# Quasi-Global Momentum: Accelerating Decentralized Deep Learning on Heterogeneous Data

**Tao Lin** [1] **Sai Praneeth Karimireddy** [1] **Sebastian U. Stich** [1] **Martin Jaggi** [1]

## Abstract

Decentralized training of deep learning models is a key element for enabling data privacy and on-device learning over networks. In realistic learning scenarios, the presence of heterogeneity across different clients' local datasets poses an optimization challenge and may severely deteriorate the generalization performance.

In this paper, we investigate and identify the limitation of several decentralized optimization algorithms for different degrees of data heterogeneity. We propose a novel momentum-based method to mitigate this decentralized training difficulty. We show in extensive empirical experiments on various CV/NLP datasets (CIFAR-10, ImageNet, and AG News) and several network topologies (Ring and Social Network) that our method is much more robust to the heterogeneity of clients' data than other existing methods, by a significant improvement in test performance ($1\% - 20\%$). Our code is publicly available[1].

## 1. Introduction

Decentralized machine learning methods—allowing communications in a peer-to-peer fashion on an underlying communication network topology (without a central coordinator)—have emerged as an important paradigm in large-scale machine learning (Lian et al., 2017; 2018; Koloskova et al., 2019; 2020b). Decentralized Stochastic Gradient Descent (DSGD) methods offer (1) scalability to large datasets and systems in large data-centers (Lian et al., 2017; Assran et al., 2019; Koloskova et al., 2020a), as well as (2) privacy-preserving learning for the emerging EdgeAI applications (Kairouz et al., 2019; Koloskova et al., 2020a), where the training data remains distributed over a large number of clients (e.g. mobile phones, sensors, or hospitals) and is kept locally (never transmitted during training).

A key challenge—in particular in the second scenario—is the large heterogeneity (non-i.i.d.-ness) in the data present on the different clients (Zhao et al., 2018; Kairouz et al., 2019; Hsieh et al., 2020). Heterogeneous data (e.g. as illustrated in Figure 1) causes very diverse optimization objectives on each client, which results in slow and unstable global convergence, as well as poor generalization performance (shown in the inline table of Figure 1). Addressing these optimization difficulties is essential to realize reliable decentralized deep learning applications. Although such challenges have been theoretically pointed out in Shi et al. (2015); Lee et al. (2015); Tang et al. (2018b); Koloskova et al. (2020b), the empirical performance of different DSGD methods remains poorly understood. To the best of our knowledge, there currently exists no efficient, effective, and robust optimization algorithm yet for decentralized deep learning on heterogeneous data.

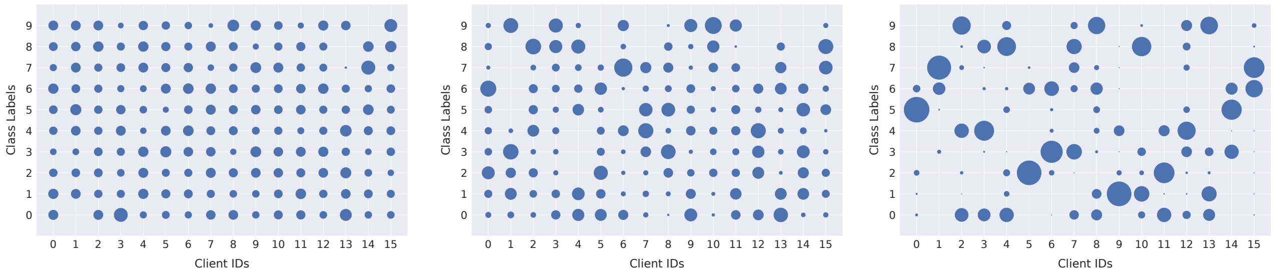
In the meantime, SGD with momentum acceleration (SGDm) remains the current workhorse for the state-of-the-art (SOTA) centralized deep learning training (He et al., 2016; Goyal et al., 2017; He et al., 2019). For decentralized deep learning, the currently used training recipes (i.e. DSGDm) maintain a local momentum buffer on each worker (Assran et al., 2019; Koloskova et al., 2020a; Nadiradze et al., 2020; Singh et al., 2020; Kong et al., 2021) while only communicating the model parameters to the neighbors. However, these attempts in prior work mainly consider homogeneous decentralized data—and there is no evidence that local momentum enhances generalization performance of decentralized deep learning on heterogeneous data.

As our first contribution, we investigate how DSGD and DSGDm are impacted by the degree of data heterogeneity and the choice of the network topology. We find that heterogeneous data hinders the local momentum acceleration in DSGDm. We further show that using a high-quality shared momentum buffer (e.g. synchronizing the momentum buffer globally) improves the optimization and generalization performance of DSGDm. However, such a global communication significantly increases the communication cost and violates the decentralized learning setup.

[1] EPFL, Lausanne, Switzerland. Correspondence to: Tao Lin <tao.lin@epfl.ch>.

*Proceedings of the $38^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] Code: github.com/epfml/quasi-global-momentum



(a) CIFAR-10, $n = 16$, $\alpha = 10$.

(b) CIFAR-10, $n = 16$, $\alpha = 1$.

(c) CIFAR-10, $n = 16$, $\alpha = 0.1$.

| Methods | Ring ($n = 16$) | | | Social Network ($n = 32$) | | |
|---|---|---|---|---|---|---|
| | $\alpha = 10$ | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 10$ | $\alpha = 1$ | $\alpha = 0.1$ |
| DSGD | $89.90 \pm 0.26$ | $88.88 \pm 0.26$ | $74.55 \pm 2.07$ | $89.95 \pm 0.23$ | $88.41 \pm 0.27$ | $77.56 \pm 1.65$ |
| DSGDm-N | $91.47 \pm 0.23$ | $89.98 \pm 0.10$ | $77.48 \pm 2.67$ | $91.17 \pm 0.11$ | $89.96 \pm 0.35$ | $80.59 \pm 2.32$ |
| QG-DSGDm-N (ours) | **91.90** $\pm 0.17$ | **91.28** $\pm 0.38$ | **82.20** $\pm 1.27$ | **91.51** $\pm 0.02$ | **91.00** $\pm 0.24$ | **85.19** $\pm 0.98$ |

Figure 1: **Illustrating the challenge of heterogeneous data in decentralized deep learning**, for training ResNet-EvoNorm-20 on CIFAR-10. The Dirichlet distribution $\alpha$ values control different non-i.i.d. degrees (Yurochkin et al., 2019; Hsu et al., 2019; He et al., 2020); the smaller $\alpha$ is, the more likely the clients hold examples from only one class. **The inline figures illustrate the # of samples per class allocated to each client (indicated by dot sizes)**, for the case of $\alpha = 10, 1, 0.1$. The test top-1 accuracy results in the table are averaged over three random seeds, with learning rate tuning for each setting. The performance upper bound (i.e. centralized training without local data re-shuffling) for $n = 16$ and 32 nodes are $92.95 \pm 0.13$ and $92.88 \pm 0.07$ respectively. Following prior work, the evaluated DSGD methods maintain local momentum buffer (without synchronization) for each worker; other experimental setup refers to Section 5.1.

We instead propose Quasi-Global (QG) momentum, a simple, yet effective, method that mitigates the difficulties for decentralized learning on heterogeneous data. Our approach is based on locally approximating the global optimization direction without introducing extra communication overhead. We demonstrate in extensive empirical results that QG momentum can stabilize the optimization trajectory, and that it can accelerate decentralized learning achieving much better generalization performance under high data heterogeneity than previous methods.

- We systematically examine the behavior of decentralized optimization algorithms on standard deep learning benchmarks for various degrees of data heterogeneity.
- We propose a novel momentum-based decentralized optimization method—QG-DSGDm and QG-DSGDm-N—to stabilize the local optimization. We validate the effectiveness of our method on a spectrum of non-i.i.d. degrees and network topologies—it is much more robust to the data heterogeneity than all other existing methods.
- We rigorously prove the convergence of our scheme.
- We additionally investigate different normalization methods alternative to Batch Normalization (BN) (Ioffe & Szegedy, 2015) in CNNs, due to its particular vulnerable to non-i.i.d. local data and the caused severe quality loss.

## 2. Related Work

**Decentralized Deep Learning.** The study of decentralized optimization algorithms dates back to Tsitsiklis (1984), relating to use gossip algorithms (Kempe et al., 2003; Xiao & Boyd, 2004; Boyd et al., 2006) to compute aggregates (find consensus) among clients. In the context of machine learning/deep learning, combining SGD with gossip averaging (Lian et al., 2017; 2018; Assran et al., 2019; Koloskova et al., 2020b) has gained a lot of attention recently for the benefits of *computational scalability*, *communication efficiency*, *data locality*, as well as the favorable leading term in the convergence rate $\mathcal{O}\big(\frac{1}{n\varepsilon^2}\big)$ (Lian et al., 2017; Scaman et al., 2017; 2018; Tang et al., 2018b; Koloskova et al., 2019; 2020a;b) which is the same as in centralized mini-batch SGD (Dekel et al., 2012). A weak version of decentralized learning also covers the recent emerging federated learning (FL) setting (Konecnỳ et al., 2016; McMahan et al., 2017; Kairouz et al., 2019; Karimireddy et al., 2020b; Lin et al., 2020b) by using (centralized) star-shaped network topology and local updates. Note that specializing our results to the FL setting is beyond the scope of our work. It is also non-trivial to adapt certain very recent techniques developed in FL for heterogeneous data (Karimireddy et al., 2020b;a; Lin et al., 2020b; Wang et al., 2020a; Das et al., 2020; Haddadpour et al., 2021) to the gossip-based decentralized deep learning.

A line of recent works on decentralized stochastic optimization, like D$^2$/Exact-diffusion (Tang et al., 2018b; Yuan et al., 2020a; Yuan & Alghunaim, 2021), and gradient tracking (Pu & Nedić, 2020; Pan et al., 2020; Lu et al., 2019), proposes different techniques to theoretically eliminate the influence of data heterogeneity between nodes. However, it remains unclear if these theoretically sound methods still endow with superior convergence and generalization properties in deep learning.

Other works focus on improving communication efficiency, from the aspect of communication compression (Tang et al., 2018a; Koloskova et al., 2019; 2020a; Lu & De Sa, 2020; Taheri et al., 2020; Singh et al., 2020; Vogels et al., 2020; Taheri et al., 2020; Nadiradze et al., 2020), less frequent communication through multiple local updates (Hendrikx et al., 2019; Koloskova et al., 2020b; Nadiradze et al., 2020), or better communication topology design (Nedić et al., 2018; Assran et al., 2019; Wang et al., 2019; 2020b; Neglia et al., 2020; Nadiradze et al., 2020; Kong et al., 2021).

**Mini-batch SGD with Momentum Acceleration.** Momentum is a critical component for training the SOTA deep neural networks (Sutskever et al., 2013; Lucas et al., 2019). Despite various empirical successes, the current theoretical understanding of momentum-based SGD methods remains limited (Bottou et al., 2016). A line of work on the serial (centralized) setting has aimed to develop a convergence analysis for different momentum methods as a special case (Yan et al., 2018; Gitman et al., 2019). However, SGD is known to be optimal in the worst case for stochastic non-convex optimization (Arjevani et al., 2019).
In distributed deep learning, most prior works focus on homogeneous data (especially for numerical evaluations) and incorporate momentum with a locally maintained buffer (which has no synchronization) (Lian et al., 2017; Assran et al., 2019; Lin et al., 2020c;a; Koloskova et al., 2020a; Singh et al., 2020). Yu et al. (2019) propose synchronizing the local momentum buffer periodically for better performance at the cost of doubling the communication. SlowMo (Wang et al., 2020c) instead proposes to periodically perform a slow momentum update on the globally synchronized model parameters (with additional All-Reduce communication cost), for centralized or decentralized methods. Parallel work (Balu et al., 2020) introduces DMSGD for decentralized learning[2]—it constructs the acceleration momentum from the mixture of the local momentum and consensus momentum. Our proposed method has no extra communication overhead and significantly outperforms all existing methods (in Section 5.2).

**Batch Normalization in Distributed Learning.** Batch Normalization (BN) (Ioffe & Szegedy, 2015) is an indispensable component in deep learning (Santurkar et al., 2018; Luo et al., 2019) and has been employed by default in most SOTA CNNs (He et al., 2016; Huang et al., 2016; Tan & Le, 2019). However, it often fails on distributed deep learning with heterogeneous local data due to the discrepancies between local activation statistics (see recent empirical examination for federated learning in Hsieh et al., 2020; Andreux et al., 2020; Li et al., 2021; Diao et al., 2021). As a remedy, Hsieh et al. (2020) propose to replace BN with Group Normalization (GN) (Wu & He, 2018) to address the issue of local BN statistics, while Andreux et al. (2020); Li et al. (2021); Diao et al. (2021) modify the way of synchronizing the local BN weight/statistics for better generalization performance. In the scope of decentralized learning, the effect of batch normalization has not been investigated yet.

## 3. Method

### 3.1. Notation and Setting

We consider sum-structured distributed optimization problems $f \colon \mathbb{R}^d \to \mathbb{R}$ of the form

$$f^\star := \min_{\mathbf{x} \in \mathbb{R}^d} \left[ f(\mathbf{x}) := \tfrac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}) \right] , \qquad (1)$$

where the components $f_i : \mathbb{R}^d \to \mathbb{R}$ are distributed among the $n$ nodes and are given in stochastic form: $f_i(\mathbf{x}) := \mathbb{E}_{\xi \sim \mathcal{D}_i} \left[ F_i(\mathbf{x}, \xi) \right]$, where $\mathcal{D}_i$ denotes the local data distribution on node $i \in [n]$. In D(ecentralized)SGD, each node $i$ maintains local parameters $\mathbf{x}_i^{(t)} \in \mathbb{R}^d$, and updates them as:

$$\mathbf{x}_i^{(t+1)} = \sum_{j=1}^n w_{ij} \left( \mathbf{x}_j^{(t)} - \eta \nabla F_j(\mathbf{x}_j^{(t)}, \xi_j^{(t)}) \right) , \qquad \text{(DSGD)}$$

that is, by a stochastic gradient step based on a sample $\xi_i^{(i)} \sim \mathcal{D}_i$, followed by gossip averaging with neighboring nodes in the network topology encoded by the mixing weights $w_{ij}$.

In this paper, we denote DSGD with local HeavyBall momentum by DSGDm, and DSGD with local Nesterov momentum by DSGDm-N; the naming rule also applies to our method. For the sake of simplicity, we use HeavyBall momentum variants in Section 3 and 4 for analysis purposes.

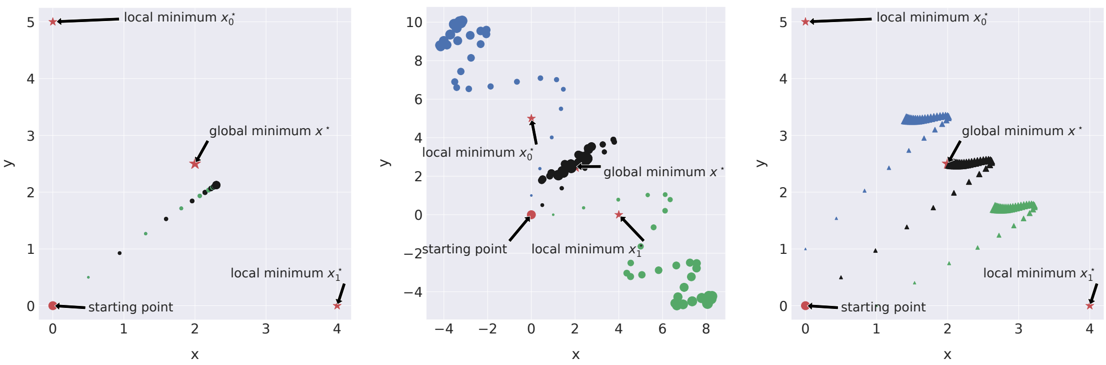
### 3.2. QG-DSGDm Algorithm

To motivate the algorithm design, we first illustrate the impact of using different momentum buffers (local vs. global) on distributed training on heterogeneous data.

**Heterogeneous data hinders local momentum acceleration—an example 2D optimization illustration.** In Figure 2 shows a toy 2D optimization example that simulates the biased local gradients caused by heterogeneous data. It depicts the optimization trajectories of two agents ($n = 2$) that start the optimization from the position $(0, 0)$ and receive in every iteration a gradient that points to the local minimum $(0, 5)$ and $(4, 0)$ respectively. The gradient is given by the direction from the current model (position) to the local minimum, and scaled to a constant update magnitude. Model synchronization (i.e. uniform averaging) is performed for every local model update step.

Heterogeneous data strongly influences the effectiveness of the local momentum acceleration. Though local momentum in Figure 2(b) assists the models to converge to the neighborhood of the global minimum (better convergence than when

[2] We detail the DMSGD algorithm and clarify the difference in Appendix B.2; we empirically compare with DMSGD in Table 5.



(a) w/o local momentum.

(b) w/ local HeavyBall momentum.

(c) Our scheme.

Figure 2: **The ineffectiveness of local momentum acceleration under heterogeneous data setup**: the local momentum buffer accumulates "biased" gradients, causing unstable and oscillation behaviors. The size of marker will increase by the number of update steps; colors blue and green indicate the local models of two workers (after performing local update), while color black is the synchronized global model. Uniform weight averaging is performed after each update step, and the new gradients will be computed on the averaged model. We use the common $\beta = 0.9$ in this illustration and more results on different $\beta$ values refer to Appendix D.2.

excluding local momentum in Figure 2(a)), it also causes an unstable and oscillation optimization trajectory. The problem gets even worse in decentralized deep learning, where the learning relies on stochastic gradients from non-convex function and only has limited communication.

**Synchronizing the local momentum buffers boosts decentralized learning.** We here consider a hypothetical method, which synchronizes the local momentum buffer as in Yu et al. (2019), to use the global momentum buffer locally (avoid using ill-conditioned local momentum buffer caused by heterogeneous data, as shown by the poor performance in Figure 1). We can witness from Table 5 that synchronizing the buffer per update step by global averaging to some extent mitigates the issue caused by heterogeneity ($1\% - 5\%$ improvement comparing row 3 with row 7 in Table 5). Despite its effectiveness, the global synchronization fundamentally violates the realistic decentralized learning setup and introduces extra communication overhead.

**Our proposal—QG-DSGDm.** Motivated by the performance gain brought by employing a global momentum buffer, we propose a **Q**uasi-**G**lobal (QG) momentum buffer—a communication-free approach to mimic the global optimization direction—to mitigate the difficulties for decentralized learning on heterogeneous data. Integrating quasi-global momentum with local stochastic gradients alleviates the drift in the local optimization direction, and thus results in a stabilized training and high robustness to heterogeneous data.

Algorithm 1 highlights the difference between DSGDm and QG-DSGDm. Instead of using local gradients from heterogeneous data to form the local momentum (line 4 for DSGDm), which may significantly deflect from the global optimization direction, for QG-DSGDm, we use the difference of two consecutive synchronized models (line 8)

**Algorithm 1** Decentralized learning algorithms: QG-DSGDm v.s. DSGDm ; Colors indicate the two alternative algorithm variants. At initialization $\mathbf{m}_i^{(0)} = \hat{\mathbf{m}}_i^{(0)} := \mathbf{0}$.

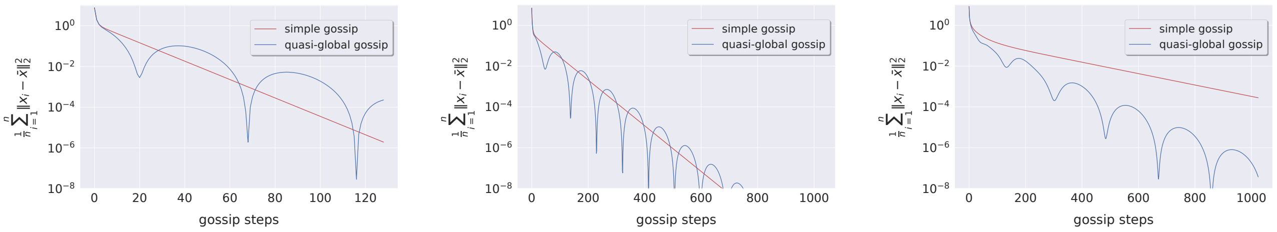
```
1:  procedure WORKER-$i$
2:      for $t \in \{1, \ldots, T\}$ do
3:          sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
4:          $\mathbf{m}_i^{(t)} = \beta \mathbf{m}_i^{(t-1)} + \mathbf{g}_i^{(t)}$
5:          $\mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}$
6:          $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta \mathbf{m}_i^{(t)}$
7:          $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+\frac{1}{2})}$
8:          $\mathbf{d}_i^{(t)} = \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}$
9:          $\hat{\mathbf{m}}_i^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu) \mathbf{d}_i^{(t)}$
10:     return $\mathbf{x}_i^{(T)}$
```

$$
\mathbf{d}_i^{(t)} = \frac{1}{\eta} \left( \mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)} \right) , \tag{2}
$$

to update the momentum buffer (line 9) by $\hat{\mathbf{m}}_i^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu)\mathbf{d}_i^{(t)}$. We set $\mu = \beta$ for all our numerical experiments, without needing hyper-parameter tuning.

The update scheme of QG-DSGDm can be re-formulated in matrix form ($\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, etc.) as follows

$$
\begin{aligned}
\mathbf{X}^{(t+1)} &= \mathbf{W} \left( \mathbf{X}^{(t)} - \eta \left( \beta \mathbf{M}^{(t-1)} + \mathbf{G}^{(t)} \right) \right) \\
\mathbf{M}^{(t)} &= \mu \mathbf{M}^{(t-1)} + (1-\mu) \tfrac{\mathbf{X}^{(t)} - \mathbf{X}^{(t+1)}}{\eta} .
\end{aligned} \tag{3}
$$



(a) $n = 16$. (b) $n = 32$. (c) $n = 64$.

Figure 3: **Understanding QG-DSGDm through the distributed average consensus problem on a fixed ring topology**. QG-DSGDm without gradient update step ((4)) still presents faster convergence (to a relative high precision) than the standard gossip averaging. Appendix D.1 illustrates the results on other communication topologies and topology scales.

