Decentralized Deep Learning using Momentum-Accelerated Consensus

Aditya Balu 1 Zhanhong Jiang 2 Sin Yong Tan 1 Chinmay Hedge 3 Young M Lee 2 Soumik Sarkar 1

Abstract
We consider the problem of decentralized deep learning where multiple agents collaborate to learn from a distributed dataset. While there exist several decentralized deep learning approaches, the majority consider a central parameter-server topology for aggregating the model parameters from the agents. However, such a topology may be inapplicable in networked systems such as ad-hoc mobile networks, field robotics, and power network systems where direct communication with the central parameter server may be inefficient. In this context, we propose and analyze a novel decentralized deep learning algorithm where the agents interact over a fixed communication topology (without a central server). Our algorithm is based on the heavy-ball acceleration method used in gradient-based optimization. We propose a novel consensus protocol where each agent shares with its neighbors its model parameters as well as gradient-momentum values during the optimization process. We consider both strongly convex and non-convex objective functions and theoretically analyze our algorithm’s performance. We present several empirical comparisons with competing decentralized learning methods to demonstrate the efficacy of our approach under different communication topologies.

1. Introduction
Spurred by the need to accelerate deep neural network training with massive distributed datasets, several recent research efforts (Dean et al., 2012; Zhang et al., 2015; Jin et al., 2016; Kairouz et al., 2019) have put forth a variety of distributed, parallel learning approaches. One line of work has focused on adapting traditional deep learning algorithms that use a single CPU-GPU environment to a distributed setting with a network of several GPUs (Wen et al., 2017; Zhang et al., 2015; Goyal et al., 2017; Chen et al., 2016). Some of these approaches also can be used in conjunction with gradient compression schemes between compute nodes in the network (Bernstein et al., 2018). A different line of works falls under the umbrella of federated learning (Konečný et al., 2016) which deals with inherently decentralized datasets, i.e., each compute node has its own corresponding set of data samples that is not shared. The majority of works in this area consider a central parameter-server topology that aggregates estimates of model parameters from the agents.

In this paper, our particular focus is on decentralized learning where there is no central server: each node in the network maintains its own model parameters (which it can communicate with its neighbors defined according to a pre-specified, but otherwise arbitrary, communication topology), and the goal is to arrive at a consensus model for the whole network. See Lian et al. (2017); Jiang et al. (2017); Assran et al. (2018); Kamp et al. (2018) for examples of such decentralized learning approaches.

While the above works are representative of key advances in the algorithmic front, several gaps remain in our understanding of centralized versus distributed learning approaches. Conspicuous among these gaps is the notion of momentum, which is a common technique to speed up convergence in gradient-based learning with minimal overhead in computational costs (Nesterov, 2013b; Sutskever et al., 2013). However, few papers (barring exceptions such as Ghadimi et al. (2013); Jiang et al. (2017); Assran et al. (2018); Yu et al. (2019)) in the decentralized learning literature have touched upon momentum-based acceleration techniques, and to our knowledge, rigorous guarantees in the context of non-convex and stochastic optimization have not been presented. Our objective in this paper is to fill this key gap from both a theoretical as well as empirical perspective.

The standard analysis of decentralized learning algorithms establishes convergence rates of gradient-based learning assuming that the loss function is strongly convex. However, strong convexity is too strict an assumption in real-world applications, and particularly in the deep learning setting. In the centralized case, one way to overcome this is to assume objective functions that obey the Polyak-Łojasiewicz, or the PL, criterion (also called as quasi-convex or invex functions). It has been shown that SGD and its variants algorithms achieve global linear convergence rates (Karimi

---

1 Equal contribution 1 Iowa State University 2 Johnson Controls 3 New York University. Correspondence to: Soumik Sarkar <soumiks@iastate.edu>.
Table 1. Comparisons between different optimization approaches. Gra.Lip.: Gradient Lipschitz. Str.Con.: strongly convex. Cen.: centralized. Non-str.: non-strongly convex. Dec: decentralized. ρ: a positive constant in (0, 1). k is the number of iterations. N: the number of agents. PL: Polyak-Łojasiewicz condition. It should be noted that each ρ in different methods vary in real values.

| Method                          | Setting | Rate          | Gra.Lip. | S. Conv. |
|---------------------------------|---------|---------------|----------|----------|
| HBM (2017)                      | Cen.    | O(ρk)         | Yes      | No       |
| MSWG (Ghadimi et al., 2013)     | Dec.    | O(ρk)         | Yes      | No       |
| SHB (Loizou & Richtárik, 2017)  | Dec.    | O(ρk)         | Yes      | Yes      |
| DMSGD (This paper)              | Dec.    | O(ρk)         | Yes      | Yes      |
| SUM (Yang et al., 2016)         | Cen.    | O(1/√k)       | Yes      | Yes      |
| CSDSGD/D-PSGD                   | Dec.    | O(1/k + 1/√Nk) | Yes | No |
| MSGD (Yu et al., 2019)          | Dec.    | O(1/k + 1/√Nk) | Yes | Yes |
| DMSGD (This paper)              | Dec.    | O(1/k)        | Yes      | Yes      |

Our contributions. We propose and analyze a stochastic optimization algorithm that we call decentralized momentum SGD (DMSGD), based on the classical notion of momentum (or the heavy-ball method) used in gradient-based optimization (Polyak, 1964). In contrast with previous analyses (Jakovetic et al., 2014; Ghadimi et al., 2013), our work takes into account the following factors: DMSGD succeeds in the stochastic setting with decentralized data/parameters for optimizing objective functions that exhibit quasi-convexity (in the PL sense) as well as non-convexity. See Table 1 for quantitative comparisons with existing approaches.

For smooth and strongly-convex objective functions, when the step-size is appropriately chosen to be constant, we first establish that the convergence rate of the proposed algorithm is linear. We then relax the requirement of strong-convexity using the Polyak-Łojasiewicz criterion and obtain the convergence rate to be linear in the quasi-convex. For smooth and nonconvex objective functions, we then show the convergence to a first-order stationary point, that is, the algorithm produces an estimate x with sufficiently small gradient (‖∇f(x)‖ ≤ ε) after O(1/ε) iterations.

We empirically compare DMSGD with baseline decentralized methods such as D-PSGD/CDSGD (Lian et al., 2017; Jiang et al., 2017). We show that when the momentum term is appropriately weighted, DMSGD is faster and more accurate than these baseline methods, suggesting the benefits of its use on practice.

Other related work. Federated learning was initially proposed by (McMahan et al., 2016; Konečný et al., 2016), with the motivation for learning in a setting where there are several clients (e.g. mobile devices etc.) and a server to communicate with all the clients. More recently, several researchers have shown interests in federated learning and the challenges involved in it. A more elaborate and detailed review is given in (Kairouz et al., 2019).

Early works in decentralized deep learning without a central server include Lian et al. (2017); Jiang et al. (2017); Assran et al. (2018); Tang et al. (2018); Wang & Joshi (2018b); Kamp et al. (2018); Xin et al. (2019); Nadiradze et al. (2019). Approaches such as Consensus based Distributed SGD (CDSGD) (Jiang et al., 2017) and Decentralized Parallel SGD (D-PSGD) (Lian et al., 2017) consider optimization over fixed communication topologies. (We note that D-PSGD and CDSGD are essentially similar and differ only in implementation). However, these works do not provide a rigorous analysis of momentum-based methods. Assran et al. (2018) implement the PushSum gossip algorithm with stochastic updates. Wang & Joshi (2018b;a) present a framework for communication-efficient SGD algorithms where they study error-runtime tradeoffs. Nadiradze et al. (2019) provide a population-based model for decentralized learning. Kamp et al. (2018) propose a protocol to handle different phases of training and adaptively perform model-averaging based on a trade-off between communication and model performance. Jiang et al. (2018); Li et al. (2019) also obtain similar communication-convergence tradeoffs.

The optimization literature puts forth several notions of momentum to accelerate gradient descent (Nesterov, 1983; Su et al., 2016; Ghadimi & Lan, 2016; Nesterov et al., 2016; Botev et al., 2016; Nesterov, 2013a; Lin et al., 2014; Jakovetic et al., 2014). Although (classical) momentum has been shown to improve convergence rates, proving this is a challenging task even in the convex case (Ghadimi et al., 2015; Zhang et al., 2015).

In (Ghadimi et al., 2015), the authors provided a global convergence analysis for the classic momentum on the convex optimization problems with Lipschitz continuous gradients with and without strong convexity. While Wang & Miller (2014) proposed the scaled classic momentum method for Poisson data optimization and showed the proof of conver-
gence rate of $O(1/k^2)$ under Lipschitz assumptions, Ochs et al. (2015) studied the convergence rate using the classic momentum when the objective function was twice continuously differentiable. Loizou & Richtárik (2017) showed that for the squared-error loss, a (centralized) stochastic heavy ball approach can achieve linear convergence rate. Ghadimi et al. (2013) used the classical momentum method for networked optimization and found a linear convergence rate when the objective function was strongly convex and twice continuously differentiable. However, their method is not applicable to stochastic non-convex optimization.

2. Problem Setup and Preliminaries

Let the parameters of the deep neural network be denoted as $x \in \mathbb{R}^d$. We define a loss function $f : \mathbb{R}^d \to \mathbb{R}$ and denote its corresponding gradient by $g$.

Decentralized learning Consider a static undirected graph $G = (V,E)$, where $V$ is the node set and $E$ is an edge set. Consequently, if we assume that there exist $N$ nodes (agents) in the networked system, we can denote $V = \{1,2,\ldots,N\}$ while $E \subseteq V \times V$. If $(j,l) \in E$, then agent $j$ can communicate with agent $l$. A node $j \in V$ has its neighbors $Nb(j) \triangleq \{j \in V : (j,l) \in E \text{ or } l = j\}$. We assume that the network $G$ is connected without loss of generality throughout this paper. In this paper, we consider a finite sum minimization problem defined as follows:

$$\min \frac{1}{n} \sum_{j=1}^{N} \sum_{i \in D_j} f_j^i(x),$$

(1)

where $D_j$ denotes the subset of the training data (comprising $n_j$ samples) only known by the $j^{th}$ agent such that $\sum_{j=1}^{N} n_j = n$, $n$ is the size of dataset, $N$ is the number of agents, $f_j : \mathbb{R}^d \to \mathbb{R}$ are local loss functions of each node. Let $x^j \in \mathbb{R}^d$ be a local copy of $x$. Then, define $x = [x^1; x^2; \ldots; x^N] \in \mathbb{R}^{Nd \times 1}$. All vector and matrix norms are Euclidean and Frobenius norms respectively.

