum Tracking consistently outperforms QG-DSGDm and DecentLaM by 9.9% and 4.5% respectively, whereas QG-DSGDm and DecentLaM are more robust to data heterogeneity than DSGDm. Hence, these results are consistent with our theoretical analysis, as discussed in Secs. 3.3 and 3.4.

Table 3: Test accuracy on CIFAR-10 with VGG-11 and ResNet-34. “ k -class” indicates that each node has only the data of randomly selected k classes, and bold font indicates the highest accuracy.

	CIFAR-10 + VGG-11			CIFAR-10 + ResNet-34		
	10-class	4-class	2-class	10-class	4-class	2-class
DSGD	89.7 ± 0.15	85.8 ± 2.39	70.3 ± 2.73	93.4 ± 0.05	87.5 ± 3.27	64.1 ± 0.54
Gradient Tracking	88.1 ± 0.19	86.0 ± 0.45	82.9 ± 0.13	85.1 ± 1.07	81.2 ± 0.73	75.9 ± 0.41
DSGDm	92.2 ± 0.09	77.3 ± 4.05	39.6 ± 5.92	95.8 ± 0.26	79.0 ± 3.69	27.7 ± 2.83
QG-DSGDm	92.0 ± 0.04	89.4 ± 1.04	77.8 ± 1.96	95.6 ± 0.37	94.0 ± 1.02	77.4 ± 3.13
DecentLaM	92.1 ± 0.09	90.9 ± 0.65	85.2 ± 0.67	95.9 ± 0.04	95.2 ± 0.51	89.2 ± 2.26
Momentum Tracking	91.9 ± 0.06	90.9 ± 0.60	87.0 ± 0.48	95.0 ± 0.13	94.4 ± 0.52	89.9 ± 0.73

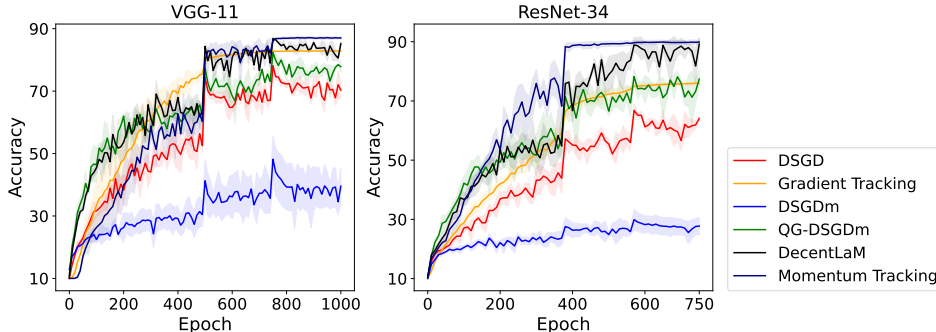


Figure 2: Learning curves for CIFAR-10 with VGG-11 and ResNet-34 in the 2-class setting.

In summary, when the data distributions are homogeneous, DSGDm, QG-DSGDm, DecentLaM, and Momentum Tracking are comparable and outperform DSGD and Gradient Tracking. When the data distributions are heterogeneous, Momentum Tracking is more robust to data heterogeneity than DSGDm, QG-DSGDm, and DecentLaM, and can outperform all comparison methods.

4.3 RESULTS WITH VARIOUS NEURAL NETWORK ARCHITECTURES

Next, we evaluated Momentum Tracking in more detail by varying the neural network architecture. Table 3 lists the test accuracy with VGG-11 (Simonyan & Zisserman, 2015) and ResNet-34 (He et al., 2016) when we set k to $\{2, 4, 10\}$, and Fig. 2 shows the learning curves.

For both neural network architectures, Table 3 reveals that when the data distributions are homogeneous (i.e., 10-class), Momentum Tracking is comparable with DSGDm, QG-DSGDm, and DecentLaM and outperforms DSGD and Gradient Tracking. By contrast, when the data distributions are heterogeneous (e.g., 2-class), Table 3 and Fig. 2 reveal that Momentum Tracking outperforms all comparison methods for both neural network architectures. In particular, Fig. 2 indicates that DSGDm, QG-DSGDm, and DecentLaM are unstable and continue to oscillate in the final training phase, whereas Momentum Tracking converges stably. These results are consistent with those of LeNet presented in Table 2. Therefore, the results indicate that Momentum Tracking is more robust to data heterogeneity than DSGDm, QG-DSGDm, and DecentLaM, and can outperform these methods regardless of the neural network architecture.