### 3.3. Convergence Analysis

We provide a convergence analysis for our novel QG-DSGDm for non-convex functions. The proof details can be found in Appendix C.

**Assumption 1.** *We assume that the following hold:*

- *The function $f(\mathbf{x})$ we are minimizing is lower bounded from below by $f^\star$, and each node's loss $f_i$ is smooth satisfying $\|\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})\| \le L\,\|\mathbf{y} - \mathbf{x}\|$.*
- *The stochastic gradients within each node satisfies $\mathbb{E}\left[g_i(\mathbf{x})\right] = \nabla f_i(\mathbf{x})$ and $\mathbb{E}\left\|g_i(\mathbf{x}) - \nabla f_i(\mathbf{x})\right\|^2 \le \sigma^2$. The variance across the workers is also bounded as $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \le \zeta^2$.*
- *The mixing matrix is doubly stochastic: for all ones vector $\mathbf{1}$, we have $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $\mathbf{W}^\top \mathbf{1} = \mathbf{1}$.*
- *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 $\mathbb{E}_{\mathbf{W}}\left\|\mathbf{Z}\mathbf{W} - \bar{\mathbf{Z}}\right\|_F^2 \le (1 - \rho)\left\|\mathbf{Z} - \bar{\mathbf{Z}}\right\|_F^2$.*

**Theorem 3.1** (Convergence of QG-DSGDm for non–convex functions)**.** *Given Assumption 1, the sequence of iterates generated by* (3) *for step size $\eta = \mathcal{O}\left(\sqrt{\frac{n}{\sigma^2 T}}\right)$ and momentum parameter $\frac{\beta}{1-\beta} \le \frac{\rho}{21}$ satisfies $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^t)\right\|^2 \le \epsilon$ in iterations*

$$T = \mathcal{O}\Big(\frac{L\sigma^2}{n\epsilon^2} + \frac{L\tilde{\zeta}}{\rho\epsilon^{3/2}} + \frac{L}{\epsilon}\big(\frac{1}{\rho} + \frac{1}{(1-\mu)(1-\beta)^2}\big)\Big)\,,$$

*where $\tilde{\zeta}^2 := \zeta^2 + (1 + \frac{1-\beta}{1-\mu})\sigma^2$.*

**Remark 3.2.** *The asymptotic number of iterations required, $\mathcal{O}\big(\frac{\sigma^2}{n\epsilon^2}\big)$ shows perfect linear speedup in the number of workers $n$, independent of the communication topology. This upper bound matches the convergence bounds of DSGD (Lian et al., 2017) and centralized mini-batch SGD (Dekel et al., 2012), and is optimal (Arjevani et al., 2019). This significantly improves over previous analyses of distributed momentum methods which need $\frac{L\sigma^2}{n(1-\beta)\epsilon^2}$ iterations, slowing down for larger values of $\beta$ (Yu et al., 2019; Balu et al., 2020). The second* drift term *$\frac{1}{\rho\epsilon^{3/2}}$ arises due to the non-iid data distribution, and matches the tightest analysis of DSGD without momentum (Koloskova et al., 2020b). Finally, our theorem imposes some constraint on the momentum parameter $\beta$ (but not on $\mu$). In practice however, QG-DSGDm performs well even when this constraint is violated.*

### 3.4. Connection with Other Methods

We bridge quasi-global momentum with two recent works below. The corresponding algorithm details are included in Appendix B.1 for clarity.

**Connection with MimeLite.** MimeLite (Karimireddy et al., 2020a) was recently introduced in a preprint for FL on heterogeneous data. It shares a similar ingredient as ours: a “global” movement direction $\mathbf{d}$ is used locally to alleviate the issue caused by heterogeneity. The difference falls into the way of forming $\mathbf{d}$ (c.f. line 8 in Algorithm 1): in MimeLite, $\mathbf{d}$ is the full batch gradients computed on the previously synchronized model, while the $\mathbf{d}$ in our QG-DSGDm is the difference on two consecutive synchronized models.
MimeLite only addresses the FL setting, which results in a computation and communication overhead (to form $\mathbf{d}$), and is non-trivial to extend to decentralized learning.

**Connection with SlowMo.** SlowMo and its “noaverage” variant (Wang et al., 2020c) aim to improve generalization performance in the homogeneous data-center training scenario, while QG-DSGDm is targeting learning with data heterogeneity. In terms of update scheme, SlowMo variants update the slow momentum buffer through the model difference $\mathbf{d}$ of $\tau \gg 1$ local update (and synchronization) steps, while QG-DSGDm only considers consecutive models (analogously $\tau = 1$)[3]. Besides, in contrast to QG-DSGDm, the slow momentum buffer in SlowMo will never interact with the local update—setting $\tau$ to 1 in SlowMo variants cannot recover QG-DSGDm.
SlowMo variants are orthogonal to QG-DSGDm; combining these two algorithms may lead to a better generalization performance, and we leave it for future work.

[3] We also study the variant of QG-DSGDm with $\tau > 1$ in Appendix D.8—we stick to $\tau = 1$ in the main paper for its superior performance and hyper-parameter ($\tau$) tuning free.

## 4. Understanding QG-DSGDm

### 4.1. Faster Convergence in Average Consensus

We now consider the simpler averaging consensus problem (isolated from the learning part of QG-DSGDm): we simplify (3) by removing gradients and step-size:

$$\begin{aligned}\mathbf{X}^{(t+1)} &= \mathbf{W}\left(\mathbf{X}^{(t)} - \beta\mathbf{M}^{(t-1)}\right) \\ \mathbf{M}^{(t)} &= \mu\mathbf{M}^{(t-1)} + (1-\mu)\left(\mathbf{X}^{(t)} - \mathbf{X}^{(t+1)}\right) ,\end{aligned} \tag{4}$$

and compare it with gossip averaging $\mathbf{X}^{(t+1)} = \mathbf{W}\mathbf{X}^{(t)}$.

Figure 3 depicts the advantages of (4) over standard gossip averaging, where QG-DSGDm can quickly converge to a critical consensus distance (e.g. $10^{-2}$). It partially explains the performance gain of QG-DSGDm from the aspect of improved decentralized communication (which leads to better optimization)—decentralized training can converge as fast as its centralized counterpart once the consensus distance is lower than the critical one, as stated in Kong et al. (2021).

### 4.2. QG-DSGDm (Single Worker Case) Recovers QHM

Considering the single worker case, QG-DSGDm can be further simplified to (derivations in Appendix B.3.1):

$$\begin{aligned}\hat{\mathbf{m}}^{(t)} &= \hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)} \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left((1 - \tfrac{\mu}{\hat{\beta}})\hat{\mathbf{m}}^{(t)} + \tfrac{\mu}{\hat{\beta}}\mathbf{g}^{(t)}\right) ,\end{aligned}$$

where $\hat{\beta} := \mu + (1-\mu)\beta$. Thus, the single worker case of QG-DSGDm (i.e. QG-SGDm) recovers Quasi-Hyperbolic Momentum (QHM) (Ma & Yarats, 2019; Gitman et al., 2019). We illustrate its acceleration benefits as well as the performance gain in Figure 12 and Figure 13 of Appendix D.3. We elaborate in Appendix B.3 that SGDm is only a special case of QG-SGDm/QHM (by setting $\mu = 0$). Besides, it is non-trivial to adapt (centralized) QHM to (decentralized) QG-DSGDm due to discrepant motivation.

**Stabilized optimization trajectory.** We study the optimization trajectory of Rosenbrock function (Rosenbrock, 1960)[4] $f(x, y) = (y - x^2)^2 + 100(x-1)^2$ as in Lucas et al. (2019) to better understand the performance gain of QG-SGDm (with zero stochastic noise). Figure 15 illustrates the effects of stabilization in QG-SGDm (much less oscillation than SGDm).

**Larger effective step-size.** Recent works (Hoffer et al., 2018; Zhang et al., 2019a) point out the larger effective step-size (i.e. $\eta/\left\|\mathbf{x}_t\right\|_2^2$) brought by weight decay provides the primary regularization effect for deep learning training. Figure 5 examines the effective step-size during the optimization procedure: QG-SGDm illustrates a larger effective step-size than SGDm, explaining the performance gain e.g. in Figure 12 and Figure 13 of Appendix D.3.

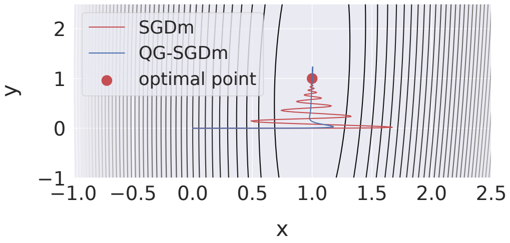
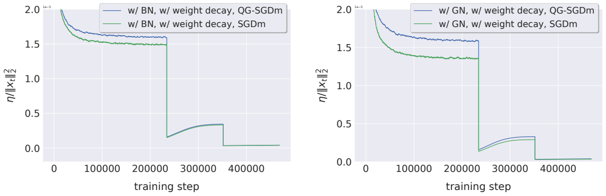


Figure 4: **Understanding the optimization trajectory of QG-SGDm and SGDm** (i.e. single worker case) via a 2D toy function $f(x, y) = (y - x^2)^2 + 100(x-1)^2$. This function has a global minimum at $(x, y) = (1, 1)$. SGDm and QG-SGDm use $\beta = 0.9, \eta = 0.001$, with initial point $(0, 0)$. Trajectories for different initial points and/or $\beta$ values refer to Appendix D.4.



(a) ResNet-BN-20.

(b) ResNet-GN-20.

Figure 5: **The effective step-size $\eta/\left\|\mathbf{x}_t\right\|_2^2$ of QG-SGDm and SGDm** (single worker case) on CIFAR-10. The weight norm curves refer to Figure 14 in Appendix D.3.

## 5. Experiments

### 5.1. Setup

**Datasets and models.** We empirically study the decentralized training behavior on both CV and NLP benchmarks, on the architecture of ResNet (He et al., 2016), VGG (Simonyan & Zisserman, 2014) and DistilBERT (Sanh et al., 2019). • Image classification (CV) benchmark: we consider training CIFAR-10 (Krizhevsky & Hinton, 2009), ImageNet-32 (i.e. image resolution of 32) (Chrabaszcz et al., 2017), and ImageNet (Deng et al., 2009) from scratch, with standard data augmentation and preprocessing scheme (He et al., 2016). We use VGG-11 (with width factor 1/2 and without BN) and ResNet-20 for CIFAR-10, ResNet-20 with width factor 2 (noted as ResNet-20-x2) for ImageNet-32, and ResNet-18 for ImageNet. The width factor indicates the proportional scaling of the network width corresponding to the original neural network. Weight initialization schemes follow He et al. (2015); Goyal et al. (2017). • Text classification (NLP) benchmark: we perform fine-tuning on a 4-class classification dataset (AG News (Zhang et al., 2015)). Unless mentioned otherwise, all our experiments are repeated over three random seeds. We report the averaged performance of local models on the full test dataset.

**Heterogeneous distribution of client data.** We use the Dirichlet distribution to create disjoint non-i.i.d. client

[4] We further study the optimization trajectory for more complicated non-convex function in Appendix D.4.

Table 1: **The test top-1 accuracy of different decentralized optimization algorithms evaluated on different degrees of non-i.i.d. local CIFAR-10 data, for various neural architectures and network topologies.** The results are averaged over three random seeds, with learning rate tuning for each setting. We also include the results of centralized baseline for reference purposes, following the decentralized experiment configuration, except that the centralized baseline uses randomly partitioned local training data (i.e. independent of $\alpha$).

| Datasets | Neural Architectures | Methods | Ring ($n=16$) | | | Social Network ($n=32$) | | |
|---|---|---|---|---|---|---|---|---|
| | | | $\alpha = 10$ | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 10$ | $\alpha = 1$ | $\alpha = 0.1$ |
| CIFAR-10 | ResNet-BN-20 | SGDm-N (centralized) | | $92.95 \pm 0.13$ | | | $92.88 \pm 0.07$ | |
| | | DSGD | $90.94 \pm 0.15$ | $88.95 \pm 0.59$ | $54.66 \pm 3.58$ | $90.52 \pm 0.24$ | $89.22 \pm 0.35$ | $58.32 \pm 3.27$ |
| | | DSGDm-N | $92.53 \pm 0.27$ | $89.13 \pm 0.81$ | $57.19 \pm 2.65$ | $92.20 \pm 0.24$ | $90.19 \pm 0.54$ | $63.00 \pm 2.50$ |
| | | QG-DSGDm-N | **92.65** $\pm 0.17$ | **91.21** $\pm 0.28$ | **58.16** $\pm 3.32$ | **92.52** $\pm 0.09$ | **91.20** $\pm 0.16$ | **64.32** $\pm 2.43$ |
| | ResNet-GN-20 | SGDm-N (centralized) | | $88.06 \pm 1.12$ | | | $86.19 \pm 1.08$ | |
| | | DSGD | $86.86 \pm 0.37$ | $85.93 \pm 0.14$ | $73.14 \pm 3.92$ | $84.00 \pm 0.67$ | $82.98 \pm 0.47$ | $67.84 \pm 3.94$ |
| | | DSGDm-N | $89.86 \pm 0.15$ | $88.30 \pm 0.49$ | $71.86 \pm 2.22$ | $88.54 \pm 0.22$ | $86.36 \pm 0.55$ | $72.02 \pm 1.79$ |
| | | QG-DSGDm-N | **90.18** $\pm 0.44$ | **89.68** $\pm 0.41$ | **82.78** $\pm 2.05$ | **88.58** $\pm 0.09$ | **88.19** $\pm 0.30$ | **83.60** $\pm 1.83$ |
| | ResNet-EvoNorm-20 | SGDm-N (centralized) | | $92.18 \pm 0.19$ | | | $91.92 \pm 0.33$ | |
| | | DSGD | $89.90 \pm 0.26$ | $88.88 \pm 0.26$ | $74.55 \pm 2.07$ | $89.95 \pm 0.23$ | $88.41 \pm 0.27$ | $77.56 \pm 1.65$ |
| | | DSGDm-N | $91.47 \pm 0.23$ | $89.98 \pm 0.10$ | $77.48 \pm 2.67$ | $91.17 \pm 0.11$ | $89.96 \pm 0.35$ | $80.59 \pm 2.32$ |
| | | QG-DSGDm-N | **91.90** $\pm 0.17$ | **91.28** $\pm 0.38$ | **82.20** $\pm 1.27$ | **91.51** $\pm 0.02$ | **91.00** $\pm 0.24$ | **85.19** $\pm 0.98$ |
| | VGG-11<br>(w/o normalization layer) | SGDm-N (centralized) | | $88.87 \pm 0.29$ | | | $87.38 \pm 0.39$ | |
| | | DSGDm-N | $88.68 \pm 0.30$ | $88.52 \pm 0.24$ | $77.45 \pm 3.15$ | $86.39 \pm 0.06$ | $85.85 \pm 0.22$ | $77.02 \pm 2.66$ |
| | | QG-DSGDm-N | **89.01** $\pm 0.04$ | **89.00** $\pm 0.22$ | **83.41** $\pm 2.20$ | **86.87** $\pm 0.60$ | **86.09** $\pm 0.30$ | **84.86** $\pm 0.58$ |

training data (Yurochkin et al., 2019; Hsu et al., 2019; He et al., 2020)—the created client data is fixed and never shuffled across clients during the training. The degree of non-i.i.d.-ness is controlled by the value of $\alpha$; the smaller $\alpha$ is, the more likely the clients hold examples from only one class. An illustration regarding how samples are distributed among 16 clients on CIFAR-10 can be found in Figure 1; more visualizations on other datasets/scales are shown in Appendix A.2. Besides, Figure 7 in Appendix A.1 visualizes the Social Network topology.

**Training schemes.** Following the SOTA deep learning training scheme, we use mini-batch SGD as the base optimizer for CV benchmark (He et al., 2016; Goyal et al., 2017), and similarly, Adam for NLP benchmark (Zhang et al., 2019b; Mosbach et al., 2021). In Section 5.2, we adapt these base optimizers to different distributed variants[5].

For the CV benchmark, the models are trained for 300 and 90 epochs for CIFAR-10 and ImageNet(-32) respectively; the local mini-batch size are set to 32 and 64. All experiments use the SOTA learning rate scheme in distributed deep learning training (Goyal et al., 2017; He et al., 2019) with learning rate scaling and warm-up. The learning rate is always gradually warmed up from a relatively small value (i.e. 0.1) for the first 5 epochs. Besides, the learning rate will be divided by 10 when the model has accessed specified fractions of the total number of training samples—$\{\frac{1}{2}, \frac{3}{4}\}$ for CIFAR and $\{\frac{1}{3}, \frac{2}{3}, \frac{8}{9}\}$ for ImageNet.
For the NLP benchmark, we fine-tune the *distilbert-base-uncased* from HuggingFace (Wolf et al., 2019) with constant learning rate and mini-batch of size 32 for 10 epochs.

We fine-tune the learning rate for both CV[6] and NLP tasks; we use constant weight decay (1e-4). The tuning procedure ensures that the best hyper-parameter lies in the middle of our search grids; otherwise we extend our search grid. Regarding momentum related hyper-parameters, we follow the common practice in the community ($\beta = 0.9$ and without dampening for Nesterov/HeavyBall momentum variants, and $\beta_1 = 0.9, \beta_2 = 0.99$ for Adam variants).

**BN and its alternatives for distributed deep learning.** The existence of BN layer is challenging for the SOTA distributed training, especially for heterogeneous data setting. To better understand the impact of different normalization schemes in distributed deep learning, we investigate:

- Distributed BN implementation. Our default implementation[7] follows Goyal et al. (2017); Andreux et al. (2020) that computes the BN statistics independently for each client while only synchronizing the BN weights.
- Using other normalization layers: for instance on ResNet with BN layers (denoted by ResNet-BN-20), we can instead use ResNet-GN by replacing all BN with GN with group number of 2, as suggested in Hsieh et al. (2020). We also examine the recently proposed S0 variant of EvoNorm (Liu et al., 2020) (which does not use runtime mini-batches statistics), noted as ResNet-EvoNorm.

[5] We by default use local momentum variants without buffer synchronization. We consider DSGDm-N as our primary competitor for CNNs, as Nesterov momentum is the SOTA training scheme. We also investigate the performance of DSGDm in Table 5.

[6] We tune the initial learning rate and warm it up from 0.1 (if the tuned one is above 0.1).

[7] We also try the BN variant (Li et al., 2021) proposed for FL, but we exclude it in our comparison due to its poor performance.

Table 2: **Comparison with Gradient Tracking (GT) methods** for training CIFAR-10. D$^2$ and D$^2_+$ do not include the momentum acceleration. We carefully tune the learning rate for each case, and results are averaged over three seeds where std is indicated.

| | ResNet-EvoNorm-20 on Ring ($n=16$) | | | | | | ResNet-EvoNorm-20 on Ring ($n=32$) | | |
|---|---|---|---|---|---|---|---|---|---|
| | DSGD (w/ GT) | DSGDm-N | DSGDm-N (w/ GT) | D$^2$ | D$^2_+$ | QG-DSGDm-N | DSGDm-N | DSGDm-N (w/ GT) | QG-DSGDm-N |
| $\alpha=1$ | $87.36 \pm 0.40$ | $89.98 \pm 0.10$ | $90.38 \pm 0.41$ | 74.89 | $85.70 \pm 0.29$ | **91.28** $\pm 0.38$ | $88.46 \pm 0.29$ | $89.44 \pm 0.60$ | **90.27** $\pm 0.27$ |
| $\alpha=0.1$ | $66.16 \pm 1.05$ | $77.48 \pm 2.67$ | $78.64 \pm 1.84$ | 49.80 | $69.18 \pm 3.30$ | **82.20** $\pm 1.27$ | $78.17 \pm 1.63$ | $79.25 \pm 2.17$ | **83.18** $\pm 1.11$ |

### 5.2. Results

**Comments on BN and its alternatives.** Table 1 and Table 3 examine the effects of BN and its alternatives on the training quality of decentralized deep learning on CIFAR-10 and ImageNet dataset. *ResNet with EvoNorm replacement outperforms its GN counterpart* on a spectrum of optimization algorithms, non-i.i.d. degrees, and network topologies, *illustrating its efficacy to be a new alternative to BN in CNNs for distributed learning on heterogeneous data.*

**Superior performance of quasi-global momentum.** We evaluate QG-DSGDm-N and compare it with several DSGD variants in Table 1, for training different neural networks on CIFAR-10 in terms of different non-i.i.d. degrees on Ring ($n = 16$) and Social Network ($n = 32$). *QG-DSGDm-N accelerates the training by stabilizing the oscillating optimization trajectory caused by heterogeneity and leads to a significant performance gain over all other strong competitors on all levels of data heterogeneity. The benefits of our method are further pronounced when considering a higher degree of non-i.i.d.-ness.* These observations are consistent with the results on the challenging ImageNet(-32) dataset in Table 3 (and the learning curves in Figure 17 in Appendix D.5).

Table 3: **Test top-1 accuracy of different decentralized optimization algorithms evaluated on different degrees of non-i.i.d. local ImageNet data**. The results are over three random seeds. We perform sufficient learning rate tuning on ImageNet-32 for each setup while we use the same one for ImageNet due to the computational feasibility. "$\star$" indicates non-convergence.

| Datasets | Neural Architectures | Methods | Ring ($n=16$) | |
|---|---|---|---|---|
| | | | $\alpha = 1$ | $\alpha = 0.1$ |
| ImageNet-32 (resolution 32) | ResNet-20-x2 (EvoNorm) | SGDm-N (centralized) | $44.43 \pm 0.20$ | |
| | | DSGDm-N | $30.35 \pm 0.05$ | $16.71 \pm 0.17$ |
| | | QG-DSGDm-N | **31.24** $\pm 0.27$ | **19.53** $\pm 0.91$ |
| | ResNet-20-x2 (GN) | SGDm-N (centralized) | $37.89 \pm 0.67$ | |
| | | DSGDm-N | $34.16 \pm 1.37$ | $\star$ |
| | | QG-DSGDm-N | **38.57** $\pm 0.45$ | **21.42** $\pm 0.81$ |
| ImageNet | ResNet-18 (EvoNorm) | SGDm-N (centralized) | $69.55 \pm 0.25$ | |
| | | DSGDm-N | $68.77 \pm 0.05$ | $53.15 \pm 0.14$ |
| | | QG-DSGDm-N | **69.20** $\pm 0.08$ | **56.50** $\pm 0.01$ |
| | ResNet-18 (GN) | SGDm-N (centralized) | $62.59 \pm 0.01$ | |
| | | DSGDm-N | $60.76 \pm 0.48$ | $39.57 \pm 1.22$ |
| | | QG-DSGDm-N | **64.92** $\pm 0.27$ | **47.86** $\pm 1.05$ |

**Decentralized Adam.** We further extend the idea of quasi-global momentum to the Adam optimizer for decentralized learning, noted as QG-DAdam (the algorithm details are deferred to Algorithm 2 in Appendix B.1). We validate the effectiveness of QG-DAdam over D(decentralized)Adam in Table 6, on fine-tuning DistilBERT on AG News and training ResNet-EvoNorm-20 on CIFAR-10 from scratch: *QG-DAdam is still preferable over DAdam*. We leave a better adaptation and theoretical proof for future work.