In this paper, for simplicity of presentation, we assume that $d = 1$, while noting that exactly the same proofs hold when $d > 1$ albeit the expense of extra notation.

Equation 1 can be rewritten as the constrained problem:

$$\min F(x) \triangleq \frac{1}{n} \sum_{j=1}^{N} \sum_{i \in D_j} f_j^i(x^j), \quad \text{s.t.} \quad \Pi x = x,$$

(2)

where the matrix $\Pi$ is the mixing matrix encoding the adjacency structure of $G$ (which is assumed to be doubly stochastic). By turning the hard constraint $\Pi x = x$ into a soft constraint that penalizes the corresponding decision variables $x$, the following equivalent objective function can be obtained:

$$\mathcal{F}(x) := F(x) + \frac{1}{2\xi} (x^T (I - \Pi)x)$$

(3)

where $\xi > 0$. In the next section, we will show that $\xi$ can be related to the step size $\alpha$.

In order to study the behavior of the proposed algorithm, we now present basic definitions and assumptions.

**Definition 1.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is $\mu$-strongly convex, if for all $x, y \in \mathbb{R}^d$, we have

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \|y - x\|^2.$$  

**Definition 2.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is $L$-smooth, if for all $x, y \in \mathbb{R}^d$, we have

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} \|y - x\|^2.$$  

**Definition 3.** A function $c(\cdot)$ is said to be coercive if it satisfies $c(x) \to \infty$ when $\|x\| \to \infty$.

**Assumption 1.** The objective functions $f_j : \mathbb{R}^d \to \mathbb{R}$ are assumed to satisfy the following conditions: a) Each $f_j$ is $L_j$-smooth; b) each $f_j$ is proper (not everywhere infinite) and coercive. An immediate consequence of Assumption 1 a) is that $\sum_{j=1}^{N} f_j(x^j)$ is $L_m$-smooth where $L_m := \max\{L_1, L_2, \ldots, L_N\}$. Similarly, when the local objective functions $f_j$ are $\mu_j$-strongly convex, then $\sum_{j=1}^{N} f_j(x^j)$ is $\mu_m$-strongly convex where $\mu_m := \min\{\mu_1, \mu_2, \ldots, \mu_N\}$.

**Assumption 2.** The unified objective function $\mathcal{F}(x)$ is $\mathcal{M}$-Lipschitz continuous.

Denoting $\mathcal{S}(x)$ by the stochastic gradient of $\mathcal{F}$ at point $x$, we next make another assumption on the variance of $\mathcal{S}(x)$ to ensure that it is bounded from above.

**Assumption 3.** The (stochastic) gradients of $\mathcal{F}$ satisfy:

(a) $\text{Var}(\mathcal{S}(x)) = E[||\mathcal{S}(x) - \nabla \mathcal{F}(x)|||^2] \leq \sigma^2$

(b) $\nabla \mathcal{F}(x) = E[\mathcal{S}(x)]$.

Assumption 3(b) implies that $\mathcal{S}(x)$ is an unbiased estimate of $\nabla \mathcal{F}(x)$. The above assumptions also imply the boundedness of the stochastic gradient of $\mathcal{F}(x)$. As $\mathcal{F}(x)$ is smooth, $||\nabla \mathcal{F}(x)||$ is also bounded above by $M$. With Assumption 2 and Assumption 3(b), we have

$$E[||\mathcal{S}(x)||] = \sqrt{E[||\mathcal{S}(x)||]^2} \leq \sqrt{E[||\nabla \mathcal{F}(x)||]^2}$$

$$= \sqrt{E[||\nabla \mathcal{F}(x)||]^2 + \text{Var}(\mathcal{S}(x))} \leq \sqrt{M^2 + \sigma^2}.$$  

3. Proposed Algorithm

We first present our proposed approach in Algorithm 1.

In the above update law, $g^j(x_k^j)$ is a stochastic gradient which is calculated by randomly selecting at uniform a mini-batch for each agent. Let $D'_{j}$ be a mini-batch of the dataset $D_j$ of the $j$-th agent. Therefore,

$$g^j(x_k^j) = \frac{1}{b} \sum_{i \in D'_{j}} \nabla f_j^i(x_k^j),$$

where $b$ is the size of $D'$. 
We now rewrite the update law with option I in a vector form:

\[ x_{k+1} = x_k - \alpha g(x_k) + \frac{1}{\alpha}(I - \Pi)x_k \]

\[ + \beta(\omega I + (1 - \omega)\Pi)(x_k - x_{k-1}) \]

Here, we define \( S(x_k) = g(x_k) + \frac{1}{\alpha}(I - \Pi)x_k \) and \( \tilde{\Pi} = \omega I + (1 - \omega)\Pi \). Consequently, Eq. 4 can be written in a compact form as:

\[ x_{k+1} = x_k - \alpha S(x_k) + \beta\tilde{\Pi}(x_k - x_{k-1}) \tag{5} \]

For option II, we can derive the analogous vector form such that

\[ x_{k+1} = x_k - \alpha g(x_k) + \beta\tilde{\Pi}(x_k - x_{k-1}) \tag{6} \]

The simplification in Eq. 5 enables us to construct a function that unifies the true objective function with a term that captures the constraint of consensus among agents (nodes of the communication graph).

\[ F(x) := F(x) + \frac{1}{2\alpha}x^T(I - \Pi)x \tag{7} \]

Comparing Eqs. 3 and 7, we can know that they have exactly the same form and in our specific case corresponding to DMSGD, the parameter \( \xi \) is the step size \( \alpha \). When \( F \) is \( \mu \)-strongly convex, we immediately obtain that \( F(x) \) is also strongly convex with parameter \( \mu' = \mu_m + \frac{1}{\alpha} - \lambda N \), where \( \lambda N \) is the \( N \)-th largest eigenvalue of \( \Pi \). Similarly, we can obtain that \( F \) is smooth with \( \lambda' = L_m + \frac{1}{\alpha} - \lambda_2 \) where \( \lambda_2 \) is the second-largest eigenvalue of \( \Pi \).

Similarly, corresponding to Eq. 6, one can obtain that the unified objective function is \( F(x) \) itself. Throughout the rest of the analysis in the main paper, we only focus on the \( F(x) \) (namely, DMSGD Option I) since all the convergence analysis techniques shown in the next section can directly apply to the case where the unified objective is \( F(x) \). For completeness, we present the specific analysis for \( F(x) \) in the supplementary materials.

### 4. Convergence Analysis

**Concensus.** We first prove that the agents achieve consensus, i.e., each agent eventually obtains a parameter that is close to the ensemble average \( \bar{x}_k = \frac{1}{N} \sum_{i=1}^{N} S_i \). Using the metrics of \( \|x_k^i - \bar{x}_k\| \). In the setting of \( d = 1 \), though \( x_k^i \) and \( x_k \) are both scalars, we use the norm notation here for generality. As defined above, \( x \) has dimension of \( N \). Define \( x_k := [\bar{x}_k; \bar{x}_k; \ldots; \bar{x}_k]_N \). Therefore, it holds that \( \|x_k^i - \bar{x}_k\| \leq \|x_k - \bar{x}_k\| \) (Berahas et al., 2018) and instead of directly bounding \( \|x_k^i - \bar{x}_k\| \), we investigate the upper bound for \( \|x_k - \bar{x}_k\| \). We first obtain:

**Proposition 1. (Concensus)** Let all assumptions hold. The iterates generated by DMSGD satisfies the following inequality \( \forall k \in \mathbb{N}, \exists \alpha > 0 \):

\[ \|x_k^i - \bar{x}_k\| \leq \frac{8\alpha\sqrt{N}\sqrt{M^2 + \sigma^2}}{\sqrt{\eta(1 - \beta\lambda)(1 - \sqrt{\beta\lambda})}}, \tag{8} \]

where \( \eta \) is defined as an arbitrarily small constant such that \( \Pi \geq \eta I, 0 < \eta < 1, \lambda = (\mathbf{1} - \omega)\lambda \).

**Proof.** The proof for this proposition is fairly technical and we provide the sketch here, referring interested readers to

---

**Algorithm 1 DMSGD**

**Input:** \( m, \Pi, x_0^i, x_1^i, \alpha, N, \omega, \beta \)

**Output:** \( x^\ast \)

for \( k = 1 : m \) do

for \( j = 1 : N \) do

- **Consensus step:** Nodes run average consensus:
  \[ v_j^k = \sum_{i \in N(j)} p_j x_i^k; \]
- **Momentum step:**
  \[ \delta_k = \omega(x_k - x_{k-1}) + (1 - \omega)(v_k - v_{k-1}); \]
- **Local gradient step** for node \( j \):
  - Option I: \( x_{k+1}^j = x_k^j - \alpha g(x_k^j) + \beta\delta_k; \)
  - Option II: \( x_{k+1}^j = x_k^j - \alpha g(x_k^j) + \beta\delta_k; \)

end

end

In Jiang et al. (2017); Assran et al. (2018), decentralized variants of classic momentum have been proposed (without analysis). On the other hand, our proposed DMSGD method with option I uses a special parameter, \( \omega \), to trade off between different momentum terms. The first momentum term is implemented over the true decision variables \( x_k^j \) while the second momentum term is implemented over the consensus variables \( v_k^j \), which are graph-smoothed averages of the decision variables.

We present a fairly general analysis for DMSGD; as special cases, we obtain known convergence properties for other methods. For example, we recover the decentralized classic momentum SGD by setting the parameter \( \omega = 1 \). When \( \omega = 0 \), DMSGD produces a new decentralized MSGD algorithm in which the momentum relies on the consensus variables. When the parameter \( \beta \) is set to 0, the proposed DMSGD boils down to the decentralized SGD method without momentum (Jiang et al., 2017; Lian et al., 2017). Option II of DMSGD is to replace \( v_k^j \) with \( x_k^j \) in the local gradient step such that the consensus only affects the momentum term. The intuition behind this variant is that for the local gradient step, agent \( j \) only relies on its current state information instead of the consensus, "refusing" to proceed the update on top of an "agreement". For convenience and simplicity, the initial values of \( x^j \) are set to 0 throughout the analysis.