5 CONCLUSION

In this study, we propose Momentum Tracking, which is a method with momentum acceleration whose convergence rate is proven to be independent of data heterogeneity. More specifically, we provide the convergence rate of Momentum Tracking in the standard deep learning setting, in which the objective function is non-convex and the stochastic gradient is used. Our theoretical analysis reveals that the convergence rate of Momentum Tracking is independent of data heterogeneity for any $\beta \in [0, 1)$. Through image classification tasks, we demonstrated that Momentum Tracking can consistently outperform the decentralized learning methods without momentum acceleration regardless of data heterogeneity. Moreover, we showed that Momentum Tracking is more to data heterogeneity than existing decentralized learning methods with momentum acceleration and can consistently outperform these existing methods when the data distributions are heterogeneous.

REFERENCES

- Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, 2019.
- Guido Carnevale, Francesco Farina, Ivano Notarnicola, and Giuseppe Notarstefano. Gtadam: Gradient tracking with adaptive momentum for distributed online optimization. In *arXiv*, 2022.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.
- Yasaman Esfandiari, Sin Yong Tan, Zhanhong Jiang, Aditya Balu, Ethan Herron, Chinmay Hegde, and Soumik Sarkar. Cross-gradient aggregation for decentralized learning from non-iid data. In *International Conference on Machine Learning*, 2021.
- Hongchang Gao and Heng Huang. Periodic stochastic gradient descent with momentum for decentralized training. In *arXiv*, 2020.
- Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-IID data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, 2020.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, 2020.
- Anastasia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. In *Advances in Neural Information Processing Systems*, 2021.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE*, 1998.
- Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2017.
- Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, 2018.
- Tao Lin, Sai Praneeth Karimireddy, Sebastian Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *International Conference on Machine Learning*, 2021.
- Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020a.
- Xiaorui Liu, Yao Li, Rongrong Wang, Jiliang Tang, and Ming Yan. Linear convergent decentralized optimization with compression. In *International Conference on Learning Representations*, 2021.
- Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. In *Advances in Neural Information Processing Systems*, 2020b.

- Paolo Di Lorenzo and Gesualdo Scutari. NEXT: in-network nonconvex optimization. In *IEEE Transactions on Signal and Information Processing over Networks*, 2016.
- Yucheng Lu and Christopher De Sa. Moniqua: Modulo quantized communication in decentralized SGD. In *International Conference on Machine Learning*, 2020.
- Angelia Nedić, Alexander Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. In *SIAM Journal on Optimization*, 2017.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop*, 2011.
- Kenta Niwa, Noboru Harada, Guoqiang Zhang, and W. Bastiaan Kleijn. Edge-consensus learning: Deep learning on p2p networks with nonhomogeneous data. In *International Conference on Knowledge Discovery and Data Mining*, 2020.
- Kenta Niwa, Guoqiang Zhang, W. Bastiaan Kleijn, Noboru Harada, Hiroshi Sawada, and Akinori Fujino. Asynchronous decentralized optimization with implicit stochastic variance reduction. In *International Conference on Machine Learning*, 2021.
- Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. In *USSR Computational Mathematics and Mathematical Physics*, 1964.
- Shi Pu and Angelia Nedic. Distributed stochastic gradient tracking methods. In *Math. Program.*, 2021.
- Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. In *IEEE Transactions on Control of Network Systems*, 2018.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- Navjot Singh, Deepesh Data, Jemin George, and Suhas Diggavi. Squarm-sgd: Communication-efficient momentum sgd for decentralized optimization. In *IEEE International Symposium on Information Theory*, 2021.
- Yuki Takezawa, Kenta Niwa, and Makoto Yamada. Communication compression for decentralized learning with operator splitting methods. In *arXiv*, 2022a.
- Yuki Takezawa, Kenta Niwa, and Makoto Yamada. Theoretical analysis of primal-dual algorithm for non-convex stochastic decentralized optimization. In *arXiv*, 2022b.
- Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, 2018a.
- Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2 : Decentralized training over decentralized data. In *International Conference on Machine Learning*, 2018b.
- Thijs Vogels, Lie He, Anastasia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U Stich, and Martin Jaggi. RelaySum for decentralized deep learning on heterogeneous data. In *Advances in Neural Information Processing Systems*, 2021.
- Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*, 2020a.
- Jun-Kun Wang, Chi-Heng Lin, and Jacob Abernethy. Escaping saddle points faster with stochastic momentum. In *International Conference on Learning Representations*, 2020b.
- Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, 2018.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. In *arXiv*, 2017.

- Ran Xin and Usman A. Khan. Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking. In *IEEE Transactions on Automatic Control*, 2020.
- Ran Xin, Usman A. Khan, and Soumya Kar. Fast decentralized nonconvex finite-sum optimization with recursive variance reduction. In *SIAM Journal on Optimization*, 2022.
- Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *International Joint Conference on Artificial Intelligence*, 2018.
- Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In *Advances in Neural Information Processing Systems*, 2021.
- Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *International Conference on Machine Learning*, 2019.
- Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. DecentLaM: Decentralized momentum sgd for large-batch deep training. In *International Conference on Computer Vision*, 2021.