Table 4: **Test top-1 accuracy of different decentralized SGD algorithms evaluated on different degrees of non-i.i.d.-ness and communication topologies**, for training ResNet-EvoNorm-18 on ImageNet. The results are over three random seeds. We use the same learning rate for different experiments due to the computational feasibility. Centralized SGDm-N reaches $69.55 \pm 0.25$.

| Communication Topology | Methods | Test Top-1 Accuracy | |
|---|---|---|---|
| | | $\alpha = 1$ | $\alpha = 0.1$ |
| Ring ($n=16$) | DSGDm-N | $68.77 \pm 0.05$ | $53.15 \pm 0.14$ |
| | QG-DSGDm-N | **69.20** $\pm 0.08$ | **56.50** $\pm 0.01$ |
| 1-peer directed exponential graph ($n=16$) (Assran et al., 2019) | DSGDm-N | $69.00 \pm 0.11$ | $58.52 \pm 0.27$ |
| | QG-DSGDm-N | **69.34** $\pm 0.17$ | **61.44** $\pm 0.20$ |

**Generalizing quasi-global momentum to time-varying topologies.** The benefits of quasi-global momentum are not limited to the fixed and undirected communication topologies, e.g. Ring and Social network in Table 1—it also generalizes to other topologies, like the time-varying directed topology (Assran et al., 2019), as shown in Table 4 for training ResNet-EvoNorm-18 on ImageNet. These results are aligned with the insights of the critical consensus distance on the generalization performance of decentralized deep learning (Kong et al., 2021), supporting the fact that *quasi-global momentum can be served as a simple plugin to further improve the performance of decentralized deep learning.*

**Comparison with D$^2$ and Gradient Tracking (GT).** As shown in Table 2, D$^2$ (Tang et al., 2018b) and GT methods (Pu & Nedić, 2020; Pan et al., 2020; Lu et al., 2019) cannot achieve comparable test performance on the standard deep learning benchmark, while QG-DSGDm-N outperforms them significantly. Additional detailed comparisons are deferred to Appendix D.9.
It is non-trivial to integrate D$^2$ with momentum. Besides, D$^2$ requires constant learning rate, which does not fit the SOTA learning rate schedules (e.g. stage-wise) in deep learning[8]. We include an improved D$^2$ variant[9] (denoted as D$^2_+$) to

[8] D$^2$ can be rewritten as $\mathbf{w}(\mathbf{x}^{(t)} - \eta((\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)})/\eta + \nabla f(\mathbf{x}^{(t)}) - \nabla f(\mathbf{x}^{(t-1)})))$, and the update would break if the magnitude of $\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)}$ is a factor of $10\eta$ (i.e. performing learning rate decay at step $t$).

[9] The update scheme of D$^2_+$ follows $\mathbf{w}(\mathbf{x}^{(t)} - \eta^{(t)}((\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)})/\eta^{(t-1)} + \nabla f(\mathbf{x}^{(t)}) - \nabla f(\mathbf{x}^{(t-1)})))$.

Table 5: **An extensive investigation for a wide spectrum of DSGD variants**, for training ResNet-EvoNorm-20 on CIFAR-10. The results are averaged over three seeds, each with learning rate tuning. We use "communication topology" to synchronize the model parameters, while some methods involve "extra communication", with specified objective to be communicated on the given network topology.

| Methods | Communication Topology | Extra Communication | Momentum Type | Test Top-1 Accuracy ($n=16$) | |
|---|---|---|---|---|---|
| | | | | $\alpha = 1$ | $\alpha = 0.1$ |
| SGDm-N | complete | - | global | $92.18 \pm 0.19$ | |
| DSGDm-N | complete | - | local | $91.47 \pm 0.10$ | $71.24 \pm 3.08$ |
| DSGDm-N | ring | momentum buffer (complete) | local | $90.96 \pm 0.33$ | $81.22 \pm 1.78$ |
| SlowMo | ring | model parameters (complete) | local & global | $91.06 \pm 0.26$ | $79.20 \pm 1.16$ |
| DSGD | ring | - | - | $88.88 \pm 0.26$ | $74.55 \pm 2.07$ |
| DSGDm | ring | - | local | $89.67 \pm 0.33$ | $77.66 \pm 0.95$ |
| DSGDm-N | ring | - | local | $89.98 \pm 0.10$ | $77.48 \pm 2.67$ |
| DSGDm | ring | momentum buffer (ring) | local | $90.42 \pm 0.32$ | $78.69 \pm 2.39$ |
| DSGDm-N | ring | momentum buffer (ring) | local | $90.48 \pm 0.67$ | $79.83 \pm 2.29$ |
| DSGDm-N | ring | local gradients (ring) | local | $90.10 \pm 0.61$ | $78.58 \pm 4.12$ |
| DMSGD | ring | - | local | $90.06 \pm 0.04$ | $79.89 \pm 0.97$ |
| QG-DSGDm | ring | - | local | $91.22 \pm 0.41$ | $\mathbf{82.24} \pm 1.05$ |
| QG-DSGDm-N | ring | - | local | $\mathbf{91.28} \pm 0.38$ | $82.20 \pm 1.27$ |

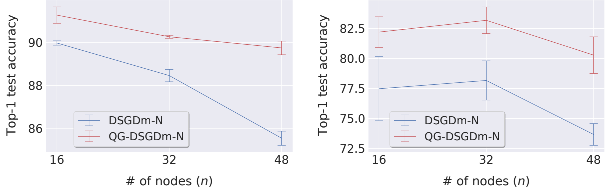
Table 6: **Test accuracy of different decentralized optimization algorithms (with Adam), evaluated on different degrees of non-i.i.d. local data**. The results are over three random seeds, with tuned learning rate.

| Models & Datasets | Methods | $\alpha = 0.1$ |
|---|---|---|
| Fine-tuning DistilBERT-base (AG News) | DAdam | $87.29 \pm 0.60$ |
| | QG-DAdam | $\mathbf{88.33} \pm 0.67$ |
| Training ResNet-EvoNorm-20 from scratch (CIFAR-10) | DAdam | $65.52 \pm 3.32$ |
| | QG-DAdam | $\mathbf{66.86} \pm 2.81$ |

address this learning rate decay issue in D$^2$, but the performance of D$^2_+$ still remains far behind our scheme.

**Ablation study.** Table 5 empirically investigates a wide range of different DSGD variants, in terms of the generalization performance on different degrees of data heterogeneity. We can witness that (1) DSGD variants with quasi-global momentum always significantly surpass all other methods (excluding the centralized upper bound), without introducing extra communication cost; (2) local momentum accelerates the decentralized optimization (c.f. the results of DSGD v.s. DSGDm and DSGDm-N), while our quasi-global momentum further improves the performance gain; (3) synchronizing local momentum buffer or local gradients only marginally improves the generalization performance, but the gains fall behind our quasi-global momentum (as we accelerate the consensus and stabilize trajectories, as illustrated in Section 4); (4) the parallel work DMSGD[10] (Balu et al., 2020) does show some improvements, but its performance gain is much less significant than ours. Table 18 in the Appendix D.6 further shows that tuning momentum factor for DSGDm-N cannot alleviate the training difficulty caused by data heterogeneity.

Besides, Figure 6 showcases the generality of quasi-global momentum for achieving remarkable performance gain on different topology scales and non-i.i.d. degrees.



(a) $\alpha=1$. (b) $\alpha=0.1$.

Figure 6: **Test top-1 accuracy of different decentralized algorithms evaluated on different topology scales and non-i.i.d. degrees**, for training ResNet-EvoNorm-20 on CIFAR-10. The results are over three random seeds, each with sufficient learning rate tuning. Colors blue and red indicate DSGDm-N and QG-DSGDm-N respectively. Numerical results refer to Table 7 in Appendix D.7.

## Conclusion

We demonstrated that heterogeneity has an out sized impact on the performance of deep learning models, leading to unstable convergence and poor performance. We proposed a novel momentum-based algorithm to stabilize the training and established its efficacy through thorough empirical evaluations. Our method, especially for mildly heterogeneous settings, leads to a 10–20% increase in accuracy. However, a gap still remains between the centralized training. Closing this gap, we believe, is critical for wider adoption of decentralized learning.

## Acknowledgements

We acknowledge funding from a Google Focused Research Award, Facebook, and European Horizon 2020 FET Proactive Project DIGIPREDICT.

[10] We tune both learning rate $\eta$ and weighting factor $\mu$ (using the grid suggested in Balu et al. (2020)) for DMSGD (option I).

## References

Andreux, M., du Terrail, J. O., Beguier, C., and Tramel, E. W. Siloed federated learning for multi-centric histopathology datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pp. 129–139. Springer, 2020.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

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

Balu, A., Jiang, Z., Tan, S. Y., Hedge, C., Lee, Y. M., and Sarkar, S. Decentralized deep learning using momentum-accelerated consensus. *arXiv preprint arXiv:2010.11166*, 2020.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. Randomized gossip algorithms. *IEEE transactions on information theory*, 52 (6):2508–2530, 2006.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of ImageNet as an alternative to the CIFAR datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Das, R., Acharya, A., Hashemi, A., Sanghavi, S., Dhillon, I. S., and Topcu, U. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061*, 2020.

Defazio, A. and Bottou, L. On the ineffectiveness of variance reduced optimization for deep learning. In *NeurIPS 2019 - Thirty-third Conference on Neural Information Processing Systems*, 2019.

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13:165–202, 2012.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Diao, E., Ding, J., and Tarokh, V. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TNkPBBYFkXg.

Gitman, I., Lang, H., Zhang, P., and Xiao, L. Understanding the role of momentum in stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pp. 9633–9643, 2019.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pp. 2350–2358. PMLR, 2021.

He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, 2019.

Hendrikx, H., Bach, F., and Massoulié, L. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 897–906. PMLR, 2019.

Hoffer, E., Banner, R., Golan, I., and Soudry, D. Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, pp. 2160–2170, 2018.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Karimireddy, S. P., Jaggi, M., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020a.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. In *International Conference on Machine Learning*, 2020b.

Kempe, D., Dobra, A., and Gehrke, J. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 482–491. IEEE, 2003.

Koloskova, A., Stich, S. U., and Jaggi, M. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *ICML 2019 - Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3479–3487. PMLR, 2019. URL http://proceedings.mlr.press/v97/koloskova19a.html.

Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. Decentralized deep learning with arbitrary communication compression. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=SkgGCkrKvH.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. U. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, 2020b.

Konecnỳ, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Kong, L., Lin, T., Koloskova, A., Jaggi, M., and Stich, S. U. Consensus control for decentralized deep learning. *ICML 2021 - International Conference on Machine Learning*, 2021.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Lee, J. D., Lin, Q., Ma, T., and Yang, T. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *arXiv preprint arXiv:1507.07595*, 2015.

Li, X., JIANG, M., Zhang, X., Kamp, M., and Dou, Q. Fedbn: Federated learning on non-{iid} features via local batch normalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6YEQUn0QICG.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5336–5346, 2017.

Lian, X., Zhang, W., Zhang, C., and Liu, J. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pp. 3043–3052. PMLR, 2018.

Lin, T., Kong, L., Stich, S., and Jaggi, M. Extrapolation for large-batch training in deep learning. In *International Conference on Machine Learning*, pp. 6094–6104. PMLR, 2020a.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, 2020b.

Lin, T., Stich, S. U., Patel, K. K., and Jaggi, M. Don't use large mini-batches, use local sgd. In *International Conference on Learning Representations*, 2020c. URL https://openreview.net/forum?id=B1eyO1BFPr.

Liu, H., Brock, A., Simonyan, K., and Le, Q. V. Evolving normalization-activation layers. *arXiv preprint arXiv:2004.02967*, 2020.

Lu, S., Zhang, X., Sun, H., and Hong, M. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pp. 315–321. IEEE, 2019.

Lu, Y. and De Sa, C. Moniqua: Modulo quantized communication in decentralized sgd. In *International Conference on Machine Learning*, 2020.

Lucas, J., Sun, S., Zemel, R., and Grosse, R. Aggregated momentum: Stability through passive damping. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Syxt5oC5YQ.

Luo, P., Wang, X., Shao, W., and Peng, Z. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlLKjR9FQ.

Ma, J. and Yarats, D. Quasi-hyperbolic momentum and adam for deep learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1fUpoR5FQ.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Mosbach, M., Andriushchenko, M., and Klakow, D. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nzpLWnVAyah.

Nadiradze, G., Sabour, A., Alistarh, D., Sharma, A., Markov, I., and Aksenov, V. Swarmsgd: Scalable decentralized sgd with local updates. *arXiv preprint arXiv:1910.12308*, 2020.

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

Neglia, G., Xu, C., Towsley, D., and Calbi, G. Decentralized gradient methods: does topology matter? In *AISTATS*, 2020.

Pan, T., Liu, J., and Wang, J. D-spider-sfo: A decentralized optimization algorithm with faster convergence rate for nonconvex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1619–1626, 2020.

Pu, S. and Nedić, A. Distributed stochastic gradient tracking methods. *Mathematical Programming*, pp. 1–49, 2020.

Rosenbrock, H. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, 2017.

Scaman, K., Bach, F., Bubeck, S., Massoulié, L., and Lee, Y. T. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pp. 2740–2749, 2018.

Shi, W., Ling, Q., Wu, G., and Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Singh, N., Data, D., George, J., and Diggavi, S. Squarm-sgd: Communication-efficient momentum sgd for decentralized optimization. *arXiv preprint arXiv:2005.07041*, 2020.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Taheri, H., Mokhtari, A., Hassani, H., and Pedarsani, R. Quantized decentralized stochastic learning over directed graphs. In *International Conference on Machine Learning*, pp. 9324–9333. PMLR, 2020.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Tang, H., Gan, S., Zhang, C., Zhang, T., and Liu, J. Communication compression for decentralized training. volume 31, pp. 7652–7662, 2018a.

Tang, H., Lian, X., Yan, M., Zhang, C., and Liu, J. $d^2$: Decentralized training over decentralized data. In *International Conference on Machine Learning*, 2018b.

Tsitsiklis, J. N. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

Vogels, T., Karimireddy, S. P., and Jaggi, M. Powergossip: Practical low-rank communication compression in decentralized deep learning. In *NeurIPS 2020 - Thirty-fourth Conference on Neural Information Processing Systems*, 2020.

Wang, J., Sahu, A. K., Yang, Z., Joshi, G., and Kar, S. Matcha: Speeding up decentralized sgd via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, 2020a.

Wang, J., Sahu, A. K., Joshi, G., and Kar, S. Exploring the error-runtime trade-off in decentralized optimization. 2020b.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*, 2020c. URL https://openreview.net/forum?id=SkxJ8REYPH.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Xiao, L. and Boyd, S. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

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

Yan, Y., Yang, T., Li, Z., Lin, Q., and Yang, Y. A unified analysis of stochastic momentum methods for deep learning. In *IJCAI*, pp. 2955–2961, 2018. URL https://doi.org/10.24963/ijcai.2018/410.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, 2019.

Yuan, K. and Alghunaim, S. A. Removing data heterogeneity influence enhances network topology dependence of decentralized sgd. *arXiv preprint arXiv:2105.08023*, 2021.

Yuan, K., Alghunaim, S. A., Ying, B., and Sayed, A. H. On the influence of bias-correction on distributed stochastic optimization. *IEEE Transactions on Signal Processing*, 68:4352–4367, 2020a.

Yuan, K., Xu, W., and Ling, Q. Can primal methods outperform primal-dual methods in decentralized dynamic optimization? *IEEE Transactions on Signal Processing*, 68:4466–4480, 2020b.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pp. 7252–7261. PMLR, 2019.

Zhang, G., Wang, C., Xu, B., and Grosse, R. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=B1lz-3Rct7.

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S. J., Kumar, S., and Sra, S. Why adam beats sgd for attention models. In *NeurIPS 2020 - Thirty-fourth Conference on Neural Information Processing Systems*, 2019b.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

# Contents of Appendix

## A. Detailed Experimental Setup

### A.1. Visualization for Communication Topologies

Figure 7 visualizes the Social Network topology we evaluated in the main paper.

### A.2. Visualization of Non-IID Local Data

**The synthetic formulation of non-i.i.d. client data.** We re-iterate the partition scheme introduced and stated in Yurochkin et al. (2019); Hsu et al. (2019) for completeness reasons.

Assume every client training example is drawn independently with class labels following a categorical distribution over $M$ classes parameterized by a vector $\mathbf{q}$ ($q_i \geq 0, i \in [1, M]$ and $\|\mathbf{q}\|_1 = 1$). To synthesize client non-i.i.d. local data distributions, we draw $\alpha \sim \text{Dir}(\alpha \mathbf{p})$ from a Dirichlet distribution, where $\mathbf{p}$ characterizes a prior class distribution over $M$ classes, and $\alpha > 0$ is a concentration parameter controlling the identicalness among clients. With $\alpha \to \infty$, all clients have identical distributions to the prior; with $\alpha \to 0$, each client holds examples from only one random class.

To better understand the local data distribution for the datasets we considered in the experiments, in Figure 8 we visualize the partition results of CIFAR-10 and ImageNet(-32) for various degrees of non-i.i.d.-ness and network scales; in Figure 9, we visualize the partitioned local data on 16 clients with $\alpha = \{10, 1, 0.1\}$ for AG News and SST-2.

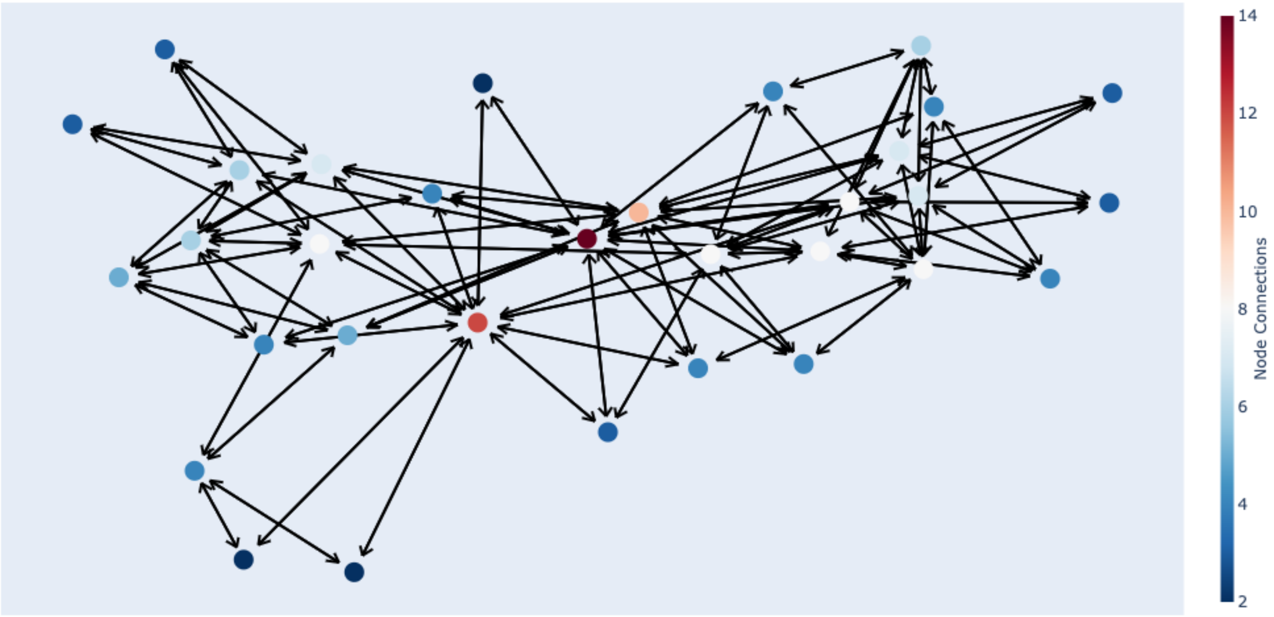
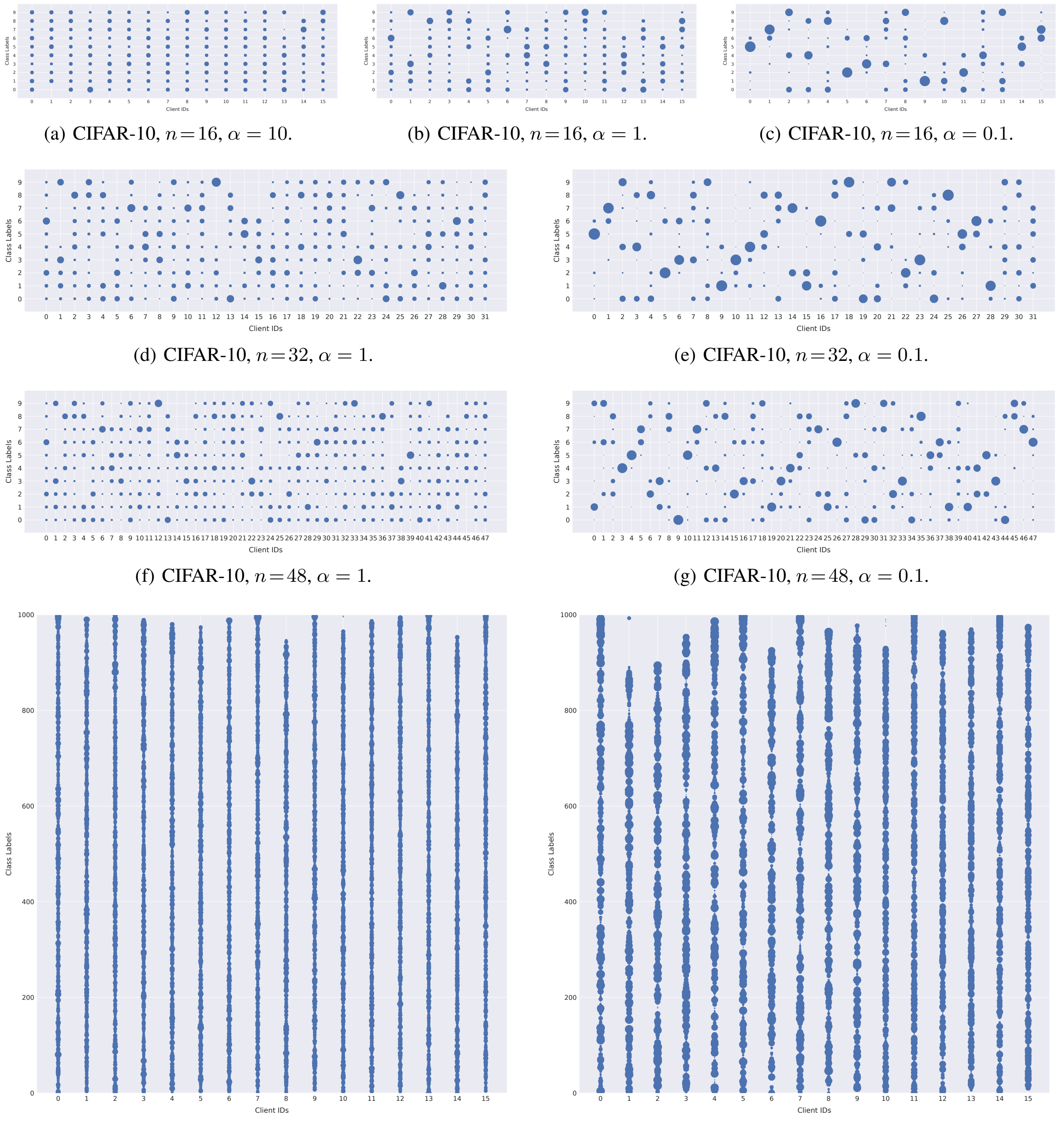


Figure 7: The visualization of the examined social topology (generated from "networkx.generators.social.davis_southern_women_graph").