**Remark 1.** In our method, \( \omega \) affects both convergence and consensus. As a heuristic, we also propose a variant where the value of \( \omega \) varies based on the magnitude of the norms, \( cm_1 = \|x_k^j - x_{k-1}^j\| \) and \( cm_2 = \|v_k^j - v_{k-1}^j\| \). Based on the norm, the \( \omega \) can be computed as \( \omega = \frac{cm_1}{cm_1 + cm_2} \).

This is inspired by methods that adaptively adjust the size of the gradient in momentum-based optimization, such as Adam (Kingma & Ba, 2014).

We now rewrite the update law with option I in a vector form as:

\[ x_{k+1} = x_k - \alpha g(x_k) + \frac{1}{\alpha}(I - \Pi)x_k \]

\[ + \beta(\omega I + (1 - \omega)\Pi)(x_k - x_{k-1}) \tag{4} \]
the supplementary materials. We first define \( \tilde{x}_k = x_k - x_k \) and construct the linear time-invariant system for \([x_{k+1}; x_k]\). Then by induction and setting initialization 0, we can express \([\tilde{x}_{k+1}; x_k]\) using only the coefficient matrices and stochastic gradient inputs. By leveraging the decomposition techniques in matrices, the upper bound of matrix norms is obtained correspondingly. Hence, the iterates converge to the consensus estimate.

**Remark 2.** Proposition 1 provides a uniform consensus error upper bound among agents, proportional to the step size \(\alpha\) and the number of agents \(N\) and inversely proportional to the gap between the largest and second-largest (in magnitude) eigenvalues of \(\beta\Pi\). When \(\omega = 0\), DMSGD achieves the "best" consensus; the upper bound simplifies to \(\frac{8\alpha\sqrt{M^2 + \sigma^2}}{\sqrt{\eta(1-\beta\lambda)(1-\sqrt{\beta\lambda})}}\). When \(\omega \to 1\), we get a worse-case upper bound on consensus error. Further, a more connected graph has a smaller value of \(\lambda\), implying better consensus (which makes intuitive sense).

**Strongly convex functions.** Next, we study the convergence for strongly convex objective functions; in this case, the algorithm converges to a globally optimal solution. We show convergence by upper bounding \(E[F(x) - F^*]\). Before showing the main results, we present two auxiliary lemmas.

**Lemma 1.** Let all stated assumptions hold. Suppose that \(F(x)\) for all \(x \in \mathbb{R}^d\) is \(\mu\)-strongly convex. Then the following relationship holds:

\[
- \langle \nabla F(x), \nabla F(x) \rangle \leq -\mu^2 \|x - x^*\|^2 \leq -\frac{2\mu^2}{L'} (F(x) - F^*)
\]

where \(x^* = \arg \min_{x \in \mathbb{R}^d} F(x)\), \(F^* := F(x^*) > -\infty\).

**Proof.** Leveraging the definition of strong convexity and smoothness yields the desired result. Please see the supplementary materials for detailed proof.

**Lemma 2.** Let all assumptions hold. Then the iterates generated by DMSGD \(\forall k \in \mathbb{N}, \exists \alpha > 0\) satisfy the following relationship:

\[
E[\|x_{k+1} - x_k\|^2] \leq \frac{\alpha^2 (M^2 + \sigma^2)}{(1-\beta)^2}
\]

where \(\Lambda = \omega + (1-\omega)\lambda_2\). Alternatively, we have a tighter bound as

\[
E[\|x_{k+1} - x_k\|^2] \leq \frac{(1 - (\Lambda\beta)^{k+1})^2 \alpha^2 (M^2 + \sigma^2)}{(1-\beta)^2}
\]

**Proof.** From the tighter bound, we can immediately obtain the first inequality. To get the tighter bound, we first recall the vector form of update law and rewrite it into \(x_{k+1} - x_k = -\alpha S(x_k) + \beta\Pi(x_k - x_k)\). By induction and setting initialization as 0, with the upper bounded stochastic gradient by \(\sqrt{M^2 + \sigma^2}\), we get the desired results. Please refer to the supplementary material for details.

We now state the main result for convergence under strongly convex functions.

**Theorem 1.** (Strongly convex case) Let all assumptions hold. The iterates \(\{x_k\}\) generated by DMSGD with \(0 < \alpha \leq \frac{L'}{2\mu^2}\) and \(0 \leq \beta < 1\) satisfy the following inequality for all \(k \in \mathbb{N}\):

\[
E[F(x_{k+1}) - F^*] \leq (1 - 2\alpha \mu^2)E[F(x_k) - F^*] + \alpha M \sigma

+ \alpha M \sqrt{M^2 + \sigma^2} + \frac{L' \alpha^2 (M^2 + \sigma^2)}{2(1-\beta\lambda)^2},
\]

where \(\Lambda = \omega + (1-\omega)\lambda_2\). Alternatively, we can have a tighter bound as follows:

\[
E[F(x_{k+1}) - F^*] \leq (1 - 2\alpha \mu^2)E[F(x_k) - F^*] + \alpha M \sigma

+ \alpha M \sqrt{M^2 + \sigma^2} \frac{1 - (\Lambda\beta)^k}{1-\beta\lambda} + \frac{L' (1 - (\Lambda\beta)^{k+1})^2 \alpha^2 (M^2 + \sigma^2)}{2(1-\beta\lambda)^2}.
\]

**Proof.** We provide the proof sketch here. Based on the smoothness condition of \(F(x)\), \(F_{k+1} - F_k \leq \langle \nabla F(x_k), x_{k+1} - x_k \rangle + \frac{L'}{2} \|x_{k+1} - x_k\|^2\), substituting the update law and using Lemmas 1 and 2, with some mathematical manipulations, the desired result can be obtained. Please refer to supplementary material for proof.

Immediately, Theorem 1 implies the following corollary to show that with a properly set constant stepsize \(\alpha\), DMSGD enables the iterates \(\{x_k\}\) to converge to a solution that is proportional to \(\alpha\) in a linear rate.

**Corollary 1.** Let all stated assumptions hold. The iterates generated by DMSGD with \(0 < \alpha \leq \frac{L'}{2\mu^2}\) and \(0 \leq \beta < 1\) satisfy the following relationship \(\forall k \in \mathbb{N}\):

\[
E[F(x_k) - F^*] \leq \frac{RL'}{2\alpha \mu^2}

+ (1 - 2\alpha \mu^2)k^{-1} (F(x_1) - F^*) - \frac{RL'}{2\alpha \mu^2},
\]

where \(R = \alpha M \sigma + \alpha M \sqrt{M^2 + \sigma^2} \frac{1}{1-\beta\lambda} + \frac{L' \alpha^2 (M^2 + \sigma^2)}{2(1-\beta\lambda)^2}\).

Corollary 1 suggests that when \(k \to \infty\), DMSGD enables the iterates \(\{x_k\}\) to converge within \(\frac{RL'}{2\alpha \mu^2}\) from the optimal point \(x^*\). The error bound is essentially with respect to the variance of stochastic gradient and the network error among agents when substituting \(R\) into \(\frac{RL'}{2\alpha \mu^2}\). Also, when \(\sigma = 0\) and \(\alpha \to 0\), the iterates \(\{x_k\}\) converges to \(x^*\).

Due to the fact that \(E[F(x_k)] \leq E[F(x_k)]\) and that \(F(x^*) = F(x^*)\), the sequence of true objective function values are bounded above by \(E[F(x_k) - F(x)] \leq E[F(x_k) - F(x^*)]\). Hence, Corollary 1 also implies that
we can establish the analogous convergence rates in terms of the true objective function value sequence \( \{F(x_k)\} \).

**Polyak-Łojasiewicz Inequality.** It is well known that in the deterministic case, under gradient descent with strongly-convex and smoothness assumptions, a global linear convergence rate can be achieved. Recently, SGD and other stochastic variants have also shown to have linear convergence (Karimi et al., 2016; Gower et al., 2019). However, most models, even simple deep learning models, are not strongly-convex. In this context, we introduce the Polyak-Łojasiewicz (PL) inequality, which enables the gradient to grow as a quadratic function of sub-optimality. With PL condition and smoothness, using gradient descent, it is shown to achieve a linear rate. Further, in Karimi et al. (2016); Gower et al. (2019), with constant step size, the stochastic gradient descent can converge to the neighborhood of \( x^* \). To the best of our knowledge, it is the first time to show that the PL condition enables a decentralized momentum SGD to converge at a linear rate.

**Theorem 2. (Quasi convex case)** Let all assumptions hold. Suppose that \( F(x) \) satisfies the Polyak-Łojasiewicz inequality such that

\[
||\nabla F(x)||^2 \geq 2\hat{\mu}(F(x) - F^*), \quad \forall x \in \mathbb{R}^d
\]

The iterates \( \{x_k\} \) generated by DMSGD with \( 0 < \alpha \leq \frac{1}{2\hat{\mu}} \) and \( 0 \leq \beta < 1 \) satisfy the following inequality for all \( k \in \mathbb{N} \)

\[
\mathbb{E}[F(x_k) - F^*] \leq \frac{R}{2\alpha \hat{\mu}^2} + (1 - 2\alpha \hat{\mu})^{k-1}(F(x_1) - F^*) - \frac{R}{2\alpha \hat{\mu}^2}
\]

where \( R = \alpha M \sigma + \alpha M \sqrt{M^2 + \sigma^2} \frac{1}{1 - \beta \Lambda} + \frac{L'(M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2} \).

**Proof.** The proof directly follows from Theorem 1 and Corollary 1 by leveraging the PL inequality. Please refer to the supplementary materials for more details.