(a) CIFAR-10, $n = 16$, $\alpha = 10$.

(b) CIFAR-10, $n = 16$, $\alpha = 1$.

(c) CIFAR-10, $n = 16$, $\alpha = 0.1$.

(d) CIFAR-10, $n = 32$, $\alpha = 1$.

(e) CIFAR-10, $n = 32$, $\alpha = 0.1$.

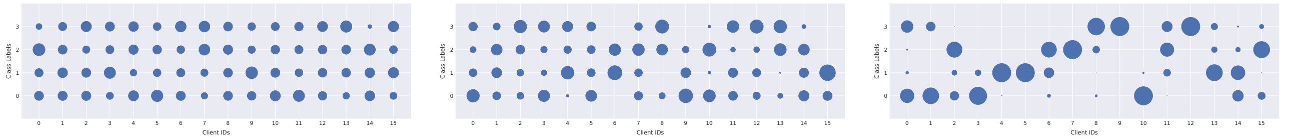
(f) CIFAR-10, $n = 48$, $\alpha = 1$.

(g) CIFAR-10, $n = 48$, $\alpha = 0.1$.

(h) ImageNet, $n = 16$, $\alpha = 1$.

(i) ImageNet, $n = 16$, $\alpha = 0.1$.

Figure 8: Illustration of # of samples per class allocated to each client (indicated by dot sizes), for different Dirichlet distribution $\alpha$ values on CV datasets.



(a) AG News, $n = 16$, $\alpha = 10$.

(b) AG News, $n = 16$, $\alpha = 1$.

(c) AG News, $n = 16$, $\alpha = 0.1$.

Figure 9: Illustration of # of samples per class allocated to each client (indicated by dot sizes), for different Dirichlet distribution $\alpha$ values on NLP datasets.

## B. Detailed Algorithm Description and Connections

### B.1. Detailed Algorithm Description

The variant of Adam with the idea of quasi-global momentum is detailed in Algorithm 2.

**Algorithm 2** QG-DAdam. $\hat{\mathbf{m}}_i^{(0)}, \hat{\mathbf{v}}_i^{(0)}$ are initialized as $\mathbf{0}$ for all workers.

```
procedure
    for $t \in \{1, \dots, T\}$ do
        sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
        $\mathbf{m}_i^{(t)} = \beta_1 \hat{\mathbf{m}}_i^{(t-1)} + (1-\beta_1)\hat{\mathbf{g}}_i^{(t)}$
        $\mathbf{v}_i^{(t)} = \beta_2 \hat{\mathbf{v}}_i^{(t-1)} + (1-\beta_2)\hat{\mathbf{g}}_i^{(t)} \odot \hat{\mathbf{g}}_i^{(t)}$
        $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta \frac{\mathbf{m}_i^{(t)}}{\sqrt{\mathbf{v}_i^{(t)}} + \epsilon}$
        $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+\frac{1}{2})}$
        $\mathbf{d}_i^{(t)} = \mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}$
        $\hat{\mathbf{d}}_i^{(t)} = \frac{\mathbf{d}_i^{(t)}}{\left\| \mathbf{d}_i^{(t)} \right\|_2}$
        $\hat{\mathbf{m}}_i^{(t)} = \beta_1 \hat{\mathbf{m}}_i^{(t-1)} + (1-\beta_1)\hat{\mathbf{d}}_i^{(t)}$
        $\hat{\mathbf{v}}_i^{(t)} = \beta_2 \hat{\mathbf{v}}_i^{(t-1)} + (1-\beta_2)\hat{\mathbf{d}}_i^{(t)} \odot \hat{\mathbf{d}}_i^{(t)}$
    return $\mathbf{x}_i^{(T)}$
```

**Algorithm 3** Multiple-step variant of QG-DSGDm. $\mathbf{m}_i^{(0)} = \hat{\mathbf{m}}_i^{(0)} := \mathbf{0}$. $\tau$ is the number of local steps.

```
procedure WORKER-$i$
    for $t \in \{1, \dots, T\}$ do
        sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
        $\mathbf{m}_i^{(t)} = \beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}$
        $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta \mathbf{m}_i^{(t)}$
        $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+\frac{1}{2})}$
        if $\mathrm{mod}(t, \tau) \neq 0$ then
            $\hat{\mathbf{m}}_i^{(t)} = \hat{\mathbf{m}}_i^{(t-1)}$
        else
            $\mathbf{d}_i^{(t)} = \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}$
            $\hat{\mathbf{m}}_i^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu)\mathbf{d}_i^{(t)}$
    return $\mathbf{x}_i^{(T)}$
```

Algorithm 4 depicts the general procedure of MimeLite in Karimireddy et al. (2020a). For SGDm, the update step $\mathcal{U}$ and the tracking step $\mathcal{V}$ follow

$$\mathcal{U}\left(\nabla F_i(\mathbf{y}_i, \xi), \mathbf{s}\right) := (1-\beta)\nabla F_i(\mathbf{y}_i, \xi) + \beta \mathbf{s}$$

$$\mathcal{V}\left(\frac{1}{|\mathcal{S}|}\sum_{i \in \mathcal{S}} \nabla f_i(\mathbf{x}), \mathbf{s}\right) := (1-\beta)\frac{1}{|\mathcal{S}|}\sum_{i \in \mathcal{S}} \nabla f_i(\mathbf{x}) + \beta \mathbf{s}\,.$$

Algorithm 5 shows the pseudocode of SlowMo (Wang et al., 2020c). For our evaluation in Table 5, we follow the hyper-parameter suggestion mentioned in Wang et al. (2020c): for CIFAR-10 dataset, we set $\alpha = 1, \tau = 12, \beta = 0.7$.

**Algorithm 4** MimeLite (Karimireddy et al., 2020a).

```
procedure
    for each round $t \in [T]$ do
        sample subset $\mathcal{S}$ of clients
        communicate $(\mathbf{x}, \mathbf{s})$ to all clients $i \in \mathcal{S}$
        for client $i \in \mathcal{S}$ in parallel do
            initialize local model $\mathbf{y}_i \leftarrow \mathbf{x}$
            for client $i \in \mathcal{S}$ in parallel do
                for $k \in [\tau]$ do
                    sample mini-batch $\xi$ from local data
                    $\mathbf{y}_i \leftarrow \mathbf{y}_i - \eta \mathcal{U}\left(\nabla F_i(\mathbf{y}_i, \xi), \mathbf{s}\right)$
            compute full local-batch gradient $\nabla f_i(\mathbf{x})$
            communicate $(\mathbf{y}_i, \nabla f_i(\mathbf{x}))$
        $\mathbf{s} \leftarrow \mathcal{V}\left(\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(\mathbf{x}), \mathbf{s}\right)$        ▷ update optimization statistics
        $\mathbf{x} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathbf{y}_i$        ▷ update server parameters
    return $\mathbf{x}_T$
```

**Algorithm 5** SlowMo (Wang et al., 2020c). $\mathbf{d}_{i,k}^{(t)}$ indicates the local update direction for communication round $t$ at local update steps $k$.

```
procedure
    for $t \in [T]$ at worker-$i$ in parallel do
        Maintain/Average base optimizer buffers
        for $k \in [\tau]$ do
            Base optimizer step: $\mathbf{x}_{i,k+1}^{(t)} = \mathbf{x}_{i,k}^{(t)} - \gamma^{(t)} \mathbf{d}_{i,k}^{(t)}$
        Exact-Average: $\mathbf{x}_{\tau}^{(t)} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{i,\tau}^{(t)}$
        Update slow momentum: $\mathbf{m}^{(t+1)} = \beta \mathbf{m}^{(t)} + \frac{1}{\gamma^{(t)}} (\mathbf{x}_{i,0}^{(t)} - \mathbf{x}_{\tau}^{(t)})$
        Update outer iterates: $\mathbf{x}_{i,0}^{(t+1)} = \mathbf{x}_{i,0}^{(t)} - \alpha \gamma^{(t)} \mathbf{m}^{(t+1)}$
    return $\mathbf{x}_T$
```

### B.2. Difference between DMSGD and QG-DSGDm

We first re-iterate DMSGD (Balu et al., 2020) in Algorithm 6 with slightly adjusted notations.

**Algorithm 6** Original formulation of DMSGD. $\hat{\mathbf{m}}_i^{(0)}$ are initialized as $\mathbf{0}$ for all workers.

```
procedure WORKER-$i$
    for $t \in \{1, \dots, T\}$ do
        $\mathbf{v}_i^{(t)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t)}$                                                  ▷ Consensus step
        $\hat{\mathbf{m}}_i^{(t)} = \mu(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t-1)}) + (1-\mu)(\mathbf{v}_i^{(t)} - \mathbf{v}_i^{(t-1)})$          ▷ Momentum step
        sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
        $\mathbf{x}_i^{(t+1)} = \mathbf{v}_i^{(t)} - \eta \mathbf{g}_i^{(t)} + \beta \hat{\mathbf{m}}_i^{(t)}$                     ▷ Option I of local gradient step
        $\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} - \eta \mathbf{g}_i^{(t)} + \beta \hat{\mathbf{m}}_i^{(t)}$                     ▷ Option II of local gradient step
```

By re-organizing, we can further simplify Algorithm 6 to Algorithm 7.

**Algorithm 7** Re-organized formulation of DMSGD. $\hat{\mathbf{m}}_i^{(0)}$ are initialized as $\mathbf{0}$ for all workers.

```
procedure WORKER-$i$
    for $t \in \{1, \dots, T\}$ do
        sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t-\frac{1}{2})}, \xi_i^{(t)})$
        $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta(\beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)})$                  ▷ Option I of local gradient step
        $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t-\frac{1}{2})} - \eta(\beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)})$            ▷ Option II of local gradient step
        $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+\frac{1}{2})}$                                    ▷ Consensus step
        $\hat{\mathbf{m}}_i^{(t)} = \mu(\mathbf{x}_i^{(t-\frac{1}{2})} - \mathbf{x}_i^{(t+\frac{1}{2})}) + (1-\mu)(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)})$   ▷ Momentum step
```

For a fair comparison, we unify Algorithm 7 with QG-DSGDm, as in Algorithm 8 (we slightly abuse the notations for comparison purpose).

**Algorithm 8** DMSGD v.s. QG-DSGDm. $\hat{\mathbf{m}}_i^{(0)}$ are initialized as $\mathbf{0}$ for all workers.

```
procedure WORKER-$i$
    for $t \in \{1, \dots, T\}$ do
        sample $\xi_i^{(t)}$ and compute $\mathbf{g}_i^{(t)} = \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$
        $\mathbf{x}_i^{(t+\frac{1}{2})} = \mathbf{x}_i^{(t)} - \eta(\beta \hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)})$
        $\mathbf{x}_i^{(t+1)} = \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij} \mathbf{x}_j^{(t+\frac{1}{2})}$
        $\hat{\mathbf{m}}_i^{(t)}$ is determined by the algorithm.
    return $\mathbf{x}_i^{(T)}$
```

Note that in Algorithm 8 (slightly different from the $\hat{\mathbf{m}}^{(t)}$ in Algorithm 7), $\hat{\mathbf{m}}_i^{(t)}$ in DMSGD is defined as

$$\hat{\mathbf{m}}_i^{(t)} = \frac{\mu(\mathbf{x}_i^{(t-\frac{1}{2})} - \mathbf{x}_i^{(t+\frac{1}{2})}) + (1-\mu)(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)})}{\eta} \,,$$

while for QG-DSGDm, we have

$$\hat{\mathbf{m}}_i^{(t)} = \mu \hat{\mathbf{m}}_i^{(t-1)} + (1-\mu) \frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta} \,.$$

Thus, for option I of DMSGD, we have

$$
\begin{aligned}
\hat{\mathbf{m}}_i^{(t)} &= \frac{\mu(\mathbf{x}_i^{(t-\frac{1}{2})} - \mathbf{x}_i^{(t+\frac{1}{2})}) + (1-\omega)(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)})}{\eta} \\
&= \frac{\mu}{\eta}\left(\left(\mathbf{x}_i^{(t-1)} - \eta(\beta\hat{\mathbf{m}}_i^{(t-2)} + \mathbf{g}_i^{(t-1)})\right) - \left(\mathbf{x}_i^{(t)} - \eta(\beta\hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)})\right)\right) + (1-\mu)\frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta} \\
&= \mu\left(\frac{\mathbf{x}_i^{(t-1)} - \mathbf{x}_i^{(t)}}{\eta} - \left(\left(\beta\hat{\mathbf{m}}_i^{(t-2)} + \mathbf{g}_i^{(t-1)}\right) - \left(\beta\hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}\right)\right)\right) + (1-\mu)\frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta} \\
&= \mu\left(\frac{\mathbf{x}_i^{(t-1)} - \mathbf{x}_i^{(t)}}{\eta} - \beta(\hat{\mathbf{m}}_i^{(t-2)} - \hat{\mathbf{m}}_i^{(t-1)}) - (\mathbf{g}_i^{(t-1)} - \mathbf{g}_i^{(t)})\right) + (1-\mu)\frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta} \\
&= \mu\left(\beta\hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)} + \frac{\mathbf{x}_i^{(t-1)} - \mathbf{x}_i^{(t)}}{\eta} - \beta\hat{\mathbf{m}}_i^{(t-2)} - \mathbf{g}_i^{(t-1)}\right) + (1-\mu)\frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}\,,
\end{aligned}
$$

for option II of DMSGD, we have

$$
\begin{aligned}
\hat{\mathbf{m}}_i^{(t)} &= \frac{\mu(\mathbf{x}_i^{(t-\frac{1}{2})} - \mathbf{x}_i^{(t+\frac{1}{2})}) + (1-\omega)(\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)})}{\eta} \\
&= \mu\left(\beta\hat{\mathbf{m}}_i^{(t-1)} + \mathbf{g}_i^{(t)}\right) + (1-\mu)\frac{\mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t+1)}}{\eta}\,.
\end{aligned}
$$

It is obvious that the design of DMSGD is different from our QG-DSGDm:

- The update scheme on the momentum buffer $\hat{\mathbf{m}}_i$ is different, as illustrate above.
- DMSGD is based on the heavy-ball momentum, while our scheme can generalize to heavy ball momentum SGD, Nesterov momentum SGD, and even Adam variants.

### B.3. Connections Between SGDm and QG-SGDm

#### B.3.1. Connections Between SGDm and QG-DSGDm

Note our scheme QG-DSGDm on the single worker case (i.e. QG-SGDm) has the following equation:

$$
\begin{aligned}
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left(\beta\mathbf{m}^{(t-1)} + \mathbf{g}^{(t)}\right) \\
\mathbf{m}^{t} &= \mu\mathbf{m}^{(t-1)} + (1-\mu)\frac{\mathbf{x}^{(t)} - \mathbf{x}^{(t+1)}}{\eta} = \left(\mu + (1-\mu)\beta\right)\mathbf{m}^{(t-1)} + (1-\mu)\mathbf{g}^{(t)}\,.
\end{aligned}
$$

By letting $\hat{\mathbf{m}}^{(t)} := \frac{\mathbf{m}^{(t)}}{1-\mu}$, we have

$$
\begin{aligned}
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left(\beta(1-\mu)\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)}\right) \\
\hat{\mathbf{m}}^{(t)} &= \left(\mu + (1-\mu)\beta\right)\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)}\,.
\end{aligned}
$$

We further let $\hat{\beta} := \mu + (1-\mu)\beta$, then we have

$$
\begin{aligned}
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left(\hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)} + \left(\beta(1-\mu) - \hat{\beta}\right)\hat{\mathbf{m}}^{(t-1)}\right) = \mathbf{x}^{(t)} - \eta\left(\hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)} - \mu\hat{\mathbf{m}}^{(t-1)}\right) \\
\hat{\mathbf{m}}^{(t)} &= \hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)}\,.
\end{aligned}
$$

By re-organizing, we have

$$
\begin{aligned}
\hat{\mathbf{m}}^{(t)} &= \hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)} \\
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left(\hat{\mathbf{m}}^{(t)} - \mu\hat{\mathbf{m}}^{(t-1)}\right) = \mathbf{x}^{(t)} - \eta\left(\hat{\mathbf{m}}^{(t)} - \mu\hat{\mathbf{m}}^{(t-1)}\right) = \mathbf{x}^{(t)} - \eta\left((1-\frac{\mu}{\hat{\beta}})\hat{\mathbf{m}}^{(t)} + \frac{\mu}{\hat{\beta}}\mathbf{g}^{(t)}\right)\,,
\end{aligned} \tag{5}
$$

which recovers the QHM (Gitman et al., 2019).

Comparing to the case of SGD with Heavy-ball Momentum (SGDm), where

$$\begin{aligned}\hat{\mathbf{m}}^{(t)} &= \beta \hat{\mathbf{m}}^{(t-1)} + \mathbf{g}^{(t)} \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta \hat{\mathbf{m}}^{(t)} \,,\end{aligned}$$

we can witness that SGDm is only a special case of (5) (when $\mu = 0$).

### B.3.2. Connections Between SGDm-N and QG-SGDm

First note that our simplified version of QG-DSGDm (i.e. QG-SGDm) can recover the QHM (Gitman et al., 2019), as illustrated in Appendix B.3.1. Furthermore, as pointed out in Gitman et al. (2019) that, the QHM is indeed equivalent to the original SGDm-N with re-scaling of $\eta \to \eta/(1-\beta)$. Therefore, we can argue that our simplified version of QG-DSGDm (i.e. QG-SGDm or QHM) is equivalent to the original SGDm-N with re-scaling of $\eta \to \eta/(1-\beta)$.

For the reason of completeness, we include the derivatives below.

First of all, SGD with Nesterov Momentum (SGDm-N) can be rewritten as

$$\begin{aligned}\mathbf{x}^{(t+\frac{1}{2})} &= \mathbf{x}^{(t)} + \beta \mathbf{m}^{(t-1)} \\ \mathbf{m}^{(t)} &= \beta \mathbf{m}^{(t-1)} - \eta \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t+\frac{1}{2})} - \eta \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) \,,\end{aligned}$$

and in PyTorch, we instead have

$$\begin{aligned}\mathbf{x}^{(t+\frac{1}{2})} &= \mathbf{x}^{(t)} + \beta \mathbf{m}^{(t-1)} \\ \mathbf{m}^{(t)} &= \beta \mathbf{m}^{(t-1)} + \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta \mathbf{m}^{(t)} \,,\end{aligned} \tag{6}$$

where the above equations are equivalent for the constant $\beta$.

Then we reiterate the derivatives in Gitman et al. (2019) below, from SGDm-N to QHM (which is equivalently Equation (5)). The SGDm-N in Gitman et al. (2019) follows

$$\begin{aligned}\mathbf{m}^{(t)} &= \beta \mathbf{m}^{(t-1)} - \eta \nabla f\left(\mathbf{x}^{(t)} + \beta \mathbf{m}^{(t-1)}\right) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} + \mathbf{m}^{(t)} \,,\end{aligned}$$

where we can move the learning rate out of the momentum into the iterates update:

$$\begin{aligned}\mathbf{m}^{(t)} &= \beta \mathbf{m}^{(t-1)} + \nabla f\left(\mathbf{x}^{(t)} - \eta\beta \mathbf{m}^{(t-1)}\right) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta \mathbf{m}^{(t)} \,,\end{aligned} \tag{7}$$

where the above two methods produce the same sequence of iterates $\mathbf{x}^{(t)}$ if $\mathbf{m}^{(0)}$ is initialized at $\mathbf{0}$. The second equation (7) is equivalent to the Pytorch implementation in Equation (6).

Let's normalize the momentum update by $1-\beta$:

$$\begin{aligned}\mathbf{m}^{(t)} &= \beta \mathbf{m}^{(t-1)} + (1-\beta) \nabla f\left(\mathbf{x}^{(t)} - \eta\beta \mathbf{m}^{(t-1)}\right) \\ \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta \mathbf{m}^{(t)} \,,\end{aligned}$$

which is equivalent to the un-normalized one by re-scaling $\eta \to \frac{\eta}{1-\beta}$ for constant parameters. We make a change of variables

$\mathbf{y}^{(t)} = \mathbf{x}^{(t)} - \eta\beta\mathbf{m}^{(t-1)}$,

$$\begin{aligned}
\mathbf{m}^{(t)} &= \beta\mathbf{m}^{(t-1)} + (1-\beta)\nabla f(\mathbf{y}^{(t)}) \\
\mathbf{y}^{(t+1)} &= \mathbf{x}^{(t+1)} - \eta\beta\mathbf{m}^{(t)} = \mathbf{x}^{(t)} - \eta\mathbf{m}^{(t)} - \eta\beta\mathbf{m}^{(t)} \\
&= \mathbf{y}^{(t)} + \eta\beta\mathbf{m}^{(t-1)} - \eta\mathbf{m}^{(t)} - \eta\beta\mathbf{m}^{(t)} \\
&= \mathbf{y}^{(t)} + \eta\left(\mathbf{m}^{(t)} - (1-\beta)\nabla f(\mathbf{y}^{(t)})\right) - \eta\mathbf{m}^{(t)} - \eta\beta\mathbf{m}^{(t)} \\
&= \mathbf{y}^{(t)} - \eta\left((1-\beta)\nabla f(\mathbf{y}^{(t)}) + \beta\mathbf{m}^{(t)}\right) ,
\end{aligned}$$

where by renaming $\mathbf{y}^{(t)}$ back to $\mathbf{x}^{(t)}$, we obtain the exact formula used in QHM update.

### B.3.3. The Simplification of QG-DSGDm-N on Single Worker Case

We can further simplify our scheme QG-DSGDm-N on the single worker case (and obtain QG-SGDm-N):

$$\begin{aligned}
\mathbf{x}^{(t+\frac{1}{2})} &= \mathbf{x}^{(t)} + \beta\mathbf{m}^{(t-1)} \\
\hat{\mathbf{m}}^{(t)} &= \beta\mathbf{m}^{(t-1)} + \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) \\
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\hat{\mathbf{m}}^{(t)} \\
\mathbf{m}^{(t)} &= \mu\mathbf{m}^{(t-1)} + (1-\mu)\frac{\mathbf{x}^{(t)} - \mathbf{x}^{(t+1)}}{\eta} .
\end{aligned}$$

We rewrite $\mathbf{m}^{(t)}$ as

$$\begin{aligned}
\mathbf{m}^{(t)} &= \mu\mathbf{m}^{(t-1)} + (1-\mu)\frac{\mathbf{x}^{(t)} - \mathbf{x}^{(t+1)}}{\eta} = \mu\mathbf{m}^{(t-1)} + (1-\mu)\hat{\mathbf{m}}^{(t)} \\
&= (\mu + \beta - \mu\beta)\,\mathbf{m}^{(t-1)} + (1-\mu)\nabla f(\mathbf{x}^{(t+\frac{1}{2})}) .
\end{aligned}$$

Similar to the treatment in Appendix B.3.1, by letting $\hat{\mathbf{m}}^{(t)} := \frac{\mathbf{m}^{(t)}}{1-\mu}$, we have

$$\begin{aligned}
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\hat{\mathbf{m}}^{(t)} = \mathbf{x}^{(t)} - \eta\left(\beta(1-\mu)\hat{\mathbf{m}}^{(t-1)} + \nabla f(\mathbf{x}^{(t+\frac{1}{2})})\right) \\
\hat{\mathbf{m}}^{(t)} &= (\mu + \beta - \mu\beta)\,\hat{\mathbf{m}}^{(t-1)} + \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) ,
\end{aligned}$$

and thus, in the end we have the following equations for QG-SGDm-N

$$\begin{aligned}
\mathbf{x}^{(t+\frac{1}{2})} &= \mathbf{x}^{(t)} + \beta(1-\mu)\hat{\mathbf{m}}^{(t-1)} \\
\hat{\mathbf{m}}^{(t)} &= \hat{\beta}\hat{\mathbf{m}}^{(t-1)} + \nabla f(\mathbf{x}^{(t+\frac{1}{2})}) \\
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \eta\left((1 - \frac{\mu}{\hat{\beta}})\hat{\mathbf{m}}^{(t)} + \frac{\mu}{\hat{\beta}}\nabla f(\mathbf{x}^{(t+\frac{1}{2})})\right) ,
\end{aligned}$$

where $\hat{\beta} := \mu + (1-\mu)\beta$ and we can recover SGDm-N by setting $\mu = 0$.

## C. Global Convergence Rate Proofs

We reiterate the update scheme of QG-DSGDm in a matrix form:

$$\begin{aligned}\mathbf{X}^{(t+1)} &= \mathbf{W}\left(\mathbf{X}^{(t)} - \eta\left(\beta\mathbf{M}^{(t)} + \mathbf{G}^{(t)}\right)\right)\\ \mathbf{M}^{(t+1)} &= \mu\mathbf{M}^{(t)} + (1-\mu)\frac{\mathbf{X}^{(t)} - \mathbf{X}^{(t+1)}}{\eta}\\ &= \left(\mu + (1-\mu)\beta\mathbf{W}\right)\mathbf{M}^{(t)} + (1-\mu)\mathbf{W}\mathbf{G}^{(t)} + \frac{1-\mu}{\eta}(\mathbf{I}-\mathbf{W})\mathbf{X}^{(t)}\,,\end{aligned} \tag{8}$$

where our numerical experiments by default use $\mu = \beta$. For each matrix $\mathbf{Z}$, we define an averaged vector $\bar{\mathbf{z}} = \mathbf{Z}\frac{1}{n}\mathbf{1}$ and matrix $\bar{\mathbf{Z}} = \mathbf{Z}\frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Note that we use bold lower-case to indicate vectors and bold upper-case to denote matrices.

First, we state some standard definitions and regularity conditions.

**Assumption 2.** *We assume that the following hold:*

1. *The function $f(\mathbf{x})$ we are minimizing is lower bounded from below by $f^\star$, and each node's loss $f_i$ is smooth satisfying $\|\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})\| \le L\|\mathbf{y}-\mathbf{x}\|$.*
2. *The stochastic gradients within each node satisfies $\mathbb{E}\left[g_i(\mathbf{x})\right] = \nabla f_i(\mathbf{x})$ and $\mathbb{E}\left\|g_i(\mathbf{x}) - \nabla f_i(\mathbf{x})\right\|^2 \le \sigma^2$. The variance across the workers is also bounded as $\frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\right\|^2 \le \zeta^2$.*
3. *The mixing matrix is doubly stochastic where for the all ones vector $\mathbf{1}$, we have $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $\mathbf{W}^\top\mathbf{1} = \mathbf{1}$. 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 $\mathbb{E}_{\mathbf{W}}\left\|\mathbf{Z}\mathbf{W} - \bar{\mathbf{Z}}\right\|_F^2 \le (1-\rho)\left\|\mathbf{Z} - \bar{\mathbf{Z}}\right\|_F^2$.*

**Average parameters.** Let us examine the effect of updates in (8) on $\bar{\mathbf{x}}^{(t)}$ which is the parameters averaged across the nodes. Note that since $\mathbf{W}$ is doubly stochastic, we can simplify the updates as follows:

$$\begin{aligned}\bar{\mathbf{x}}^{(t+1)} &= \bar{\mathbf{x}}^{(t)} - \eta\left(\beta\bar{\mathbf{m}}^{(t)} + \bar{\mathbf{g}}^{(t)}\right), \text{ and}\\ \bar{\mathbf{m}}^{(t+1)} &= \mu\bar{\mathbf{m}}^{(t)} + (1-\mu)\frac{\bar{\mathbf{x}}^{(t)} - \bar{\mathbf{x}}^{(t+1)}}{\eta}\\ &= (1-(1-\mu)(1-\beta))\bar{\mathbf{m}}^{(t)} + (1-\mu)\bar{\mathbf{g}}^{(t)}\,.\end{aligned} \tag{9}$$

Here, $\bar{\mathbf{g}}^{(t)} := \frac{1}{n}\sum_{i=1}^n \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$ is the average of the stochastic gradients across the nodes.

**Virtual Sequence.** Now we define a virtual sequence of parameters $\{\hat{\mathbf{x}}^{(t)}\}$ which has a simple SGD style update, which will be easy to analyze and an error sequence:

$$\begin{aligned}\hat{\mathbf{x}}^{(t+1)} &= \hat{\mathbf{x}}^{(t)} - \frac{\eta}{1-\beta}\bar{\mathbf{g}}^{(t)}, \text{ and}\\ \bar{\mathbf{e}}^{(t)} &:= \hat{\mathbf{x}}^{(t)} - \bar{\mathbf{x}}^{(t)}\,.\end{aligned} \tag{10}$$

Our strategy for the analysis will be to analyze the virtual sequence $\{\hat{\mathbf{x}}^{(t)}\}$ and prove that the real sequence of iterates remains close i.e. that the $\mathbf{e}^{(t)}$ remains small.

**Single-step progress of virtual update.** We will show that every step we make some progress, but have to balance three sources of error: i) the stochastic error which depends on $\sigma^2$ due to using stochastic gradients, ii) consensus error which depends on $\mathbf{X}^t - \bar{\mathbf{X}}^t$, and finally iii) momentum error due to using momentum which depends on $\mathbf{e}^{(t)}$.

**Lemma C.1** (Non-convex one step progress)**.** *Given assumptions 2, the sequence of iterates generated by (8) using $\eta \le \frac{1-\beta}{4L}$ satisfy*

$$\mathbb{E}f(\hat{\mathbf{x}}^{(t+1)}) \le \mathbb{E}f(\hat{\mathbf{x}}^{(t)}) - \frac{\tilde{\eta}}{4}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 - \frac{\tilde{\eta}}{4}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} + \frac{3L^2\tilde{\eta}}{2}\left\|\mathbf{e}^{(t)}\right\|^2 + \frac{3L^2\tilde{\eta}}{n}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2\,,$$

*where we define $\tilde{\eta} := \frac{\eta}{1-\beta}$.*

*Proof.* Starting from the smoothness of $f$, we have

$$\begin{aligned}
\mathbb{E} f(\hat{\mathbf{x}}^{(t+1)}) &\leq \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) + \mathbb{E}\left\langle \nabla f(\hat{\mathbf{x}}^{(t)}), \hat{\mathbf{x}}^{(t+1)} - \hat{\mathbf{x}}^{(t)} \right\rangle + \frac{L}{2}\mathbb{E}\left\|\hat{\mathbf{x}}^{(t+1)} - \hat{\mathbf{x}}^{(t)}\right\|^2 \\
&= \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) - \frac{\eta}{1-\beta}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\langle \nabla f(\hat{\mathbf{x}}^{(t)}), \nabla f_i(\mathbf{x}_i^{(t)}) \right\rangle + \frac{L\eta^2}{(1-\beta)^2}\mathbb{E}\left\|\bar{\mathbf{g}}^{(t)}\right\|^2 \\
&\leq \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) - \tilde{\eta}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\langle \nabla f(\hat{\mathbf{x}}^{(t)}), \nabla f_i(\mathbf{x}_i^{(t)}) \right\rangle + L\tilde{\eta}^2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} \\
&= \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) + L\tilde{\eta}^2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} \\
&\quad - \frac{\tilde{\eta}}{2}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 - \frac{\tilde{\eta}}{2}\mathbb{E}\left\|\nabla f(\hat{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f(\hat{\mathbf{x}}^{(t)})\right\|^2 \\
&\leq \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) + (L\tilde{\eta}^2 - \tilde{\eta}/2)\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} \\
&\quad - \frac{\tilde{\eta}}{4}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\left\|\nabla f(\hat{\mathbf{x}}^{(t)}) - \nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tilde{\eta}}{2}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f(\hat{\mathbf{x}}^{(t)})\right\|^2 \\
&\leq \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) - \frac{\tilde{\eta}}{4}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} - \tilde{\eta}/4\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 \\
&\quad + \frac{3\tilde{\eta}}{2}\mathbb{E}\left\|\nabla f(\hat{\mathbf{x}}^{(t)}) - \nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tilde{\eta}}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f_i(\bar{\mathbf{x}}^{(t)})\right\|^2 .
\end{aligned}$$

In the last inequality we used our bound on the step-size that $\eta \leq \frac{1-\beta}{4L}$. In the rest of the inequalities, we repeatedly use the identity that $2ab = -a^2 - b^2 + (a-b)^2$. Finally, using the smoothness of the function $f$ and the definition $\tilde{\eta} := \frac{\eta}{1-\beta}$, we get

$$\mathbb{E} f(\hat{\mathbf{x}}^{(t+1)}) \leq \mathbb{E} f(\hat{\mathbf{x}}^{(t)}) - \frac{\tilde{\eta}}{4}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} + \frac{3L^2\tilde{\eta}}{2}\left\|\bar{\mathbf{x}}^{(t)} - \hat{\mathbf{x}}^{(t)}\right\|^2 + \frac{L^2\tilde{\eta}}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 .$$

Recalling the definition of $\mathbf{e}^{(t)}$ and $\mathbf{X}^{(t)}$ yields the lemma. □

**Lemma C.2** (Strongly-convex one step progress)**.** *Suppose that the set of functions $\{f_i\}$ are $\mu$-strongly convex in addition to assumptions 2. Then the sequence of iterates generated by (8) using $\eta \leq \frac{1-\beta}{4L}$ satisfy for $\tilde{\eta} := \frac{\eta}{1-\beta}$,*

$$\mathbb{E}\left\|\hat{\mathbf{x}}^{(t+1)} - \mathbf{x}^\star\right\|^2 \leq (1 - \mu\tilde{\eta}/2)\mathbb{E}\left\|\hat{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\|^2 + \frac{\tilde{\eta}^2\sigma^2}{n} - 3\tilde{\eta}/4(f(\bar{\mathbf{x}}^{(t)}) - f^\star) + 8L\tilde{\eta}\left\|\mathbf{e}^{(t)}\right\|^2 + \frac{5L\tilde{\eta}}{n}\left\|\mathbf{X}^t - \bar{\mathbf{X}}^t\right\|_F^2 .$$

*Proof.* Starting from the update rule for $\hat{\mathbf{x}}^{(t+1)}$, and expanding very similar to the steps performed in the previous lemma we get,

$$\begin{aligned}
\mathbb{E}\left\|\hat{\mathbf{x}}^{(t+1)}-\mathbf{x}^\star\right\|^2 &= \mathbb{E}\left\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\|^2 + 2\mathbb{E}\left\langle \hat{\mathbf{x}}^{(t+1)}-\hat{\mathbf{x}}^{(t)}, \hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\rangle + \mathbb{E}\left\|\hat{\mathbf{x}}^{(t+1)}-\hat{\mathbf{x}}^{(t)}\right\|^2 \\
&\leq \mathbb{E}\left\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\|^2 - \frac{2\tilde{\eta}}{n}\sum_{i=1}^n \left\langle \nabla f_i(\mathbf{x}_i^{(t)}), \hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\rangle \\
&\qquad + \frac{\tilde{\eta}^2}{n}\sum_{i=1}^n \mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f_i(\bar{\mathbf{x}}^{(t)})\right\|^2 + \tilde{\eta}^2\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{\tilde{\eta}^2\sigma^2}{n} \\
&\leq \mathbb{E}\left\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\|^2 - 2\tilde{\eta}\underbrace{\tfrac{1}{n}\sum_{i=1}^n \left\langle \nabla f_i(\mathbf{x}_i^{(t)}), \hat{\mathbf{x}}^{(t)}-\mathbf{x}^\star\right\rangle}_{\mathcal{T}_1} \\
&\qquad + \frac{\tilde{\eta}^2 L^2}{n}\sum_{i=1}^n \mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 + 2L\tilde{\eta}^2\mathbb{E}(f(\bar{\mathbf{x}}^{(t)}) - f^\star) + \frac{\tilde{\eta}^2\sigma^2}{n}\,.
\end{aligned}$$

We will examine the term $\mathcal{T}_1$ now. Using strong convexity and smoothness of each of the functions $\{f_i\}$, we have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n \langle \nabla f_i(\mathbf{x}_i), \mathbf{x}_i - \mathbf{x}^\star\rangle &\geq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}_i) - f(\mathbf{x}^\star) + \frac{\mu}{2}\frac{1}{n}\sum_{i=1}^n \|\mathbf{x}_i - \mathbf{x}^\star\|^2 \\
&\geq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}_i) - f(\mathbf{x}^\star) + \frac{\mu}{4}\|\hat{\mathbf{x}} - \mathbf{x}^\star\|^2 - \frac{\mu}{2n}\sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}\|^2 \\
&\geq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}_i) - f(\mathbf{x}^\star) + \frac{\mu}{4}\|\hat{\mathbf{x}} - \mathbf{x}^\star\|^2 - \frac{\mu}{n}\sum_{i=1}^n \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 - \mu\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2 \\
&\geq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}_i) - f(\mathbf{x}^\star) + \frac{\mu}{4}\|\hat{\mathbf{x}} - \mathbf{x}^\star\|^2 - \frac{L}{n}\sum_{i=1}^n \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 - L\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2\ ,
\end{aligned}$$

and

$$\frac{1}{n}\sum_{i=1}^n \langle \nabla f_i(\mathbf{x}_i), \bar{\mathbf{x}} - \mathbf{x}_i\rangle \geq f(\bar{\mathbf{x}}) - \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}_i) - \frac{L}{2n}\sum_{i=1}^n \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2\ ,$$

and finally,

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n \langle \nabla f_i(\mathbf{x}_i), \hat{\mathbf{x}} - \bar{\mathbf{x}}\rangle &= \left\langle \frac{1}{n}\sum_{i=1}^n \left(\nabla f_i(\mathbf{x}_i) \pm \nabla f_i(\bar{\mathbf{x}})\right), \hat{\mathbf{x}} - \bar{\mathbf{x}}\right\rangle \\
&\geq -\frac{1}{8L}\left\|\frac{1}{n}\sum_{i=1}^n (\nabla f_i(\mathbf{x}_i) \pm \nabla f_i(\bar{\mathbf{x}}))\right\|^2 - 2L\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2 \\
&\geq -\frac{1}{4Ln}\sum_{i=1}^n \|\nabla f_i(\mathbf{x}_i) - \nabla f_i(\bar{\mathbf{x}})\|^2 - \frac{1}{4L}\|\nabla f(\bar{\mathbf{x}})\|^2 - 2L\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2 \\
&\geq -\frac{L}{2n}\sum_{i=1}^n \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 - \frac{1}{2}(f(\bar{\mathbf{x}}) - f(\mathbf{x}^\star)) - 3L\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|^2\ .
\end{aligned}$$

Adding up the three inequalities together yields the following expression for the term $\mathcal{T}_1$

$$\frac{1}{n}\sum_{i=1}^n \left\langle \nabla f_i(\mathbf{x}_i^{(t)}), \hat{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\rangle \geq \frac{1}{2}(f(\bar{\mathbf{x}}^{(t)}) - f(\mathbf{x}^\star)) + \frac{\mu}{4}\left\|\hat{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\|^2 - \frac{2L}{n}\sum_{i=1}^n \left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 - 4L\left\|\hat{\mathbf{x}}^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2\ .$$

Plugging this back into the previous inequality and the using $\tilde{\eta} \leq \frac{1}{4L}$ finishes the proof of the lemma.

$$\begin{aligned}\mathbb{E}\left\|\hat{\mathbf{x}}^{(t+1)} - \mathbf{x}^\star\right\|^2 \leq (1-\tilde{\eta}\mu/2)\mathbb{E}\left\|\hat{\mathbf{x}}^{(t)} - \mathbf{x}^\star\right\|^2 - \tilde{\eta}(1-2L\tilde{\eta})\mathbb{E}(f(\bar{\mathbf{x}}^{(t)}) - f^\star) + \frac{\tilde{\eta}^2\sigma^2}{n} \\ + \frac{\tilde{\eta}^2L^2 + 4\tilde{\eta}L}{n}\sum_{i=1}^n \mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 + 8L\tilde{\eta}\left\|\hat{\mathbf{x}}^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 .\end{aligned}$$

□

**Bounding the consensus error.** We will now try to bound the consensus error $(\mathbf{X}^t - \bar{\mathbf{X}}^t)$ between the node's parameters and its average. During each step, we perform a diffusion step (communication with neighbors) which brings the parameters of the nodes closer to each other. However we also perform additional gradient/momentum steps which moves the distance away from each other.