It can be observed that with only smoothness and PL condition, DMSGD still converges in a linear rate to the neighborhood. However, it should be noted that the strong convexity can imply PL condition, which is presented in Lemma 3 for completeness, but not vice versa. That is the reason why we use \( \hat{\mu} \), not \( \mu \). Since when objective function \( F(x) \) is strongly convex, \( -||\nabla F(x_k)||^2 \leq -2\mu'(F(x_k) - F^*) \) holds. The following result is well-known:

**Lemma 3.** If \( F(x) \) is \( \mu' \)- strongly convex then it satisfies the Polyak-Łojasiewicz condition, that is

\[
||\nabla F(x_k)||^2 \geq 2\mu'(F(x_k) - F^*).
\]

**Non-convex functions.** We next discuss the case of non-convex function. We summarize the main result on the convergence of DMSGD for non-convex function in Theorem 3. But first, we give an auxiliary lemma to simplify the proof process for Theorem 3.

**Lemma 4.** Let all assumptions hold. The iterates generated by DMSGD with \( 0 \leq \beta < 1 \) satisfy the following inequality for all \( k \in \mathbb{N}, \exists \alpha > 0 \),

\[
\mathbb{E}[F(x_{k+1}) - F(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[||\nabla F(x_k)||^2] + \frac{(L'\alpha^2 - \alpha)(M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2} + \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{1 - \beta \Lambda} + \frac{\alpha(\beta \Lambda)^2(M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
\]

where \( \Lambda = \omega + (1 - \omega)\lambda_2 \). Alternatively, we have a tighter bound as

\[
\mathbb{E}[F(x_{k+1}) - F(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[||\nabla F(x_k)||^2] + \frac{(1 - (\beta \Lambda)^{k+1})^2}{(1 - \beta \Lambda)^2} (M^2 + \sigma^2) + \frac{\alpha \sigma}{2} \frac{\sqrt{M^2 + \sigma^2} \beta \Lambda (1 - \beta \Lambda)^k}{1 - \beta \Lambda} + \frac{\alpha (\beta \Lambda)^2 (1 - (\beta \Lambda)^k)^2 (M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2}.
\]

**Proof.** As the proof for this lemma is fairly technical, we refer interested readers to the supplementary materials for the detailed proof. The proof requires the smoothness condition of \( F(x) \), the update law, and some inequalities. \( \square \)

**Theorem 3. (Non-convex case)** Let all assumptions hold. The iterates generated by DMSGD with \( 0 \leq \beta < 1 \) satisfy the following inequality, when selecting

\[
\alpha = \frac{(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) - \sigma^2(1 - \beta \Lambda)^2}{L'(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) - 2(1 - \beta \Lambda)\beta \Lambda \sigma \sqrt{M^2 + \sigma^2} L'(M^2 + \sigma^2)}
\]

\[
\frac{1}{k+1} \sum_{m=0}^{k} \mathbb{E}[||\nabla F(x_m)||^2] \leq \frac{2(F(x_0) - F^*)}{\alpha(k + 1)}
\]

Alternatively, without variance, when selecting \( \alpha = \frac{1}{L'(M^2 + \sigma^2)} \), the last inequality still holds.

**Proof.** Using the conclusion from Lemma 4, by induction, we can get the desired results. Please refer to the supplementary materials for more details. \( \square \)
We can observe that the step size in Theorem 3 is complex when the variance is existing and the positiveness needs verification. In the supplementary materials, we show that \( \alpha > 0 \) naturally.

Theorem 3 shows that with a properly selected constant stepsize, for non-convex functions, DMSGD can converge to the optimal solution \( x^\ast \) (which essentially is a stationary point) with a rate of \( \mathcal{O}(\frac{1}{k}) \). This improves previous results that it only converges to the neighborhood of \( x^\ast \) (Jiang et al., 2017; Lian et al., 2017; Yu et al., 2019). In previous works, they required the diminishing step size to converge to \( x^\ast \) such that it slowed down the convergence. Hence, the constant step sizes in Theorem 3 are in an optimal sense. However, the determination of \( \alpha \) can be challenging as one needs to know the smoothness constant \( L' \) and estimate the variance, which is typically unknown.

5. Experimental Results

We now support the utility of our proposed DMSGD algorithm by simulating a distributed environment over a GPU cluster with multiple GPUs, similar to the experiments of Lin et al. (2014); Jiang et al. (2017); Tang et al. (2018). We define a graph topology where each agent in the graph can communicate with another agent with an interaction matrix initialized by the user, ensuring that it is doubly stochastic (in our experiments, we explore a fully connected topology, a ring topology, and a bipartite graph just as in Asrani et al. (2018)).

We split the given (complete) training dataset among different agents equally, creating two data simulation strategies:

1. **iid**: the dataset is shuffled completely and distributed amongst the agents to simulate an environment where each of the agents has an independently identical draw from the data distribution.

2. **non-iid**: We first segregate the dataset based on the target labels, then we create chunks of data and distribute the chunks with unique target labels to all the agents. If the number of agents is larger than the number of target labels, each agent gets only a chunk of data corresponding to each target label, and if the number of agents is lesser than the number of target labels, each agent gets a set of multiple chunks with unique target labels unavailable with other agents. This strategy simulates an extreme imbalance across different agents and we expect to see significant loss in the performance of decentralized learning algorithms.

In this work, we implement proposed algorithms with both the data simulation strategies. The implementation is using Pytorch (Paszke et al., 2017). Code to reproduce our results is submitted along with the supplementary material.

Another empirical evidence of good consensus is that it produces a lesser generalization gap (Jiang et al., 2017; Tang et al., 2018). In Figure 2, the dotted lines denote the performance of agents on test data, which closely follow the solid line (performance on training data) but lag slightly. We attribute to the averaging of several weights, which promotes generalization, as explained in (Izmailov et al., 2018; Huang et al., 2017). In (Izmailov et al., 2018), authors show that by averaging the weights of the network, they get wider and flat optima that generalize well. We note that the consensus step provides us with similar conditions. Another observation from our experiments is a validation of Remark 2; we see that as \( \omega \) increases, the generalization gap increases with a weaker consensus bound occurring at \( \omega \to 1 \) as explained in Remark 2. Therefore, we see that at \( \omega = 0.99 \), our algorithm does not converge.

Now, we analyze the convergence and performance of the DMSGD algorithm. Due to space constraints, we only present a few anecdotal results here. In Figure 2(a and b),
we show the performance of DMSGD (I and II respectively) with different $\omega$ values for CIFAR-10 dataset. All the results shown here are for a sufficiently large Convolutional Neural Network, whose details are presented in the supplementary material. While we could perform comparisons with the algorithm proposed by Assran et al. (2018); Kamp et al. (2018), it would be unfair as the protocol for communication used by them is different (Push-Sum and Dynamic Model Averaging). Note that we could extend our momentum-accelerated consensus to these models, analysis of the same is beyond the scope of this work. Therefore, as a baseline, we use a non-momentum decentralized algorithm that would have a fair comparison. For this, we compare with CDSGD (Jiang et al., 2017) in this simple experiment. We observe that DMSGD-I performs with similar performance as the CDSGD algorithm, i.e., without any acceleration. At the same time, DMSGD-II performs much faster and better for the same task. These results are corresponding to the iid data simulation strategy. However, while working on a non-iid data simulation strategy, we observe that DMSGD-II does not perform well. On the contrary, DMSGD-I performs better than the CDSGD algorithm. We believe that this is a trade-off between consensus and convergence, which Li et al. (2019) explores in detail.

We also note from the results shown and the analysis in the previous section that as $\omega \to 1$, the convergence bounds become weaker. This explains why the performance dies down as a function of $\omega$, e.g. $\omega = 0.5$ performs better than $\omega = 0.75$. However, setting $\omega = 0$ performs very badly for non-iid data. The dynamics of $\omega$ with respect to the data distribution is not explored in this work and can be considered as future work.

Finally, we would like to add another result for the performance of our proposed DMSGD-I algorithm for different communication topologies in Figure 3. We consider three communication topologies: (1) Fully connected topology (2) Ring topology (3) Bipartite topology. As the communication topology has sparse communication, the consensus and convergence bounds also become weaker. In Figure 3, where we see that the Bipartite graph with very sparse connections performs worse than fully connected graph, which validates the analysis.

6. Conclusions and Future Work

This paper addresses the problem of deep learning in a decentralized setting using momentum accelerated consensus. We establish a consensus-based decentralized learning algorithm using the stochastic heavy ball approach that can assist in finding the optimal solution faster than conventional SGD-style methods. We show that the proposed DMSGD with different choices of momentum terms can achieve linear convergence rate with appropriately chosen step size for strongly-convex, quasi-convex objective functions along with the assumption of smoothness, and convergence to a stationary point for non-convex objective functions.

Relevant experimental results using benchmark datasets show that the proposed algorithms can achieve better accuracy with sufficient training epochs. While our current research focuses on extensive testing and validation of the proposed framework (especially for large networks), a few directions for future research include an extension to the analysis of Nesterov momentum with nonconvex objective functions, analysis of non-iid data setting and variance reduction strategies for further convergence speed-up techniques in the stochastic setting.
References

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

Berahas, A. S., Bollapragada, R., Keskar, N. S., and Wei, E. Balancing communication and computation in distributed optimization. *IEEE Transactions on Automatic Control*, 64(8):3141–3155, 2018.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. Signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

Botev, A., lever, G., and Barber, D. Nesterov’s accelerated gradient and momentum as approximations to regularised update descent. *arXiv preprint arXiv:1607.01981*, 2016.

Chen, J., Pan, X., Monga, R., Bengio, S., and Jozefowicz, R. Revisiting distributed synchronous sgd. *arXiv preprint arXiv:1604.00981*, 2016.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al. Large scale distributed deep networks. *Advances in neural information processing systems*, pp. 1223–1231, 2012.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *arXiv preprint arXiv:1802.10026*, 2018.

Ghadimi, E., Shames, I., and Johansson, M. Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*, 61(21):5417–5429, 2013.

Ghadimi, E., Feyzmahdavian, H. R., and Johansson, M. Global convergence of the heavy-ball method for convex optimization. In *Control Conference (ECC), 2015 European*, pp. 310–315. IEEE, 2015.

Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. Sgd: General analysis and improved rates. *arXiv preprint arXiv:1901.09401*, 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.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Jakovetic, D., Xavier, J., and Moura, J. M. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, pp. 5906–5916, 2017.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. On consensus-optimality trade-offs in collaborative deep learning. *arXiv preprint arXiv:1805.12120*, 2018.