**Lemma C.3** (One step consensus change)**.** *Given assumptions 2, the sequence of iterates generated by* (8) *using* $\eta \leq \frac{\rho}{7L}$ *satisfy for* $\tilde{\eta} := \frac{\eta}{1-\beta}$*,*

$$\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{t+1} - \bar{\mathbf{X}}^{t+1}\right\|_F^2 \leq \frac{(1-\rho/4)}{n}\mathbb{E}\left\|\mathbf{X}^t - \bar{\mathbf{X}}^t\right\|_F^2 + \frac{12\eta^2\zeta^2}{\rho} + 4(1-\rho)\eta^2\sigma^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2$$

*Proof.* Starting from the update step (8),

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{t+1} - \bar{\mathbf{X}}^{t+1}\right\|_F^2 &= \frac{1}{n}\mathbb{E}\left\|\mathbf{W}\left(\mathbf{X}^{(t)} - \eta\left(\beta\mathbf{M}^{(t)} + \mathbf{G}^{(t)}\right)\right) - \left(\bar{\mathbf{X}}^{(t)} - \eta\left(\beta\bar{\mathbf{M}}^{(t)} + \bar{\mathbf{G}}^{(t)}\right)\right)\right\|_F^2 \\ &\leq \frac{1-\rho}{n}\mathbb{E}\left\|\left(\mathbf{X}^{(t)} - \eta\left(\beta\mathbf{M}^{(t)} + \mathbf{G}^{(t)}\right)\right) - \left(\bar{\mathbf{X}}^{(t)} - \eta\left(\beta\bar{\mathbf{M}}^{(t)} + \bar{\mathbf{G}}^{(t)}\right)\right)\right\|_F^2 \\ &\leq \frac{1-\rho}{n}\mathbb{E}\left\|\left(\mathbf{X}^{(t)} - \eta\left(\beta\mathbf{M}^{(t)} + \mathbb{E}_t\left[\mathbf{G}^{(t)}\right]\right)\right) - \left(\bar{\mathbf{X}}^{(t)} - \eta\left(\beta\bar{\mathbf{M}}^{(t)} + \mathbb{E}_t\left[\bar{\mathbf{G}}^{(t)}\right]\right)\right)\right\|_F^2 \\ &\qquad + 4(1-\rho)\eta^2\sigma^2 \\ &\leq \frac{(1-\rho)(1+\rho/2)}{n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 \\ &\qquad + \frac{6\eta^2}{\rho n}\mathbb{E}\left\|\mathbb{E}_t\left[\mathbf{G}^{(t)}\right] - \mathbb{E}_t\left[\bar{\mathbf{G}}^{(t)}\right]\right\|_F^2 + 4(1-\rho)\eta^2\sigma^2 .\end{aligned}$$

Here we used the contractivity of the mixing matrix and Young's inequality. We can proceed as

$$\begin{aligned}\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{t+1} - \bar{\mathbf{X}}^{t+1}\right\|_F^2 &\leq \frac{(1-\rho/2)}{n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4(1-\rho)\eta^2\sigma^2 \\ &\qquad + \frac{6\eta^2}{\rho n}\mathbb{E}\left\|\mathbb{E}_t\left[\mathbf{G}^{(t)}\right] - \nabla f(\bar{\mathbf{x}}^{(t)})\right\|_F^2 \\ &= \frac{(1-\rho/2)}{n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4(1-\rho)\eta^2\sigma^2 \\ &\qquad + \frac{6\eta^2}{\rho n}\sum_{i=1}^n \mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{(t)}) \pm \nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 \\ &\leq \frac{(1-\rho/2)}{n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4(1-\rho)\eta^2\sigma^2 \\ &\qquad + \frac{12\eta^2}{\rho n}\sum_{i=1}^n \mathbb{E}\left\|\nabla f_i(\mathbf{x}_i^{(t)}) - \nabla f_i(\bar{\mathbf{x}}^{(t)})\right\|^2 + \frac{12\eta^2}{\rho n}\sum_{i=1}^n \mathbb{E}\left\|\nabla f_i(\bar{\mathbf{x}}^{(t)}) - \nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 \\ &\leq \frac{(1-\rho/2)}{n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{6\eta^2\beta^2}{\rho n}\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4(1-\rho)\eta^2\sigma^2 \\ &\qquad + \frac{12\eta^2L^2}{\rho n}\sum_{i=1}^n \mathbb{E}\left\|\mathbf{x}_i^{(t)} - \bar{\mathbf{x}}^{(t)}\right\|^2 + \frac{12\eta^2\zeta^2}{\rho} .\end{aligned}$$

Our assumption that the step size $\eta \leq \frac{\rho}{7L}$ ensures that $12\eta^2 L^2 \leq \rho^2/4$, finishing the proof. □

We will now try to bound the momentum error $(\mathbf{X}^t - \bar{\mathbf{X}}^t)$ between the momentum on each node and its average across nodes.

**Lemma C.4** (One step momentum change)**.** *Given assumptions 2, the sequence of iterates generated by (8) using momentum satisfying* $\frac{\beta}{1-\beta} \leq \frac{\rho}{21}$,

$$\frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{M}^{t+1} - \bar{\mathbf{M}}^{t+1}\right\|_F^2 \leq \left(\frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)} - \frac{6\eta^2\beta^2}{n\rho}\right)\mathbb{E}\left\|(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 + \frac{\rho}{8n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{\eta^2\rho\zeta^2}{8} + \frac{\eta^2\rho\sigma^2(1-\beta)}{8(1-\mu)}\,.$$

*Proof.* Starting from the update step (8) and proceeding similar to the previous lemma, we have

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}\left\|\mathbf{M}^{(t+1)} - \bar{\mathbf{M}}^{(t+1)}\right\|_F^2 \\
&\quad= \frac{1}{n}\mathbb{E}\left\|(\mu\mathbf{I} + (1-\mu)\beta\mathbf{W})\,(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}) + (1-\mu)\mathbf{W}(\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}) + \frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)}\right\|_F^2 \\
&\quad= \frac{1}{n}\mathbb{E}\left\|(\mu\mathbf{I} + (1-\mu)\beta\mathbf{W})\,(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}) + \frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)} + (1-\mu)\mathbf{W}(\mathbb{E}\left[\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}\right])\right\|_F^2 \\
&\qquad\quad + \frac{1}{n}\mathbb{E}\left\|(1-\mu)\mathbf{W}\left(\mathbf{G}^{(t)} - \mathbb{E}\left[\mathbf{G}^{(t)}\right] - (\bar{\mathbf{G}}^{(t)} - \mathbb{E}\left[\bar{\mathbf{G}}^{(t)}\right])\right)\right\|_F^2 \\
&\quad= \frac{1}{n}\mathbb{E}\left\|(\mu\mathbf{I} + (1-\mu)\beta\mathbf{W})\,(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}) + \frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)} + (1-\mu)\mathbf{W}(\mathbb{E}\left[\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}\right])\right\|_F^2 + 4\sigma^2 \\
&\quad\leq \frac{1}{n}\left(1 + \frac{(1-\mu)(1-\beta)}{1-(1-\mu)(1-\beta)}\right)\mathbb{E}\left\|(\mu\mathbf{I} + (1-\mu)\beta\mathbf{W})\,(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 + 4\sigma^2 \\
&\qquad\quad + \frac{1}{n}\left(1 + \frac{1-(1-\mu)(1-\beta)}{(1-\mu)(1-\beta)}\right)\mathbb{E}\left\|\frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)} + (1-\mu)\mathbf{W}\left(\mathbb{E}\left[\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}\right]\right)\right\|_F^2 .
\end{aligned}$$

Note that since $\mathbf{W} \prec \mathbf{I}$, we have $(\mu\mathbf{I} + (1-\mu)\beta\mathbf{W}) \prec (\mu + (1-\mu)\beta)\mathbf{I} = (1 - (1-\beta)(1-\mu))\mathbf{I}$. Further, since $-\mathbf{I} \prec \mathbf{W}$, we have $\mathbf{I} - \mathbf{W} \prec 2\mathbf{I}$. With these observations, we can continue

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}\left\|\mathbf{M}^{(t+1)} - \bar{\mathbf{M}}^{(t+1)}\right\|_F^2 \\
&\quad\leq \frac{1}{n}\left(1 + \frac{(1-\mu)(1-\beta)}{1-(1-\mu)(1-\beta)}\right)\mathbb{E}\left\|(1-(1-\mu)(1-\beta))\,(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 + 4\sigma^2 \\
&\qquad\quad + \frac{1}{n}\left(1 + \frac{1-(1-\mu)(1-\beta)}{(1-\mu)(1-\beta)}\right)\mathbb{E}\left\|\frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)} + (1-\mu)\mathbf{W}\left(\mathbb{E}\left[\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}\right]\right)\right\|_F^2 \\
&\quad\leq \frac{1}{n}\left(1-(1-\mu)(1-\beta)\right)\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4\sigma^2 \\
&\qquad\quad + \frac{1}{(1-\mu)(1-\beta)n}\mathbb{E}\left\|\frac{1-\mu}{\eta}(\mathbf{I} - \mathbf{W})\mathbf{X}^{(t)} + (1-\mu)\mathbf{W}\left(\mathbb{E}\left[\mathbf{G}^{(t)} - \bar{\mathbf{G}}^{(t)}\right]\right)\right\|_F^2 \\
&\quad\leq \frac{1}{n}\left(1-(1-\mu)(1-\beta)\right)\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4\sigma^2 \\
&\qquad\quad + \frac{4(1-\mu)}{(1-\beta)n\eta^2}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{2(1-\mu)}{(1-\beta)n}\mathbb{E}\left\|\mathbb{E}\left[\mathbf{G}^{(t)}\right] - \bar{\mathbf{G}}^{(t)}\right\|_F^2 .
\end{aligned}$$

From the proof of the previous lemma, we can simplify the last term as

$$\begin{aligned}
\frac{1}{n}\mathbb{E}\left\|\mathbf{M}^{(t+1)} - \bar{\mathbf{M}}^{(t+1)}\right\|_F^2 &\le \frac{1}{n}\left(1-(1-\mu)(1-\beta)\right)\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4\sigma^2 \\
&\qquad + \frac{4(1-\mu)}{n\eta^2(1-\beta)}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{2(1-\mu)}{n(1-\beta)}\mathbb{E}\left\|\mathbb{E}\left[\mathbf{G}^{(t)}\right] \pm \nabla f(\bar{\mathbf{X}}^{(t)}) - \bar{\mathbf{G}}^{(t)}\right\|_F^2 \\
&\le \frac{1}{n}\left(1-(1-\mu)(1-\beta)\right)\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4\sigma^2 \\
&\qquad + \frac{4(1-\mu)}{n\eta^2(1-\beta)}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{8(1-\mu)\zeta^2}{(1-\beta)} + \frac{4(1-\mu)L^2}{n(1-\beta)}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 \\
&= \frac{1}{n}\left(1-(1-\mu)(1-\beta)\right)\mathbb{E}\left\|\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)}\right\|_F^2 + 4\sigma^2 \\
&\qquad + \frac{4(1-\mu)(1+\eta^2L^2)}{n\eta^2(1-\beta)}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{8(1-\mu)\zeta^2}{(1-\beta)}
\end{aligned}$$

Multiplying both sides by $\frac{6\eta^2\beta^2}{\rho(1-\mu)(1-\beta)}$ yields

$$\begin{aligned}
&\frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{M}^{t+1} - \bar{\mathbf{M}}^{t+1}\right\|_F^2 \\
&\qquad \le \frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 - \frac{6\eta^2\beta^2}{n\rho}\mathbb{E}\left\|(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 \\
&\qquad\qquad + \frac{48\beta^2(1+L^2\eta^2)}{n\rho(1-\beta)^2}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{24\eta^2\beta^2\sigma^2}{\rho(1-\mu)(1-\beta)} + \frac{48\eta^2\beta^2\zeta^2}{\rho(1-\beta)^2} \\
&\qquad \le \frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 - \frac{6\eta^2\beta^2}{n\rho}\mathbb{E}\left\|(\mathbf{M}^{(t)} - \bar{\mathbf{M}}^{(t)})\right\|_F^2 \\
&\qquad\qquad + \frac{\rho}{8n}\mathbb{E}\left\|\mathbf{X}^{(t)} - \bar{\mathbf{X}}^{(t)}\right\|_F^2 + \frac{\eta^2\rho\zeta^2}{8} + \frac{\eta^2\rho\sigma^2(1-\beta)}{8(1-\mu)}\,.
\end{aligned}$$

The last step follows from our assumption that the momentum parameter that $\frac{\beta}{1-\beta} \le \frac{\rho}{21}$ and $\eta \le \frac{1}{7L}$. This ensures that $\frac{48\beta^2(1+L^2\eta^2)}{\rho(1-\beta)^2} \le \frac{49\beta^2}{\rho(1-\beta)^2} \le \frac{\rho}{8}$. □

We can now exactly describe the progress in consensus made each round.

**Lemma C.5** (One step consensus improvement)**.** *Given assumptions 2, the sequence of iterates generated by* (8) *using step-size* $\eta \le \frac{\rho}{7L}$ *and momentum* $\frac{\beta}{1-\beta} \le \frac{\rho}{21}$, *satisfy*

$$\begin{aligned}
&\frac{1}{n}\mathbb{E}\left\|\mathbf{X}^{t+1} - \bar{\mathbf{X}}^{t+1}\right\|_F^2 + \frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{M}^{t+1} - \bar{\mathbf{M}}^{t+1}\right\|_F^2 \\
&\qquad \le \frac{1-\rho/8}{n}\mathbb{E}\left\|\mathbf{X}^{t} - \bar{\mathbf{X}}^{t}\right\|_F^2 + \frac{6\eta^2\beta^2}{n\rho(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{M}^{t} - \bar{\mathbf{M}}^{t}\right\|_F^2 + \frac{13\eta^2\zeta^2}{\rho} + \frac{13\eta^2\sigma^2(2-\beta-\mu))}{(1-\mu)\rho}
\end{aligned}$$

*Proof.* Simply adding the results of Lemmas C.3 and C.4 gives the result. □

**Average and virtual sequences.** We will now bound the difference between the average and the virtual sequences $\mathbf{e}^{(t)}$ which recall was defined to be $\mathbf{e}^{(t)} = \hat{\mathbf{x}}^{(t)} - \bar{\mathbf{x}}^{(t)}$. The latter runs SGD with momentum whereas the former only runs SGD. We view the momentum terms as accumulating the gradient terms, delayed over time and hence the proof views SGDm as simply SGD run with a larger step-size.

**Lemma C.6** (One step error contraction)**.** *Given assumptions 2, the sequence of iterates generated by* (8) *satisfy*

$$\mathbb{E}\left\|\mathbf{e}^{(t+1)}\right\|^2 \le (1-(1-\mu)(1-\beta))\mathbb{E}\left\|\mathbf{e}^{(t)}\right\|^2 + \frac{2\tilde{\eta}^2\beta^2}{(1-\beta)(1-\mu)}\mathbb{E}\left\|\mathbb{E}_t[\bar{\mathbf{g}}^t]\right\|^2 + \tilde{\eta}^2\beta^2\sigma^2$$

*Proof.* By definition, $\hat{\mathbf{x}}^{(0)} = \bar{\mathbf{x}}^{(0)}$ and hence we have $\mathbf{e}^{(0)} = 0$. For $t \geq 0$, starting from the definition of the error term we have

$$
\begin{aligned}
\mathbf{e}^{(t+1)} &= \hat{\mathbf{x}}^{(t+1)} - \bar{\mathbf{x}}^{(t+1)} \\
&= \left(\hat{\mathbf{x}}^{(t)} - \frac{\eta}{1-\beta}\bar{\mathbf{g}}^{(t)}\right) - \left(\bar{\mathbf{x}}^{(t)} - \eta(\beta\bar{\mathbf{m}}^{(t)} + \bar{\mathbf{g}}^{(t)})\right) \\
&= \mathbf{e}^{(t)} - \eta\beta(\frac{1}{1-\beta}\bar{\mathbf{g}}^{(t)} - \bar{\mathbf{m}}^{(t)}) \\
&= \sum_{k=0}^{t} -\eta\beta(\frac{1}{1-\beta}\bar{\mathbf{g}}^{(k)} - \bar{\mathbf{m}}^{(k)})\,.
\end{aligned}
$$

Using the update of the average momentum (9), we can write

$$
\begin{aligned}
\mathbf{e}^{t+1} &= \sum_{k=0}^{(t)} -\eta\beta(\frac{1}{1-\beta}\bar{\mathbf{g}}^{(k)} - \bar{\mathbf{m}}^{(k)}) \\
&= \sum_{k=0}^{(t)} -\eta\beta\left(\frac{1}{1-\beta}\bar{\mathbf{g}}^{(k)} - \left((1-(1-\mu)(1-\beta))\bar{\mathbf{m}}^{(k-1)} + (1-\mu)\bar{\mathbf{g}}^{(k-1)}\right)\right) \\
&= (1-(1-\mu)(1-\beta))\sum_{k=0}^{t} -\eta\beta\left(\frac{1}{1-\beta}\bar{\mathbf{g}}^{(k-1)} - \bar{\mathbf{m}}^{(k-1)}\right) + \sum_{k=0}^{(t)} -\frac{\eta\beta}{1-\beta}\left(\bar{\mathbf{g}}^{(k)} - \bar{\mathbf{g}}^{(k-1)}\right) \\
&= (1-(1-\mu)(1-\beta))\mathbf{e}^{(t)} - \frac{\eta\beta}{1-\beta}\bar{\mathbf{g}}^{(t)}\,.
\end{aligned}
$$

By convention, we assume that vectors with negative indices are 0. Taking norms and expectations gives

$$
\begin{aligned}
\mathbb{E}\left\|\mathbf{e}^{(t+1)}\right\|^2 &= \mathbb{E}\left\|(1-(1-\mu)(1-\beta))\mathbf{e}^{(t)} - \frac{\eta\beta}{1-\beta}\bar{\mathbf{g}}^{(t)}\right\|^2 \\
&\leq \mathbb{E}\left\|(1-(1-\mu)(1-\beta))\mathbf{e}^{(t)} - \frac{\eta\beta}{1-\beta}\mathbb{E}_t[\bar{\mathbf{g}}^{(t)}]\right\|^2 + \frac{\eta^2\beta^2\sigma^2}{(1-\beta)^2} \\
&\leq (1-(1-\mu)(1-\beta))\mathbb{E}\left\|\mathbf{e}^{(t)}\right\|^2 + \frac{2\eta^2\beta^2}{(1-\beta)^3(1-\mu)}\mathbb{E}\left\|\mathbb{E}_t[\bar{\mathbf{g}}^{(t)}]\right\|^2 + \frac{\eta^2\beta^2\sigma^2}{(1-\beta)^2}\,.
\end{aligned}
$$

□

**Convergence rate for non-convex case.**

**Theorem C.7.** *Given assumptions 2, the sequence of iterates generated by (8) for step size* $\eta = \min\left(\frac{\rho}{7L}, \frac{1-\beta}{4L}, \frac{(1-\mu)(1-\beta)^2}{4\beta L}, \sqrt{\frac{4n(f(\bar{\mathbf{x}}^0)-f^\star)}{L\sigma^2 T}}\right)$ *and momentum parameter* $\frac{\beta}{1-\beta} \leq \frac{\rho}{21}$ *satisfies*

$$
\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\bar{\mathbf{x}}^t)\right\| \leq \mathcal{O}\Bigg(&\sqrt{\frac{L\sigma^2(f(\bar{\mathbf{x}}^0)-f^\star)}{nT}} + \\
&\sqrt[3]{L^2(f(\bar{\mathbf{x}}^0)-f^\star)^2\frac{\tilde{\zeta}^2}{\rho^2 T^2}} + \\
&\left(\frac{1}{\rho} + \frac{1}{1-\beta} + \frac{\beta}{(1-\mu)(1-\beta)^2}\right)\frac{L(f(\bar{\mathbf{x}}^0)-f^\star)}{T}\Bigg)
\end{aligned}
$$

*Proof.* Define $\tilde{\zeta}^2 := \zeta^2 + \sigma^2 \left(1 + \frac{1-\beta}{1-\mu}\right)$. Scaling Lemma C.5 by $\frac{24L^2\tilde{\eta}}{\rho}$ gives

$$\begin{aligned} \frac{24L^2\tilde{\eta}}{\rho n}\mathbb{E}\left\|\mathbf{X}^{t+1} - \bar{\mathbf{X}}^{t+1}\right\|_F^2 + \frac{144L^2\tilde{\eta}^3\beta^2(1-\beta)}{n\rho^2(1-\mu)}\mathbb{E}\left\|\mathbf{M}^{t+1} - \bar{\mathbf{M}}^{t+1}\right\|_F^2 \\ \leq \frac{24L^2\tilde{\eta}}{\rho n}\mathbb{E}\left\|\mathbf{X}^{t} - \bar{\mathbf{X}}^{t}\right\|_F^2 + \frac{144L^2\tilde{\eta}^3\beta^2(1-\beta)}{n\rho^2(1-\mu)}\mathbb{E}\left\|\mathbf{M}^{t} - \bar{\mathbf{M}}^{t}\right\|_F^2 \\ - \frac{3L^2\tilde{\eta}}{n}\mathbb{E}\left\|\mathbf{X}^{t} - \bar{\mathbf{X}}^{t}\right\|_F^2 + \frac{312L^2\tilde{\eta}^3(1-\beta)^2\tilde{\zeta}^2}{\rho^2} \end{aligned}$$

Scaling Lemma C.6 by $\frac{3L^2\tilde{\eta}}{2(1-\mu)(1-\beta)}$ gives

$$\begin{aligned} \frac{3L^2\tilde{\eta}}{2(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{e}^{(t+1)}\right\|^2 \leq \frac{3L^2\tilde{\eta}}{2(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{e}^{(t)}\right\|^2 - \frac{3L^2\tilde{\eta}}{2}\mathbb{E}\left\|\mathbf{e}^{(t)}\right\|^2 \\ + \frac{3L^2\tilde{\eta}^3\beta^2}{(1-\beta)^2(1-\mu)^2}\mathbb{E}\left\|\mathbb{E}_t[\bar{\mathbf{g}}^t]\right\|^2 + \frac{3L^2\tilde{\eta}^3\beta^2\sigma^2}{2(1-\mu)(1-\beta)} \end{aligned}$$

Finally Lemma C.1 gives

$$\mathbb{E}f(\hat{\mathbf{x}}^{(t+1)}) \leq \mathbb{E}f(\hat{\mathbf{x}}^{(t)}) - \frac{\tilde{\eta}}{4}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 - \frac{\tilde{\eta}}{4}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 + \frac{L\tilde{\eta}^2\sigma^2}{n} + \frac{3L^2\tilde{\eta}}{2}\left\|\mathbf{e}^{(t)}\right\|^2 + \frac{3L^2\tilde{\eta}}{n}\left\|\mathbf{X}^t - \bar{\mathbf{X}}^t\right\|_F^2 ,$$

Define

$$\Phi^t := \frac{24L^2\tilde{\eta}}{\rho n}\mathbb{E}\left\|\mathbf{X}^t - \bar{\mathbf{X}}^t\right\|_F^2 + \frac{144L^2\tilde{\eta}^3\beta^2(1-\beta)}{n\rho^2(1-\mu)}\mathbb{E}\left\|\mathbf{M}^t - \bar{\mathbf{M}}^t\right\|_F^2 + \frac{3L^2\tilde{\eta}}{2(1-\mu)(1-\beta)}\mathbb{E}\left\|\mathbf{e}^{(t)}\right\|^2 + \mathbb{E}[f(\bar{\mathbf{x}}^t) - f^\star]$$

Note that $\Phi^0 = \mathbb{E}[f(\bar{\mathbf{x}}^0)] - f^\star$ and that $\Phi^t \geq 0$ for any $t$. Then adding the three inequalities from the lemmas as described above gives

$$\begin{aligned} \Phi^{t+1} \leq \Phi^t - \frac{\tilde{\eta}}{4}\left\|\nabla f(\bar{\mathbf{x}}^{(t)})\right\|^2 + \left(\frac{3L^2\tilde{\eta}^3\beta^2}{(1-\beta)^2(1-\mu)^2} - \frac{\tilde{\eta}}{4}\right)\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{x}_i^{(t)})\right\|^2 \\ + \frac{L\tilde{\eta}^2\sigma^2}{n} + \frac{3L^2\tilde{\eta}^3\beta^2\sigma^2}{2(1-\mu)(1-\beta)} + \frac{312L^2\tilde{\eta}^3(1-\beta)^2\tilde{\zeta}^2}{\rho^2} \end{aligned}$$

Since, $\eta \leq \frac{(1-\mu)(1-\beta)^2}{4\beta L}$, we have that $\frac{3L^2\tilde{\eta}^3\beta^2}{(1-\beta)^2(1-\mu)^2} \leq \frac{\tilde{\eta}}{4}$. Rearranging the terms and averaging over $t$, we get

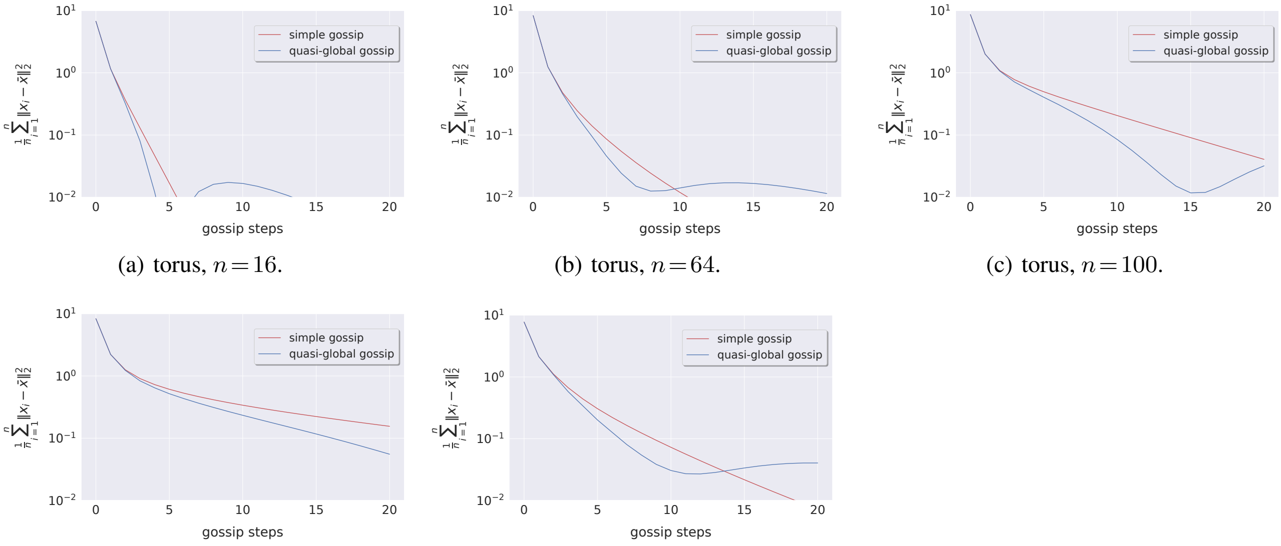
$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\mathbf{x}^{(t)})\right\|^2 &\leq \frac{4}{\tilde{\eta}T}(\Phi^0 - \Phi^T) + \frac{L\tilde{\eta}\sigma^2}{n} + \frac{6L^2\tilde{\eta}^2\beta^2\sigma^2}{(1-\mu)} + \frac{1248L^2\tilde{\eta}^2(1-\beta)^2\tilde{\zeta}^2}{\rho^2} \\ &\leq \frac{1}{\tilde{\eta}T}4(f(\bar{\mathbf{x}}^0) - f^\star) + \tilde{\eta}\left(\frac{L\sigma^2}{n}\right) + \tilde{\eta}^2\left(\frac{L^2\rho^2\sigma^2(1-\beta)}{(1-\mu)} + \frac{1248L^2(1-\beta)^2\tilde{\zeta}^2}{\rho^2}\right) . \\ &\leq \frac{1}{\tilde{\eta}T}4(f(\bar{\mathbf{x}}^0) - f^\star) + \tilde{\eta}\left(\frac{L\sigma^2}{n}\right) + \tilde{\eta}^2\left(\frac{L^2\sigma^2(1-\beta)}{(1-\mu)} + \frac{1248L^2\tilde{\zeta}^2}{\rho^2}\right) \\ &\leq \frac{1}{\tilde{\eta}T}4(f(\bar{\mathbf{x}}^0) - f^\star) + \tilde{\eta}\left(\frac{L\sigma^2}{n}\right) + \tilde{\eta}^2\left(\frac{1249L^2\tilde{\zeta}^2}{\rho^2}\right) . \end{aligned}$$

Choosing an appropriate steps-size $\eta$ proves the theorem. □

# D. Additional Results

## D.1. Results on Distributed Average Consensus Problem

Figure 10 illustrates the results for average consensus problem on other communication topologies and topology scales.



(a) torus, $n = 16$.

(b) torus, $n = 64$.

(c) torus, $n = 100$.

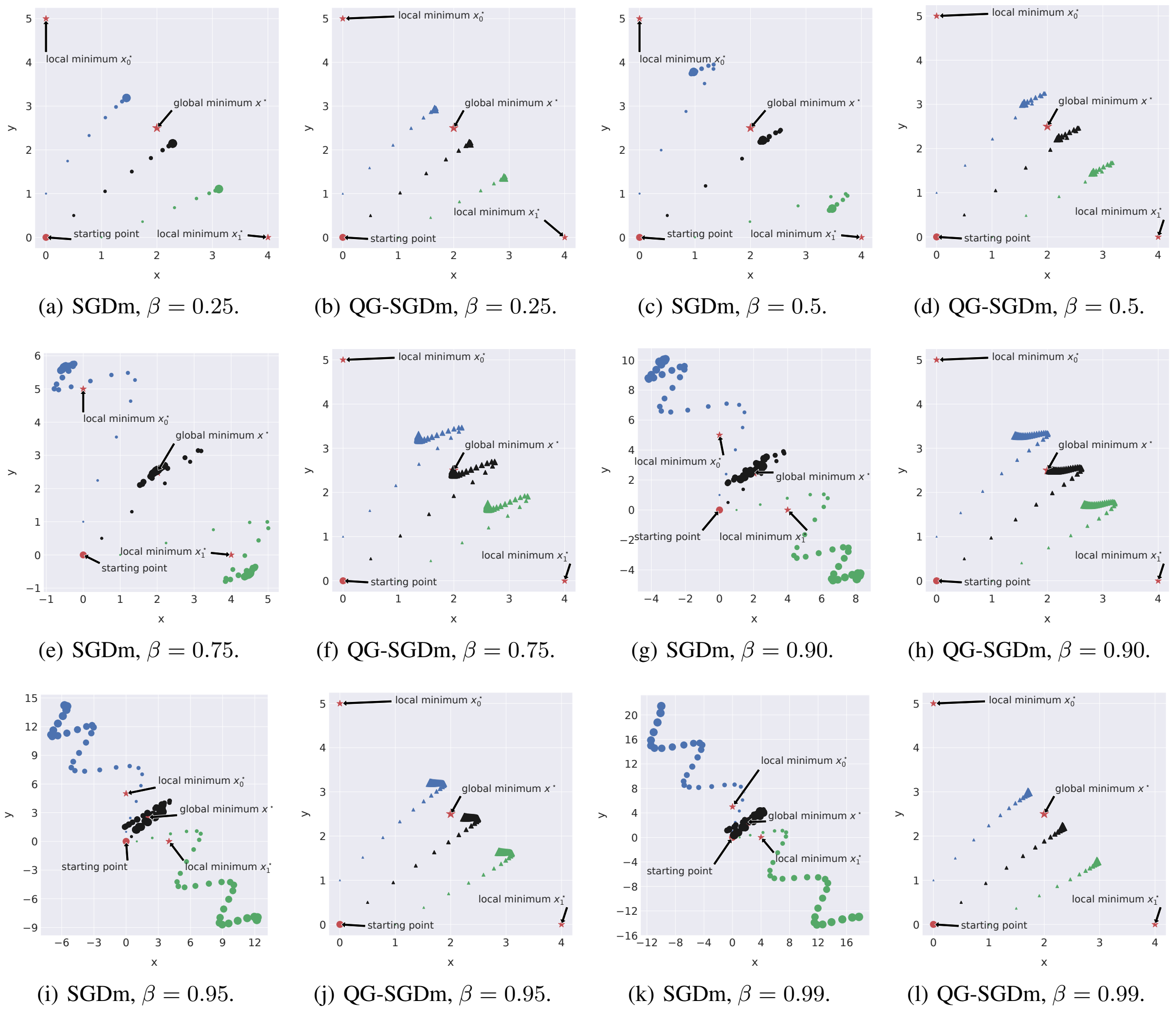
(d) torus, $n = 256$.

(e) social network, $n = 32$.

Figure 10: More results on understanding QG-DSGDm through the aspect of distributed consensus averaging problem on different communication topologies and scales. QG-DSGDm without gradient update step (as in (4)) still presents faster convergence (to a relative high precision) than the normal gossip algorithm.

## D.2. Results on 2D Illustration

Following the 2D illustration in Figure 2, we elaborate below the different choices of momentum factor for SGDm and QG-SGDm in Figure 11. We can witness that the effectiveness of local momentum (oscillation) in SGDm is always impacted by the data heterogeneity, no matter the choices of momentum factor. While for QG-SGDm, there exists a trade-off between stabilized optimization and fast convergence, controlled by the momentum factor $\beta$. Note that we always set $\mu := \beta$ in QG-SGDm, which may result in undesirable behavior.



(a) SGDm, $\beta = 0.25$. (b) QG-SGDm, $\beta = 0.25$. (c) SGDm, $\beta = 0.5$. (d) QG-SGDm, $\beta = 0.5$.

(e) SGDm, $\beta = 0.75$. (f) QG-SGDm, $\beta = 0.75$. (g) SGDm, $\beta = 0.90$. (h) QG-SGDm, $\beta = 0.90$.

(i) SGDm, $\beta = 0.95$. (j) QG-SGDm, $\beta = 0.95$. (k) SGDm, $\beta = 0.99$. (l) QG-SGDm, $\beta = 0.99$.

Figure 11: **The ineffectiveness of local momentum acceleration under heterogeneous data setup**: the local momentum buffer accumulates "biased" gradients, causing unstable and oscillation behaviors. The gradient is estimated by the direction from a given model to the local minimum with a constant update magnitude. The size of marker will increase by the number of update steps; colors blue and green indicate the local models of two workers (after performing local update), while color black is the synchronized global model. Uniform weight averaging is performed after each update step, and the new gradients will be computed on the averaged model.

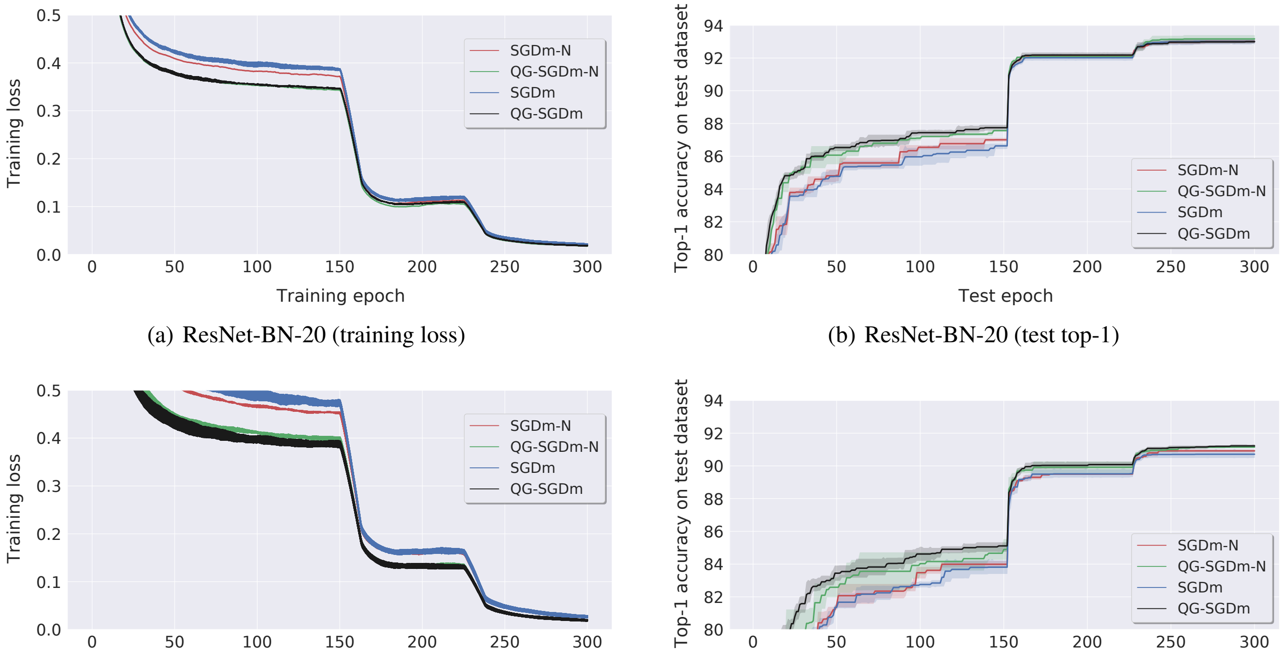
### D.3. Understanding QG-DSGDm and QG-DSGDm-N on the Single Worker Case

Recall that the single worker case of QG-DSGDm and QG-DSGDm-N refers to QG-SGDm and QG-SGDm-N.

Figure 12 studies the learning behavior (learning curves for both training loss and top-1 test accuracy, as well as final best test accuracy) of QG-SGDm and QG-SGDm-N on two different normalization methods (BN and GN) for ResNet-20 on CIFAR-10. In general Nesterov momentum variants outperforms that of HeavyBall momentum, and we can witness a larger performance gain when the optimization is challenging (e.g. in the of using GN replacement).

Figure 13 further investigates the impact of weight decay on Nesterov momentum variants. Excluding weight decay from the training procedure is detrimental to the final generalization performance. We also notice larger benefits of QG-SGDm-N when the optimization procedure is fragile/unstable.

Figure 14 in addition illustrates the curves of weight norm and effective step-size during the optimization procedure, to interpret the potential causes of the performance gain.



(a) ResNet-BN-20 (training loss)

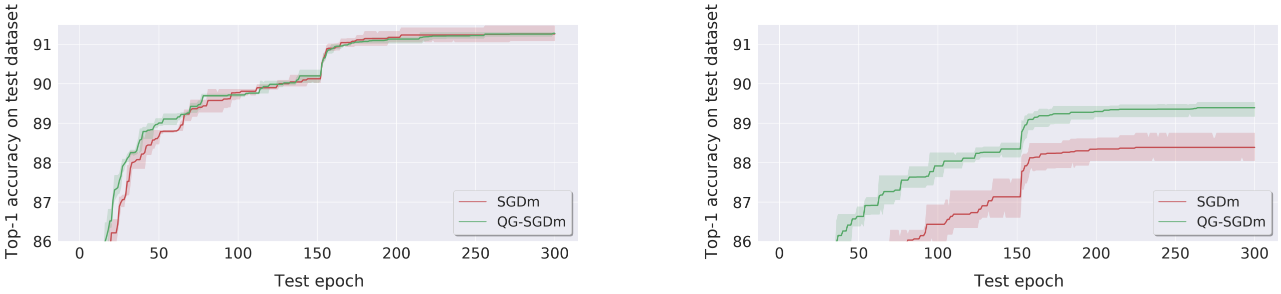
(b) ResNet-BN-20 (test top-1)

(c) ResNet-GN-20 (training loss)

(d) ResNet-GN-20 (test top-1)

| | SGDm-N | QG-SGDm-N | SGDm | QG-SGDm |
|---|---|---|---|---|
| BN | $93.01 \pm 0.12$ | **93.17** $\pm 0.30$ | $92.90 \pm 0.15$ | **92.95** $\pm 0.13$ |
| GN | $90.92 \pm 0.08$ | **91.15** $\pm 0.06$ | $90.61 \pm 0.15$ | **91.23** $\pm 0.01$ |

Figure 12: Understanding the learning behavior of QG-SGDm and QG-SGDm-N, for training ResNet-20 on CIFAR-10 with mini-batch size of 32.

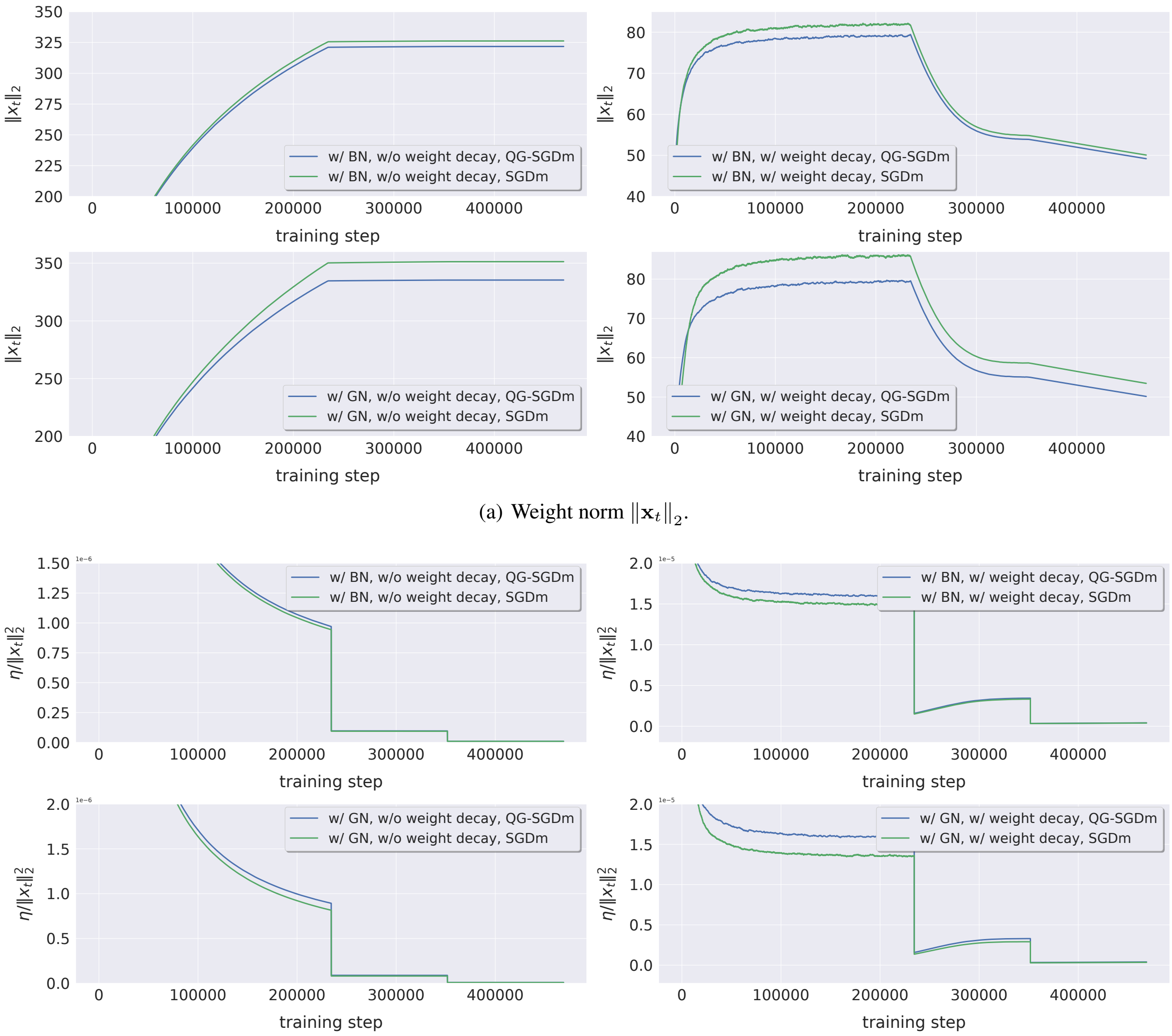


(a) ResNet-BN-20.