Jin, P. H., Yuan, Q., Iandola, F., and Keutzer, K. How to scale distributed deep learning? *arXiv preprint arXiv:1611.04581*, 2016.

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.

Kamp, M., Adilova, L., Sicking, J., Hüger, F., Schlicht, P., Wirtz, T., and Wrobel, S. Efficient decentralized deep learning by dynamic model averaging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 393–409. Springer, 2018.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

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

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

Li, X., Yang, W., Wang, S., and Zhang, Z. Communication efficient decentralized training with multiple local updates. *arXiv preprint arXiv:1910.09126*, 2019.

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.

Lin, Q., Lu, Z., and Xiao, L. An accelerated proximal coordinate method. In Advances in Neural Information Processing Systems, pp. 3059–3067, 2014.

Loizou, N. and Richtárik, P. Linearly convergent stochastic heavy ball method for minimizing generalization error. arXiv preprint arXiv:1710.10737, 2017.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629, 2016.

Nadiradze, G., Sabour, A., Sharma, A., Markov, I., Aksenov, V., and Alistarh, D. Popsgd: Decentralized stochastic gradient descent in the population model. arXiv preprint arXiv:1910.12308, 2019.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In Soviet Mathematics Doklady, volume 27, pp. 372–376, 1983.

Nesterov, Y. Gradient methods for minimizing composite functions. Mathematical Programming, 140(1):125–161, 2013a.

Nesterov, Y. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013b.

Nesterov, Y., Stich, S., et al. Efficiency of accelerated coordinate descent method on structured optimization problems. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2016.

Ochs, P., Brox, T., and Pock, T. ipiasco: Inertial proximal algorithm for strongly convex optimization. Journal of Mathematical Imaging and Vision, 53(2):171–181, 2015.

Paszke, A., Gross, S., Chintala, S., and Chanan, G. Pytorch, 2017.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 4(5):1–17, 1964.

Su, W., Boyd, S., and Candes, E. J. A differential equation for modeling nesterov’s accelerated gradient method: theory and insights. Journal of Machine Learning Research, 17(153):1–43, 2016.

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.
A. Supplementary Materials for 
“Decentralized Momentum Stochastic 
Gradient Descent for Multi-agent 
Learning”

This section presents additional analytical and experimental results. The statements of the lemmas and theorems with their detailed proof for completeness.

A.1. Additional Analytical Results and Proofs

We denote by $\Lambda$ the equivalent second large eigenvalue of $\Pi$ such that $\Lambda = \omega + (1 - \omega)\lambda_2$. $\lambda_2 < 1$ and $\omega \in [0, 1]$ such that $\Lambda \leq 1$.

**Proposition 1** Let all assumptions hold, the iterates generated by DMSGD satisfies the following inequality $\forall k \in \mathbb{N}, \exists \alpha > 0$,

$$
\mathbb{E}||x_{k+1}^2 - \bar{x}_k|| \leq \frac{8\alpha \sqrt{N} \sqrt{M^2 + \sigma^2}}{\sqrt{\eta(1 - \beta\Lambda)}(1 - \sqrt{\beta\Lambda})}
$$

where, $\eta$ is defined as an arbitrarily small positive constant such that $\Pi \succ \eta I$, $\eta < 1$, $\Lambda = \omega + (1 - \omega)\lambda_2$.

**Proof.** We define

$$
\tilde{x}_k = x_k - \bar{x}_k
$$

the compact form of which is written as

$$
\bar{x}_k = x_k - \bar{x}_k
$$

Similarly, we have $\tilde{x}_{k+1} = x_{k+1} - \bar{x}_{k+1}$ and next construct the linear time-invariant system for $[\tilde{x}_{k+1}; \tilde{x}_k]$. Let $J = \frac{1}{N}11^T$, where $1$ is a $N \times 1$ dimension vector with entries being 1. Substituting the update law 5 into $\tilde{x}_{k+1}$, we have

$$
\tilde{x}_{k+1} = (I - J)x_k - \alpha(I - J)S(x_k) + \beta(I - J)\Pi(I - J)
$$

As $(I - J)\Pi = (I - J)(\omega I + (1 - \omega)\Pi) = \omega(I - J) + (1 - \omega)\Pi(I - J)$

$$
\lambda_2(I - J) + (1 - \omega)\Pi(I - J) = \omega(I - J) + (1 - \omega)\Pi(I - J) = \lambda(I - J)
$$

Where the third inequality holds due to $\Pi$ being doubly stochastic, then Eq. 13 can be rewritten as

$$
\tilde{x}_{k+1} = \tilde{x}_k - \alpha(I - J)S(x_k) + \beta\Pi(I - J)(x_k - \bar{x}_{k-1})
$$

Hence, we can obtain the linear time-invariant system for $[\tilde{x}_{k+1}; \tilde{x}_k]$.

$$
\begin{bmatrix}
\tilde{x}_{k+1} \\
\tilde{x}_k
\end{bmatrix} =
\begin{bmatrix}
I + \beta\Pi & -\beta\Pi \\
I & 0
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_k \\
\tilde{x}_{k-1}
\end{bmatrix} - \alpha
\begin{bmatrix}
(I - J)S(x_k) \\
0
\end{bmatrix}
$$

Eq. 14 shows an equivalent linear time-invariant system with an input with respect to stochastic gradient.

We define

$$
\tilde{X}_{k+1} = \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{x}_k \end{bmatrix}
$$

$$
A = \begin{bmatrix} I + \beta\Pi & -\beta\Pi \\ I & 0 \end{bmatrix}, B_k = (I - J)S(x_k)
$$

such that

$$
\tilde{X}_{k+1} = A\tilde{X}_k - \alpha B_k
$$

We perform induction on $k$ for Eq. 15 and have the following relationship.

$$
\tilde{X}_{k+1} = A^k\tilde{X}_0 - \alpha \sum_{s=0}^{k} A^{k-s}B_s
$$

The next step of the proof is to analyze $A^k$, $\forall k \geq s$ using Schur decomposition and to bound the norm of it. It can directly follow from the proof of Theorem 4 in Jakovetic et al. (2014) and we can arrive at

$$
||A^{-s}|| \leq \frac{8(\sqrt{\beta\Lambda})^{k-s}}{\sqrt{\eta(1 - \beta\Lambda)}}
$$

We next bound the norm of $B_k$, $\forall k \geq 0$. $B_k$ only consists of $(I - J)S(x_k)$ and 0 such that essentially it is a column vector in this context. Therefore, we have

$$
||B_k|| = ||(I - J)S(x_k)|| \leq ||I - J|| ||S(x_k)||\sqrt{N},
$$

By taking the expectation on both sides, we have

$$
\mathbb{E}||B_k|| \leq \mathbb{E}||S(x_k)||\sqrt{N} \leq \sqrt{M^2 + \sigma^2}\sqrt{N},
$$

Hence, with initialization being 0, we have

$$
||\tilde{X}_0|| = \alpha||\sum_{s=0}^{k} A^{k-s}B_s|| \leq \alpha \sum_{s=0}^{k} ||A^{k-s}||||B_s||
$$

Taking expectation on both sides yields

$$
\mathbb{E}||\tilde{X}_0|| \leq \alpha \sum_{s=0}^{k} ||A^{k-s}||\mathbb{E}||B_s||
$$

$$
\leq \alpha \sum_{s=0}^{k} \frac{8(\sqrt{\beta\Lambda})^{k-s}}{\sqrt{\eta(1 - \beta\Lambda)}}\sqrt{N}\sqrt{M^2 + \sigma^2}
$$

$$
\leq \frac{8\alpha\sqrt{N}\sqrt{M^2 + \sigma^2}}{\sqrt{\eta(1 - \beta\Lambda)}(1 - \sqrt{\beta\Lambda})}
$$
which completes the proof with \( \| x_k' - \bar{x}_k \| \leq \| \bar{x}_k \| \). \( \square \)

**Lemma 1**: Let all assumptions hold. Suppose that \( F(x) \) for all \( x \in \mathbb{R}^d \) is \( \mu' \)-strongly convex. Then the following relationship holds

\[
-\langle \nabla F(x), \nabla F(x) \rangle \leq -\mu'^2 \| x - x^* \|^2 \leq -\frac{2\mu'^2}{L} (F(x) - F^*)
\]

where \( x^* = \arg \min_{x \in \mathbb{R}^d} F(x) \), \( F^* := F(x^*) > -\infty \).

**Proof.** As \( F(x) \) is \( \mu' \)-strongly convex, we can immediately obtain the following relationship based on its definition,

\[
(\nabla F(x) - \nabla F(y))^T (x - y) \geq \mu' \| x - y \|^2
\]

for all \( x, y \in \mathbb{R}^d \). Using Cauchy-Schwarz inequality, we have

\[
\| \nabla F(x) - \nabla F(y) \| \| x - y \| \geq \mu' \| x - y \|^2
\]

Let \( y = x^* \) such that

\[
\| \nabla F(x) - \nabla F(x^*) \| \geq \mu' \| x - x^* \|
\]

Since \( \nabla F^* = 0 \), we have

\[
-\langle \nabla F(x), \nabla F(x) \rangle \leq -\mu'^2 \| x - x^* \|^2
\]

Further, based on the definition of smoothness, we have

\[
F(x) \leq F^* + \frac{L'}{2} \| x - x^* \|^2
\]

combining the last two inequalities completes the proof. \( \square \)

**Lemma 2**: Let all assumptions hold. Then the iterates generated by DMSGD \( \forall k \in \mathbb{N} \), \( \exists \alpha > 0 \), satisfy the following relationship:

\[
\mathbb{E}[\| x_{k+1} - x_k \|^2] \leq \frac{\alpha^2 (M^2 + \sigma^2)}{(1 - \Lambda \beta)^2}
\]

where \( \Lambda = \omega + (1 - \omega) \lambda_2 \). Alternatively, we have a tighter bound as

\[
\mathbb{E}[\| x_{k+1} - x_k \|^2] \leq \frac{(1 - (\Lambda \beta)^k + 1) \alpha^2 (M^2 + \sigma^2)}{(1 - \Lambda \beta)^2}
\]

**Proof.** Recall the update law in a vector form:

\[
x_{k+1} = x_k - \alpha S(x_k) + \beta \Pi(x_k - x_k-1)
\]

such that

\[
x_{k+1} - x_k = -\alpha S(x_k) + \beta \Pi(x_k - x_k-1)
\]

By induction, we can have

\[
x_{k+1} - x_k = -\alpha \sum_{m=0}^{k} \beta^{k-m} \Pi^{k-m} S(x_m)
\]

Let \( T_{k+1} = \sum_{m=0}^{k} (\beta \Lambda)^m = \frac{1 - (\beta \Lambda)^{k+1}}{1 - \beta \Lambda} \), then

\[
\| x_{k+1} - x_k \|^2 = \alpha^2 \sum_{m=0}^{k} \frac{\beta^{k-m} \Pi^{k-m} S(x_m)}{T_{k+1}}^2 T_{k+1}^2
\]

\[
\leq \alpha^2 T_{k+1}^2 \sum_{m=0}^{k} \frac{\beta^{k-m} \Pi^{k-m} S(x_m)}{T_{k+1}}^2
\]

\[
\leq \alpha^2 T_{k+1} \sum_{m=0}^{k} (\beta \Lambda)^{k-m} \| S(x_m) \|^2
\]

(25)

Here, the second inequality follows from the convexity of \( \| \cdot \|^2 \) and Jensen’s inequality. The last inequality follows from the property of doubly stochastic matrix. Taking the expectation on both sides results in

\[
\mathbb{E}[\| x_{k+1} - x_k \|^2] \leq \frac{(1 - (\Lambda \beta)^k + 1)^2 \alpha^2 (M^2 + \sigma^2)}{(1 - \Lambda \beta)^2},
\]

which completes the proof. \( \square \)

**Theorem 1**: Let all assumptions hold. The iterates \( x_k \) generated by DMSGD with \( 0 < \alpha \leq \frac{1}{2\mu'} \) and \( 0 \leq \beta < 1 \) satisfy the following inequality for all \( k \in \mathbb{N} \).

\[
\mathbb{E}[\| x_{k+1} - x_k \|^2] \leq (1 - 2\alpha \mu'^2) \mathbb{E}[\| x_k \|^2] + \alpha M \sigma + \alpha M \sqrt{M^2 + \sigma^2} + \frac{L' \alpha^2 (M^2 + \sigma^2)}{2(1 - \Lambda \beta)^2},
\]

where \( \Lambda = \omega + (1 - \omega) \lambda_2 \). Alternatively, we can have a tighter bound as

\[
\mathbb{E}[\| x_{k+1} - x_k \|^2] \leq (1 - 2\alpha \mu'^2) \mathbb{E}[\| x_k \|^2] + \alpha M \sigma + \alpha M \sqrt{M^2 + \sigma^2} + \frac{L' \alpha^2 (M^2 + \sigma^2)}{2(1 - \Lambda \beta)^2}
\]

**Proof.** Based on the definition of smoothness, we can obtain that

\[
\| x_{k+1} - x_k \|^2 \leq (\nabla F(x_k), x_{k+1} - x_k) + \frac{L' \alpha}{2} \| x_{k+1} - x_k \|^2
\]

\[
= (\nabla F(x_k), -\alpha S(x_k) + \beta \Pi(x_k - x_k-1)) + \frac{L' \alpha}{2} \| x_{k+1} - x_k \|^2
\]

\[
= (\nabla F(x_k), -\alpha \nabla F(x_k)) + \alpha \nabla F(x_k) - \alpha S(x_k)) + (\nabla F(x_k), \beta \Pi(x_k - x_k-1)) + \frac{L' \alpha}{2} \| x_{k+1} - x_k \|^2
\]
According to Lemma 1, we have

\[ F(x_{k+1}) - F(x_k) \leq -\frac{2\alpha\mu^2}{L'}(F(x_k) - F^*) + \alpha||\nabla F(x_k)||||\nabla F(x_k) - S(x_k)|| + \alpha||\nabla F(x_k)||k-1 \sum_{m=0}^{k-1} (\beta\Lambda)^{k-1-m} ||S(x_m)|| + \frac{L'}{2}||x_{k+1} - x||^2 \]

The last inequality follows from Cauchy-Schwarz inequality. Hence, we have

\[ F(x_{k+1}) - F^* \leq F(x_k) - F^* - \frac{2\alpha\mu^2}{L'}(F(x_k) - F^*) + \alpha||\nabla F(x_k)||||\nabla F(x_k) - S(x_k)|| + \alpha||\nabla F(x_k)||k-1 \sum_{m=0}^{k-1} (\beta\Lambda)^{k-1-m} ||S(x_m)|| + \frac{L'}{2}||x_{k+1} - x||^2 \]

Rearranging the last inequality and taking expectations on both sides yields

\[ \mathbb{E}[F(x_{k+1}) - F^*] \leq (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] + \alpha\mathbb{E}[||\nabla F(x_k)||||\nabla F(x_k) - S(x_k)||] + \alpha\mathbb{E}[||\nabla F(x_k)||k-1 \sum_{m=0}^{k-1} (\beta\Lambda)^{k-1-m} ||S(x_m)||] + \frac{L'}{2}\mathbb{E}[||x_{k+1} - x||^2] \]

As \( \mathbb{E}[||\nabla F(x_k) - S(x_k)||] = \sqrt{(\mathbb{E}[||\nabla F(x_k) - S(x_k)||]^2) = \sqrt{\mathbb{E}[||\nabla F(x_k) - S(x_k)||]^2} = \sqrt{\sigma^2} = \sigma \)

Substituting the bound of \( S(x_k) \) and using Lemma 2, we have

\[ \mathbb{E}[||\nabla F(x_k) - S(x_k)||] \leq (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] + \alpha M\sigma + \alpha M\sqrt{M^2 + \sigma^2} + \frac{L'\alpha^2(M^2 + \sigma^2)(1 - (\beta\Lambda)^{k+1})}{2(1 - \beta\Lambda)^2} \]

which completes the proof. \( \square \)

**Corollary 1:** Let all assumptions hold. The iterates generated by DMGD with \( 0 < \alpha \leq \frac{L'}{2\alpha^2\mu^2} \) and \( 0 \leq \beta < 1 \) satisfy the following relationship \( \forall k \in \mathbb{N} \).

\[ \mathbb{E}[F(x_k) - F^*] \leq (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] + R \]

**Proof.** Using Theorem 1, we can obtain that

\[ \mathbb{E}[F(x_k) - F^*] \leq (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] + R \]

subtracting the constant \( \frac{RL'}{2\alpha\mu^2} \) from both sides, one can get

\[ \mathbb{E}[F(x_k) - F^*] - \frac{RL'}{2\alpha\mu^2} \leq (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] + R \]

\[ = (1 - \frac{2\alpha\mu^2}{L'})\mathbb{E}[F(x_k) - F^*] \]

We can know that the above inequality is a contraction inequality since \( 0 < \frac{2\alpha\mu^2}{L'} \leq 1 \) due to \( 0 < \alpha \leq \frac{L'}{2\mu^2} \). Hence the desired result is obtained by applying the inequality repeatedly through iteration \( k \in \mathbb{N} \). \( \square \)

**Theorem 2:** Let all assumptions hold. Suppose that \( F(x) \) satisfies the Polyak-Łojasiewicz inequality such that, \( \exists \tilde{\mu} > 0 \)

\[ ||\nabla F(x)||^2 \geq 2\tilde{\mu}(F(x) - F^*), \forall x \in \mathbb{R}^d \]
The iterates \( \{x_k\} \) generated by DMSGD with \( 0 < \alpha < \frac{1}{2\mu} \) and \( 0 < \beta < 1 \) satisfy the following inequality for all \( k \in \mathbb{N} \)

\[
\mathbb{E}[\mathcal{F}(x_k) - \mathcal{F}^*] \leq \frac{R'}{2\alpha \mu^2} + (1 - 2\alpha \hat{\mu})^{k-1} (\mathcal{F}(x_1) - \mathcal{F}^*)
\]

where \( R = \alpha M \sigma + \alpha M \sqrt{M^2 + 2\sigma^2} \frac{1}{\sqrt{1 - \beta \lambda}} + \frac{L' \alpha^2}{2(1 - \beta \lambda)^2} \)

**Proof.** According to the definition of smoothness, we have

\[
\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k) \leq \langle \nabla \mathcal{F}(x_k), x_{k+1} - x_k \rangle + \frac{L'}{2} ||x_{k+1} - x_k||^2
\]

As \( ||\nabla \mathcal{F}(x_k)||^2 \geq 2\hat{\mu}(\mathcal{F}(x_k) - \mathcal{F}^*) \), we get

\[-\alpha ||\nabla \mathcal{F}(x_k)||^2 \leq -2\alpha \hat{\mu}(\mathcal{F}(x_k) - \mathcal{F}^*)\]

We skip the rest of the proof as it directly follows from the proof of Theorem 1 and Corollary 1 immediately.

**Lemma 3:** If \( \mathcal{F}(x) \) is \( \mu' \)-strongly convex then it satisfies the Polyak-Łojasiewicz condition, that is

\[ ||\nabla \mathcal{F}(x)||^2 \geq 2\mu' (\mathcal{F}(x) - \mathcal{F}^*) \]

**Proof.** Based on the definition of strong convexity,

\[
\mathcal{F}(y) \geq \mathcal{F}(x) + \langle \nabla \mathcal{F}(x), y - x \rangle + \frac{\mu'}{2} ||y - x||^2, \forall x, y \in \mathbb{R}^d
\]

\[-\mathcal{F}(y) \leq -\mathcal{F}(x) + \langle \nabla \mathcal{F}(x), y - x \rangle \leq \frac{\mu'}{2} ||y - x||^2
\]

Let \( y = x^* \) such that

\[
\mathcal{F}(x) - \mathcal{F}(x^*) \leq \langle \nabla \mathcal{F}(x), x - x^* \rangle + \frac{\mu'}{2} ||x^* - x||^2
\]

\[
= -\frac{1}{2} ||\sqrt{\mu'}(x - x^*)||^2 - \frac{1}{\sqrt{\mu'}} \nabla \mathcal{F}(x)||^2
\]

\[
+ \frac{1}{2\mu'} ||\nabla \mathcal{F}(x)||^2
\]

\[
\leq \frac{1}{2\mu'} ||\nabla \mathcal{F}(x)||^2
\]

which completes the proof.