(b) ResNet-GN-20.

| | w/ weight decay | | w/o weight decay | |
|---|---|---|---|---|
| | SGDm-N | QG-SGDm-N | SGDm-N | QG-SGDm-N |
| BN | $93.01 \pm 0.12$ | **93.17** $\pm 0.30$ | $91.26 \pm 0.19$ | **91.28** $\pm 0.03$ |
| GN | $90.92 \pm 0.08$ | **91.15** $\pm 0.06$ | $88.39 \pm 0.35$ | **89.39** $\pm 0.20$ |

Figure 13: Understanding the impact of weight decay on QG-SGDm-N, for training ResNet-20 on CIFAR-10 with mini-batch size of 32,



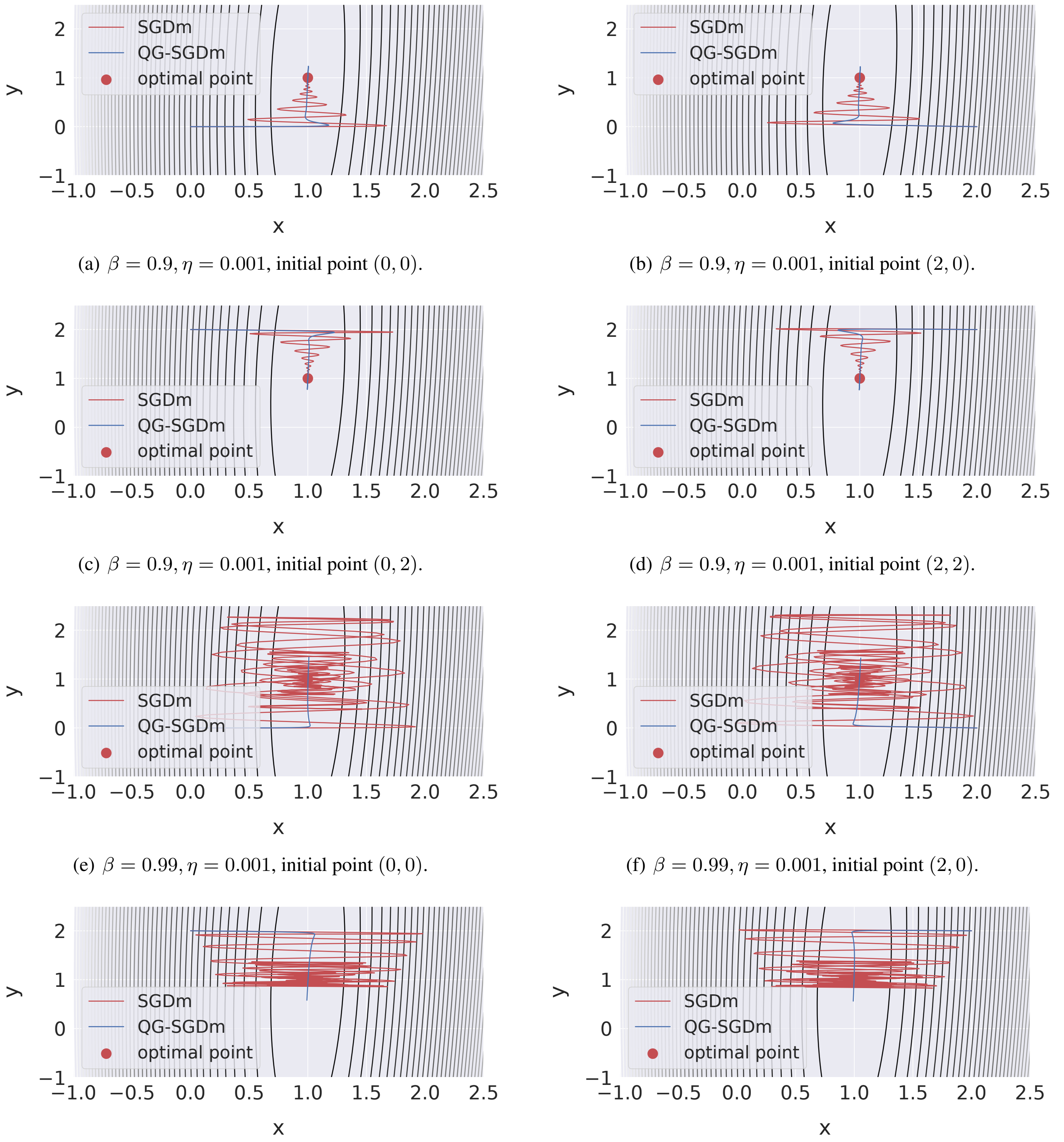
(a) Weight norm $\|\mathbf{x}_t\|_2$.

(b) Effective stepsize $\frac{\eta}{\|\mathbf{x}_t\|_2^2}$.

Figure 14: Understanding QG-SGDm through the lens of the effective step-size, for training ResNet-20 on CIFAR-10 with mini-batch size of 32.

### D.4. Understanding QG-DSGDm on the Single Worker Case via Toy Function

Similar to Lucas et al. (2019), we first optimize the Rosenbrock function, defined as $f(x, y) = (y - x^2)^2 + 100(x - 1)^2$. Figure 4 illustrates the stabilized optimization trajectory in QG-SGDm (much less oscillation than SGDm).



(a) $\beta = 0.9, \eta = 0.001$, initial point $(0, 0)$.

(b) $\beta = 0.9, \eta = 0.001$, initial point $(2, 0)$.

(c) $\beta = 0.9, \eta = 0.001$, initial point $(0, 2)$.

(d) $\beta = 0.9, \eta = 0.001$, initial point $(2, 2)$.

(e) $\beta = 0.99, \eta = 0.001$, initial point $(0, 0)$.

(f) $\beta = 0.99, \eta = 0.001$, initial point $(2, 0)$.

(g) $\beta = 0.99, \eta = 0.001$, initial point $(0, 2)$.

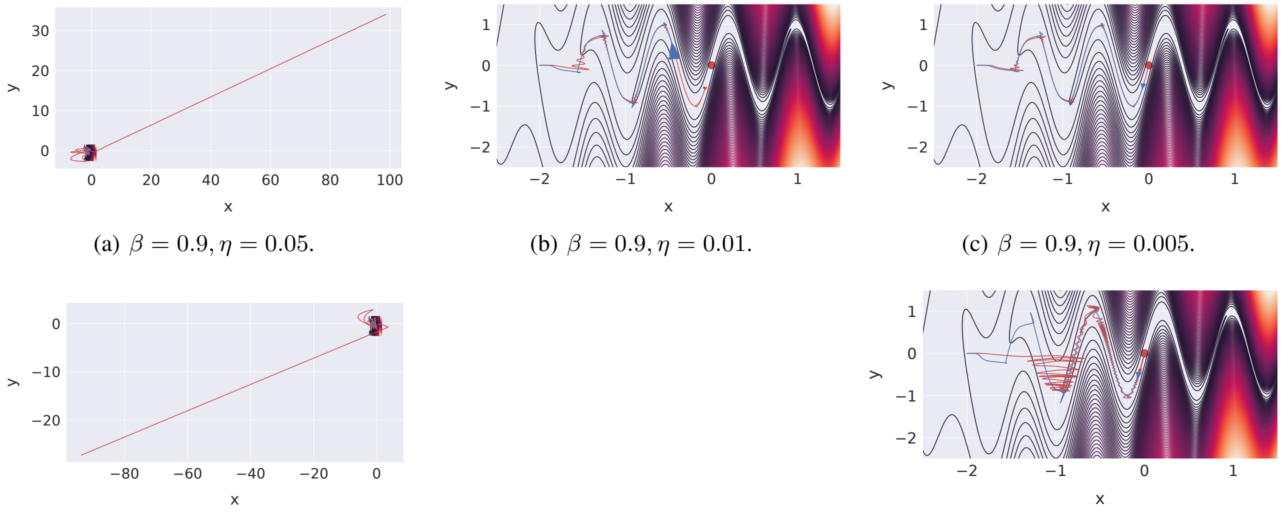
(h) $\beta = 0.99, \eta = 0.001$, initial point $(2, 2)$.

Figure 15: Understanding the optimization trajectory of QG-SGDm and Heavy-ball momentum SGD (SGDm), via a 2D toy function $f(x, y) = (y - x^2)^2 + 100(x - 1)^2$. This function has a global minimum at $(x, y) = (1, 1)$. Red line corresponds to SGDm and blue line indicates QG-SGDm. Red line illustrates larger oscillation than QG-SGDm on the optimization trajectory.

We further study a simple non-convex toy problem Lucas et al. (2019):

$$f(x, y) = \log(e^x + e^{-x}) + 10 \log \left( e^{e^x (y - sin(ax))} + e^{-e^x (y - sin(ax))} \right) ,$$

in Figure 16. In our experiments, we choose $a = 8$ and $b = 10$, and initialize the optimizer at $(x, y) = (-2, 0)$.



(a) $\beta = 0.9, \eta = 0.05$.

(b) $\beta = 0.9, \eta = 0.01$.

(c) $\beta = 0.9, \eta = 0.005$.

(d) $\beta = 0.99, \eta = 0.01$.

(e) $\beta = 0.99, \eta = 0.005$.

Figure 16: Understanding the optimization trajectory of QG-SGDm and Heavy-ball momentum SGD (SGDm), via a 2D toy function $f(x, y) = \log(e^x + e^{-x}) + 10 \log \left( e^{e^x (y - sin(8x))} + e^{-e^x (y - sin(8x))} \right)$. This function has an optimal value at $(x, y) = (0, 0)$. Red line corresponds to SGDm and blue line indicates QG-SGDm. Red line illustrates larger oscillation than QG-SGDm on the optimization trajectory.

### D.5. The Learning Curves on CV tasks

Figure 17 visualizes the learning curves for training ResNet-EvoNorm-20 on CIFAR-10, in terms of different degrees of non-i.i.d.-ness and network topologies (Ring and Social topology).

### D.6. The Ineffectiveness of Tuning Momentum Factor for DSGDm-N

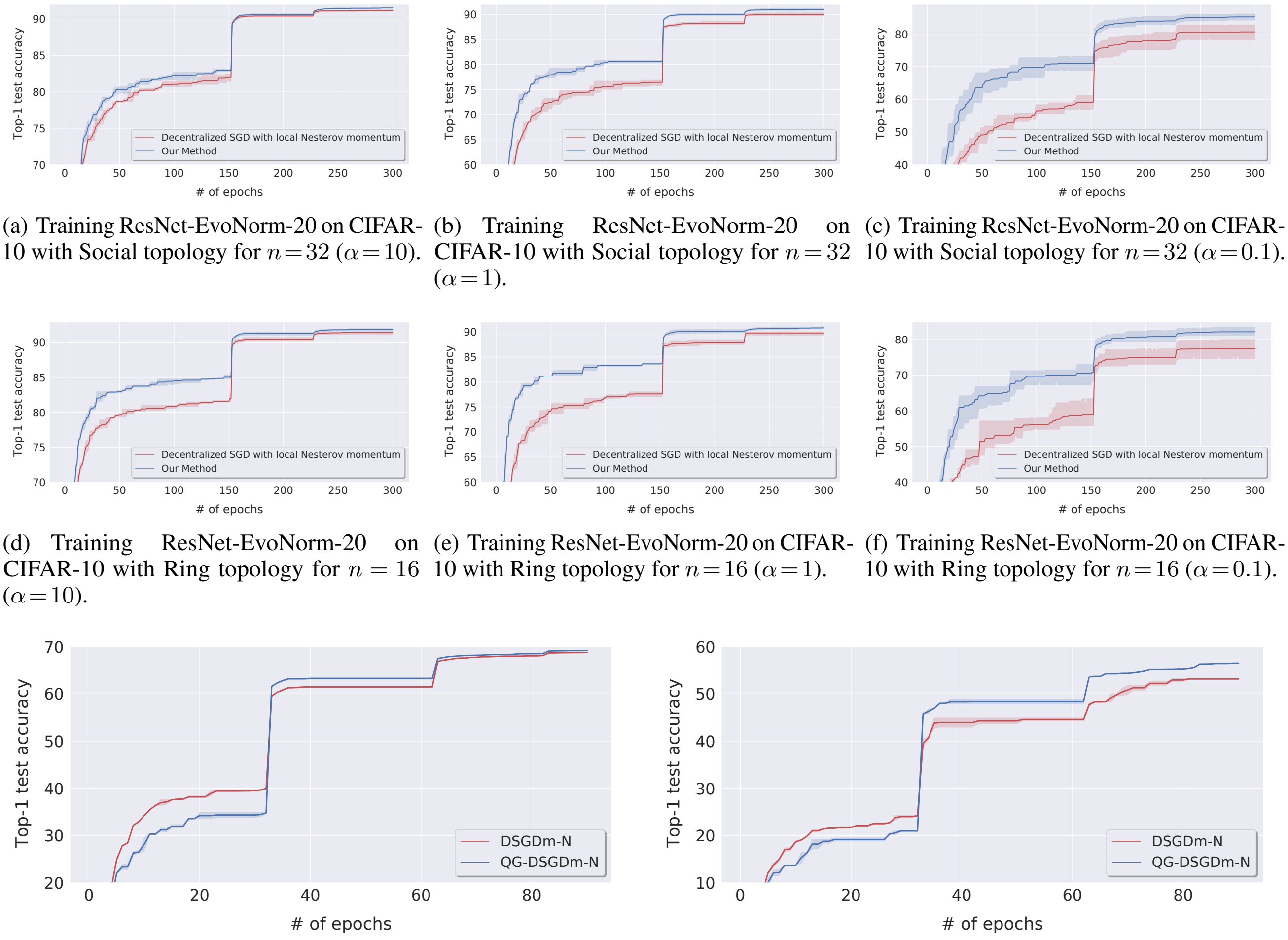
Table 18 shows that tuning momentum factor for DSGDm-N cannot alleviate the training difficulty caused by heterogeneity.

### D.7. The Superior Performance of QG-DSGDm-N Generalize Different Topology Scales

Table 7 further showcases the generality of the predominant performance gain of quasi-global momentum on different topology scales ($n$).

Table 7: **The test top-1 accuracy of different decentralized algorithms evaluated on different topology scales and non-i.i.d.-ness**, for training ResNet-EvoNorm-20 on CIFAR-10. The results are over three random seeds, with sufficient learning rate tuning. The table corresponds to Figure 6 in the main paper.

| Methods | Ring ($n$=16) | | Ring ($n$=32) | | Ring ($n$=48) | |
|---|---|---|---|---|---|---|
| | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 1$ | $\alpha = 0.1$ |
| SGDm-N (centralized) | $92.18 \pm 0.19$ | | $91.92 \pm 0.33$ | | $91.63 \pm 0.25$ | |
| DSGDm-N | $89.98 \pm 0.10$ | $77.48 \pm 2.67$ | $88.46 \pm 0.29$ | $78.17 \pm 1.63$ | $85.54 \pm 0.33$ | $73.67 \pm 0.90$ |
| QG-DSGDm-N | **91.28** $\pm 0.38$ | **82.20** $\pm 1.27$ | **90.27** $\pm 0.07$ | **83.18** $\pm 1.11$ | **89.75** $\pm 0.32$ | **80.28** $\pm 1.52$ |



(a) Training ResNet-EvoNorm-20 on CIFAR-10 with Social topology for $n = 32$ ($\alpha = 10$).

(b) Training ResNet-EvoNorm-20 on CIFAR-10 with Social topology for $n = 32$ ($\alpha = 1$).

(c) Training ResNet-EvoNorm-20 on CIFAR-10 with Social topology for $n = 32$ ($\alpha = 0.1$).

(d) Training ResNet-EvoNorm-20 on CIFAR-10 with Ring topology for $n = 16$ ($\alpha = 10$).

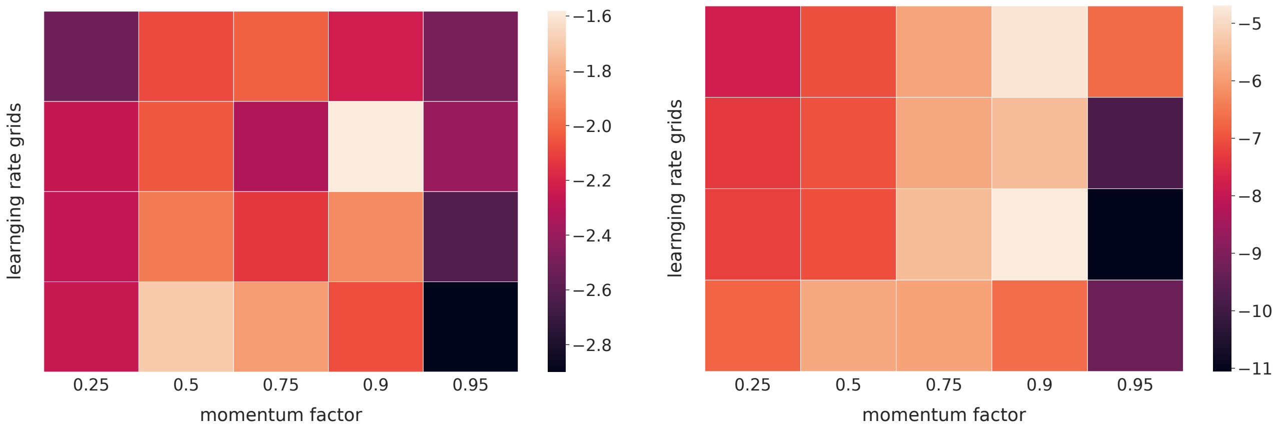
(e) Training ResNet-EvoNorm-20 on CIFAR-10 with Ring topology for $n = 16$ ($\alpha = 1$).

(f) Training ResNet-EvoNorm-20 on CIFAR-10 with Ring topology for $n = 16$ ($\alpha = 0.1$).

(g) Training ResNet-EvoNorm-18 on ImageNet with Ring topology for $n = 16$ ($\alpha = 1$).

(h) Training ResNet-EvoNorm-18 on ImageNet with Ring topology for $n = 16$ ($\alpha = 0.1$).

Figure 17: Learning curves for cv tasks.



(a)

(b)

Figure 18: The ineffectiveness of tuning momentum factors for DSGDm-N, for training ResNet-EvoNorm-20 on CIFAR-10. We illustrate the performance gap between DSGDm-N (different combination of learning rate and momentum factor) and QG-DSGDm-N (tuned learning rate from the grid with default momentum factor 0.9).

### D.8. Multiple-step QG-DSGDm-N variant

Table 8 illustrates the performance for the multiple-step variant of QG-DSGDm-N. We can witness that tuning the value of $\tau$ cannot lead to a significant performance gain.

Table 8: **Ablation study for the variant of multiple-step QG-DSGDm-N** (illustrated in Algorithm 3), for training ResNet-EvoNorm-20 on CIFAR-10. The results are averaged over three seeds with tuned learning rate.

| Methods | Ring ($n = 16$) | |
|---|---|---|
| | $\alpha = 1$ | $\alpha = 0.1$ |
| SGDm-N (centralized) | $92.18 \pm 0.19$ | |
| DSGD | $88.88 \pm 0.26$ | $74.55 \pm 2.07$ |
| DSGDm-N | $89.98 \pm 0.10$ | $77.48 \pm 2.67$ |
| QG-DSGDm-N ($\tau = 1$) | $91.28 \pm 0.38$ | $82.20 \pm 1.27$ |
| QG-DSGDm-N ($\tau = 2$) | $91.11 \pm 0.18$ | $82.25 \pm 1.68$ |
| QG-DSGDm-N ($\tau = 3$) | $91.04 \pm 0.01$ | $81.57 \pm 2.21$ |
| QG-DSGDm-N ($\tau = 4$) | $91.26 \pm 0.25$ | $82.55 \pm 1.55$ |

### D.9. Comparison with D$^2$ and Gradient Tracking (GT) methods

We comment on GT methods below (including D$^2$ (Tang et al., 2018b)), in order to 1) highlight the distinctions between different algorithms, and 2) justify the comparison with existing GT methods.

- Distinctions between algorithms: 1) Both D$^2$ and GT do not consider momentum in their algorithm design and theoretical analysis, while one of our main contributions is the design of quasi-global momentum—a simple yet effective approach for the SOTA decentralized deep learning training; 2) It is unclear how to integrate D$^2$ with momentum, given the original design intuition of D$^2$; 3) QG-DSGDm is different from D$^2$, where the updates of QG-DSGDm and D$^2$ follow $\mathbf{W}((1+\beta\frac{\eta^{(t)}}{\eta^{(t-1)}})\mathbf{X}^{(t)}-\beta\frac{\eta^{(t)}}{\eta^{(t-1)}}\mathbf{X}^{(t-1)}-\eta^{(t)}\nabla f(\mathbf{X}^{(t)}))$ and $\mathbf{W}(2\mathbf{X}^{(t)}-\mathbf{X}^{(t-1)}-\eta(\nabla f(\mathbf{X}^{(t)})-\nabla f(\mathbf{X}^{(t-1)})))$ respectively (we simplify the comparison by letting $\mu=0$ in QG-DSGDm); 4) Compared to QG-DSGDm, GT requires extra one communication step per update to approximate the global average of local gradients.
- D$^2$ cannot achieve comparable test performance on the standard deep learning benchmark.
  - D$^2$ requires a constant learning rate, which does not fit the SOTA learning rate schedule (e.g. stage-wise) in deep learning. Note that D$^2$ can be rewritten as $\mathbf{W}(\mathbf{X}^{(t)}-\eta((\mathbf{X}^{(t-1)}-\mathbf{X}^{(t)})/\eta+\nabla f(\mathbf{X}^{(t)})-\nabla f(\mathbf{X}^{(t-1)})))$, and the update would break if the magnitude of $\mathbf{X}^{(t-1)}-\mathbf{X}^{(t)}$ is a factor of $10\eta$ (i.e. performing learning rate decay at step $t$).
  - It is non-trivial to improve D$^2$ for the SOTA deep learning training. To support our argument, Table 2 compares to an improved D$^2$ variant (noted as D$^2_+$) to address the issue of learning rate decay in D$^2$ (though it breaks the design intuition of D$^2$); the performance of D$^2_+$ is far behind our scheme. The update of D$^2_+$ follows $\mathbf{W}(\mathbf{X}^{(t)}-\eta^{(t)}((\mathbf{X}^{(t-1)}-\mathbf{X}^{(t)})/\eta^{(t-1)}+\nabla f(\mathbf{X}^{(t)})-\nabla f(\mathbf{X}^{(t-1)})))$.
  - The numerical results of D$^2$ in Tang et al. (2018b); Pan et al. (2020); Lu et al. (2019) cannot support the practicability of D$^2$: 1) The experiments of Tang et al. (2018b); Pan et al. (2020) only consider a very small scale setup (# of nodes $n=5$ or 8) for CIFAR-10, while (Lu et al., 2019) only evaluates on the toy MNIST dataset—these setups are much less challenging than ours; 2) Only training loss curves are reported in Tang et al. (2018b); Pan et al. (2020); Lu et al. (2019), and the final training loss values of Tang et al. (2018b); Pan et al. (2020) are much higher than 0 (i.e. not converge to a local minimum).
- As suggested by anonymous reviewers, we compare with GT methods in Table 2: QG-DSGDm-N outperforms GT by a large margin. We would like to point out that 1) the observations of the marginal performance gain in GT are aligned with prior works, e.g. similar numerical results in Figure 3 & 4 & 5 of Xin et al. (2020); 2) GT may have more benefits in the extreme low training loss regime (e.g. less than $1e-4$) where gains might increase when combining with variance reduction techniques (Xin et al., 2020)—however, we focus on the test performance for deep learning (Defazio & Bottou, 2019); 3) recent work (Yuan et al., 2020b) also proves that gradient tracking methods are in general much more sensitive than diffusion-based methods.