**Lemma 4** Let all assumptions hold. The iterates generated by DMSGD with \( 0 \leq \beta < 1 \) satisfy the following inequality for all \( k \in \mathbb{N}, \exists \alpha > 0 \),

\[
\mathbb{E}[\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[||\nabla \mathcal{F}(x_k)||^2]
\]

\[
+ \frac{(L' \alpha^2 - \alpha)(M^2 + \sigma^2)}{2(1 - \beta \lambda)^2} + \frac{\alpha \sigma^2}{2}
\]

\[
+ \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{1 - \beta \Lambda}
\]

\[
+ \frac{\alpha(\beta \Lambda)^2 (M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
\]

where \( \Lambda = \omega + (1 - \omega) \lambda_2 \). Alternatively, we have a tighter bound as

\[
\mathbb{E}[\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[||\nabla \mathcal{F}(x_k)||^2]
\]

\[
+ \frac{(L' \alpha^2 - \alpha)(M^2 + \sigma^2)}{2(1 - \beta \lambda)^2} + \frac{\alpha \sigma^2}{2}
\]

\[
+ \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{1 - \beta \Lambda}
\]

\[
+ \frac{\alpha(\beta \Lambda)^2 [1 - (\beta \Lambda)^k]^2 (M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
\]

**Proof.** Based on the smoothness condition, we have

\[
\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k) \leq \langle \nabla \mathcal{F}(x_k), x_{k+1} - x_k \rangle + \frac{L'}{2} ||x_{k+1} - x_k||^2
\]

\[
= \langle \nabla \mathcal{F}(x_k), -\alpha S(x_k) + \beta \Pi(x_k - x_{k-1}) \rangle
\]

\[
+ \frac{L'}{2} ||x_{k+1} - x_k||^2
\]

\[
= \langle \nabla \mathcal{F}(x_k), \alpha \Pi(x_k - x_{k-1}) \rangle
\]

\[
+ \frac{L'}{2} ||x_{k+1} - x_k||^2
\]

\[
\leq \frac{1}{2\mu'} ||\nabla \mathcal{F}(x_k)||^2
\]

which completes the proof.

*Momentum-Accelerated Consensus*
We now discuss the term \( \| \nabla F(x_k) \| - \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \| \) and denote it as \( T_1 \).

\[
T_1 = \| \nabla F(x_k) - S(x_k) - \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|
\]

\[
= \| \nabla F(x_k) - S(x_k) \|
\]

\[
- 2 \langle \nabla F(x_k) - S(x_k), \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \rangle
\]

\[
+ \left\| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \right\|^2
\]

Hence, we have

\[
\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k) \leq - \frac{\alpha}{2} \left( \| \nabla \mathcal{F}(x_k) \|^2 \right)
\]

\[
+ \| \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

\[
+ \frac{\alpha}{2} \left( \| \nabla \mathcal{F}(x_k) - S(x_k) \|^2 \right)
\]

\[
- 2 \langle \nabla \mathcal{F}(x_k) - S(x_k), \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \rangle
\]

\[
+ \left\| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \right\|^2
\]

\[
+ \frac{L' \alpha^2}{2} \| \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

\[
= - \frac{\alpha}{2} \| \nabla \mathcal{F}(x_k) \|^2
\]

\[
+ \left( \frac{L' \alpha^2}{2} - \frac{\alpha}{2} \right) \| \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

\[
+ \frac{\alpha}{2} \| \nabla \mathcal{F}(x_k) - S(x_k) \|^2
\]

\[
- \alpha \langle \nabla \mathcal{F}(x_k) - S(x_k), \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \rangle
\]

\[
+ \frac{\alpha}{2} \| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

\[
\leq - \frac{\alpha}{2} \| \nabla \mathcal{F}(x_k) \|^2
\]

\[
+ \left( \frac{L' \alpha^2}{2} - \frac{\alpha}{2} \right) \| \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

\[
+ \frac{\alpha}{2} \| \nabla \mathcal{F}(x_k) - S(x_k) \|^2
\]

\[
+ \alpha \| \nabla \mathcal{F}(x_k) - S(x_k) \| \| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|
\]

\[
+ \frac{\alpha}{2} \| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2
\]

Taking the expectations on both sides yields

\[
\mathbb{E}[\mathcal{F}(x_{k+1}) - \mathcal{F}(x_k)] \leq - \frac{\alpha}{2} \mathbb{E}\left[ \| \nabla \mathcal{F}(x_k) \|^2 \right]
\]

\[
+ \frac{L' \alpha^2 - \alpha}{2} \mathbb{E}\left[ \| \sum_{m=0}^k \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2 \right]
\]

\[
+ \frac{\alpha}{2} \mathbb{E}[\| \nabla \mathcal{F}(x_k) - S(x_k) \|^2]
\]

\[
+ \alpha \mathbb{E}[\| \nabla \mathcal{F}(x_k) - S(x_k) \| \| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|]
\]

\[
+ \frac{\alpha}{2} \mathbb{E}[\| \sum_{m=0}^{k-1} \beta^{k-m} \tilde{\Pi}^{k-m} S(x_m) \|^2]
\]

The last inequality follows from the Hölder’s inequality. We now discuss each term on the right hand side of last inequality. According to Lemma 2, we know that immediately
that
\[
\mathbb{E}\left[\| \sum_{m=0}^{k} \beta^{k-m} \Pi^{k-m} S(x_m) \|^2 \right] \leq \frac{(1 - (\beta \Lambda)^{k+1})^2}{(1 - (\beta \Lambda)^2)} (M^2 + \sigma^2)
\]
and
\[
\mathbb{E}\left[\| \sum_{m=0}^{k-1} \beta^{k-m} \Pi^{k-m} S(x_m) \|^2 \right]
= \mathbb{E}\left[\| \beta \Pi \sum_{m=0}^{k-1} \beta^{k-m-1} \Pi^{k-m-1} S(x_m) \|^2 \right]
\leq \mathbb{E}\left[\| \beta \Pi \|^2 \| \sum_{m=0}^{k-1} \beta^{k-m-1} \Pi^{k-m-1} S(x_m) \|^2 \right]
\leq \frac{(\beta \Lambda)^2 (1 - (\beta \Lambda)^k)^2 (M^2 + \sigma^2)}{(1 - (\beta \Lambda)^2)},
\]
\[
\mathbb{E}\left[\| \nabla F(x_k) - S(x_k) \|^2 \right] \leq \sigma^2,
\]
\[
\sqrt{\mathbb{E}\left[\| \nabla F(x_k) - S(x_k) \|^2 \right]} \sqrt{\mathbb{E}\left[\| \sum_{m=0}^{k-1} \beta^{k-m} \Pi^{k-m} S(x_m) \|^2 \right]}
\leq \sigma \sqrt{M^2 + \sigma^2 \beta \Lambda (1 - (\beta \Lambda)^k) / (1 - \beta \Lambda)}.
\]

Thus we obtain the following relationship.

\[
\mathbb{E}\left[\nabla F(x_{k+1}) - \nabla F(x_k) \right]
\leq -\frac{\alpha}{2} \mathbb{E}\left[\| \nabla F(x_k) \|^2 \right] + \frac{L' \alpha^2 - \alpha^2 (1 - (\beta \Lambda)^{k+1})^2}{2(1 - (\beta \Lambda)^2)} (M^2 + \sigma^2) + \frac{\alpha \sigma^2}{2} + \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda (1 - (\beta \Lambda)^k)}{2(1 - (\beta \Lambda)^2)} + \frac{\alpha \sigma^2 \beta \Lambda (1 - (\beta \Lambda)^k)^2 (M^2 + \sigma^2)}{2(1 - (\beta \Lambda)^2)}
\]

which completes the proof.

Now we are ready to state the theorem.

**Theorem 3** Let all assumptions hold. The iterates generated by DMSGD with $0 \leq \beta < 1$ satisfy the following inequality.

when selecting
\[
\alpha = \frac{(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) - \sigma^2 (1 - \beta \Lambda)^2}{L' (M^2 + \sigma^2)} - \frac{2(1 - \beta \Lambda) \beta \Lambda \sigma \sqrt{M^2 + \sigma^2}}{L' (M^2 + \sigma^2)} + \frac{1}{k+1} \sum_{m=0}^{k} \mathbb{E}[\| \nabla F(x_m) \|^2] \leq \frac{2(F(x_0) - F^*)}{\alpha (k+1)}
\]

Alternatively, without variance, when selecting $\alpha = \frac{1 - (\beta \Lambda)^2}{L'}$, the last inequality still holds.

**Proof.** Based on the conclusion from Lemma 4, by recalling
\[
\mathbb{E}[F(x_{k+1}) - F(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[\| \nabla F(x_k) \|^2] + \frac{(L' \alpha^2 - \alpha^2 (M^2 + \sigma^2))}{2(1 - \beta \Lambda)^2} + \frac{\alpha \sigma^2}{2} + \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{2(1 - \beta \Lambda)^2} + \frac{\alpha \sigma^2 \beta \Lambda (M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
\]

conducting induction for $k + 1$ times yields
\[
F^* - F(x_0) \leq \mathbb{E}[F(x_{k+1})] - F(x_0) \leq \frac{\alpha}{2} \sum_{m=0}^{k} \mathbb{E}[\| \nabla F(x_m) \|^2] + (k+1)
\]

Rearranging the last inequality and dividing both sides by $\alpha$, leads to
\[
\sum_{m=0}^{k} \mathbb{E}[\| \nabla F(x_m) \|^2] \leq \frac{2(F(x_0) - F^*)}{\alpha} + (k+1) \left[\frac{(L' \alpha^2 - \alpha^2 (M^2 + \sigma^2))}{2(1 - \beta \Lambda)^2} + \frac{\alpha \sigma^2}{2} + \frac{\alpha \sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{2(1 - \beta \Lambda)^2} + \frac{\alpha \sigma^2 \beta \Lambda (M^2 + \sigma^2)}{2(1 - \beta \Lambda)^2}\right]
\]

Substituting $\alpha$ into the last inequality and dividing both sides by $k + 1$ can result in the conclusion. Setting $\sigma = 0$ and $\alpha = \frac{1 - (\beta \Lambda)^2}{L'}$ also yields the same conclusion. The proof is complete.
We discuss the first inequality in Theorem 3 to make it valid. As we need
\[
(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) - \sigma^2(1 - \beta \Lambda)^2 - \]
\[2(1 - \beta \Lambda)\beta \Lambda \sigma \sqrt{M^2 + \sigma^2} > 0,
\]
then if supposing that this holds, we have
\[
(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) > \sigma^2(1 - \beta \Lambda)^2
\]
\[+ 2(1 - \beta \Lambda)\beta \Lambda \sigma \sqrt{M^2 + \sigma^2}
\]
Since \(M^2 + \sigma^2 > \sigma^2\), we now have
\[
(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) > (M^2 + \sigma^2)(1 - \beta \Lambda)^2 + 2(1 - \beta \Lambda)\beta \Lambda \sigma \sqrt{M^2 + \sigma^2}
\]
We discuss the first inequality
\[
(M^2 + \sigma^2)(1 - (\beta \Lambda)^2) > (M^2 + \sigma^2)(1 - \beta \Lambda)^2 + 2(1 - \beta \Lambda)\beta \Lambda \sigma \sqrt{M^2 + \sigma^2}
\]
Dividing both sides by \((1 - \beta \Lambda)\) and \(\sqrt{M^2 + \sigma^2}\), we have
\[
(1 + \beta \Lambda)\sqrt{M^2 + \sigma^2} > (1 - \beta \Lambda)\sqrt{M^2 + \sigma^2} + 2\beta \Lambda \sigma
\]
Then rearranging the inequality, we can obtain
\[
2\beta \Lambda \sqrt{M^2 + \sigma^2} > 2\beta \Lambda \sigma
\]
\[\sqrt{M^2 + \sigma^2} > \sigma,
\]
which naturally holds. □

**Convergence with Simpler Step Size.** We have so far shown the convergence rate with some constant step sizes such that DMSGD only enables \(\{x_k\}\) to converge to the neighborhood of the optimal solution \(x^*\) for both strongly convex and quasi-convex scenarios. For the non-convex scenario, though when the step size is appropriately set, DMSGD is able to asymptotically converge to \(x^*\), obtaining such a step size could be intractable due to the requirement of knowing \(L\) and \(M\) as well as \(\sigma\). However, theoretically speaking, to cancel out the variance caused by stochastic gradients and network errors caused by the disagreement, a simpler step size is required correspondingly, without explicitly knowing \(L\), \(M\), and \(\sigma\). In most previous works on distributed optimization or learning, when the step size is a function of the number of iterations, \(k\), SGD type of algorithms can asymptotically converge to a globally or locally optimal solution. Therefore, when the function is non-convex, in this context, we can also have such a step size pre-specified to allow DMSGD converging \(\{x_k\}\) asymptotically to \(x^*\). It is shown in the following main theorem.

**Theorem 4.** Let all assumptions hold. The iterates generated by DMSGD with \(0 \leq \beta < 1\) satisfy the following inequality, when \(\alpha = \sqrt{\frac{B}{\beta}}\), where \(B > 0\), for all \(k \geq 1\),
\[
\frac{1}{k} \sum_{m=1}^{k} \mathbb{E}[||\nabla F(x_m)||^2] \leq \frac{Q}{\sqrt{k}},
\]
where \(Q = \frac{2 \Delta}{\sqrt{B}} + \sqrt{B L' \frac{M^2 + \sigma^2}{(1 - \beta \Lambda)^2}}\), \(\Delta = F(x_1) - F^*\).

**Proof.** Based on Theorem 3,
\[
\sum_{m=1}^{k} \mathbb{E}[||\nabla F(x_m)||^2] \leq \frac{2(F(x_1) - F^*)}{\alpha^2} + k \left[ \frac{L'(\alpha - 1)(M^2 + \sigma^2)}{(1 - \beta \Lambda)^2} + \sigma^2 + \frac{2\sigma \sqrt{M^2 + \sigma^2} \beta \Lambda}{1 - \beta \Lambda} + \frac{(\beta \Lambda)^2(M^2 + \sigma^2)}{(1 - \beta \Lambda)^2} \right]
\]
substituting \(\Delta = F(x_1) - F^*\) and simplifying the above inequality, we get the following
\[
\sum_{m=1}^{k} \mathbb{E}[||\nabla F(x_m)||^2] \leq \frac{2 \Delta}{\alpha} + k \left[ \frac{L'(\alpha - 1)(M^2 + \sigma^2)}{(1 - \beta \Lambda)^2} + \frac{\sigma(1 - \beta \Lambda) + \beta \Lambda \sqrt{M^2 + \sigma^2} - (M^2 + \sigma^2)}{1 - \beta \Lambda} \right]
\]
since \(\sigma \leq \sqrt{M^2 + \sigma^2}\) and hence \(\sigma(1 - \beta \Lambda) \leq (1 - \beta \Lambda)\sqrt{M^2 + \sigma^2}\), we get
\[
\sum_{m=1}^{k} \mathbb{E}[||\nabla F(x_m)||^2] \leq \frac{2 \Delta}{\alpha} + k \frac{L'(\alpha)(M^2 + \sigma^2)}{(1 - \beta \Lambda)^2}.
\]
The last inequality follows from that
\[
\left[ \frac{\sigma(1 - \beta \Lambda) + (1 + \beta \Lambda) \sqrt{M^2 + \sigma^2} - (M^2 + \sigma^2)}{(1 - \beta \Lambda)^2} \right] < 0
\]
Substituting \(\alpha = \sqrt{\frac{B}{\beta}}\) and dividing both sides by \(k\) in the above inequality gives the result. □

**A.2. Additional Analytical Results and Proofs for Option II**

This section presents additional analytical results and proofs for DMSGD with option II.

Recall the update law Eq. 6,
\[
x_{k+1} = x_k - \alpha g(x_k) + \beta \Pi(x_k - x_{k-1}).
\]
Assuming that \(f_j\) are \(\gamma_j\)-Lipschitz continuous, we can obtain that \(F\) is \(\gamma_m := \max\{\gamma_1, \gamma_2, ..., \gamma_N\}\)-Lipschitz continuous. Similarly, we follow Assumption 3 such that the gradient of
We next show the convergence rate for the strongly convex function $F$. By denoting the vector form of the stochastic gradient of $F$ as $g(x)$ at point $x$, the following relationship can be obtained correspondingly:

$$
\nabla F(x) = \mathbb{E}[g(x)]
$$

(27)

To characterize the main analysis, we summarize the assumptions for DMSGD option II in the following. 

**Assumption 4.** The objective functions $f_j : \mathbb{R}^d \to \mathbb{R}$ are assumed to satisfy the following conditions: a) Each $f_j$ is $L_j$-smooth; b) each $f_j$ is proper (not everywhere infinite) and coercive; c) each $f_j$ is $\gamma_j$ Lipschitz-continuous.

**Assumption 5.** The (stochastic) gradients of $F$ satisfy:

(a) $\text{Var}(g(x)) = \mathbb{E}[\|g(x) - \nabla g(x)\|^2] \leq \sigma^2$

(b) $\nabla F(x) = \mathbb{E}[g(x)]$.

Hence, with the upper bound for the variance of $g(x)$, analogously, $\mathbb{E}[g(x)]$ is bounded above by $\sqrt{\gamma_m^2 + \sigma^2}$. It can be derived simply by using the relationship $\mathbb{E}[\|g(x)\|] \leq \sqrt{\mathbb{E}[\|g(x)\|^2]}$.

We next present the main results for DMSGD with option II. Note that, the proof techniques for proving the convergence either for strongly convex, quasi-convex, or non-convex functions are the same as those for option I. Therefore, we only present the statements for all lemmas, corollary, and theorems.

**Proposition 2. (Consensus for Option II)** Let Assumptions 4 and 5 hold. The iterates generated by DMSGD option II satisfies the following inequality $\forall k \in \mathbb{N}, \exists \alpha > 0$: $\{(x_{k+i} - \overline{x}_k)\}$:

$$
\mathbb{E}[\|x_k - \overline{x}_k\|^2] \leq \frac{8\alpha \sqrt{\Lambda} \sqrt{\gamma_m^2 + \sigma^2}}{\eta(1 - \beta \Lambda)(1 - \sqrt{\beta \Lambda})},
$$

(28)

where $\eta$ is defined as an arbitrarily small constant such that $\bar{\eta} \gg \eta I$, $0 < \eta < 1$, $\Lambda = \omega + (1 - \omega)\lambda_2$.

We next show the convergence rate for the strongly convex functions. Before that we present two auxiliary lemmas adapting from Lemmas 1 and 2.

**Lemma 5.** Let Assumptions 4 and 5 hold. Suppose that $F(x)$ for all $x \in \mathbb{R}^d$ is $\mu_m$-strongly convex. Then the following relationship holds:

$$
-\langle \nabla F(x), \nabla F(x) \rangle \leq -\mu_m^2 \|x - x^*\|^2 \leq -\frac{2\mu_m^2}{L_m} (F(x) - F^*)
$$

(29)

where $x^* = \arg\min_{x \in \mathbb{R}^d} F$, $F^* := F(x^*) > -\infty$.

**Lemma 6.** Let Assumptions 4 and 5 hold. Then the iterates generated by DMSGD option II $\forall k \in \mathbb{N}, \exists \alpha > 0$ satisfy the following relationship:

$$
\mathbb{E}[\|x_{k+1} - x_k\|^2] \leq \frac{\alpha^2 (\gamma_m^2 + \sigma^2)}{(1 - \Lambda \beta)^2}
$$

(30)

where $\Lambda = \omega + (1 - \omega)\lambda_2$. Alternatively, we have a tighter bound as

$$
\mathbb{E}[\|x_{k+1} - x_k\|^2] \leq \frac{(1 - (\Lambda \beta)^{k+1}) \alpha^2 (\gamma_m^2 + \sigma^2)}{(1 - \Lambda \beta)^2}
$$

(31)

**Theorem 5. (Strongly convex case for DMSGD Option II)** Let Assumptions 4 and 5 hold. The iterates $\{x_k\}$ generated by DMSGD option II with $0 < \alpha \leq \frac{L_m}{2\mu_m^2}$ and $0 \leq \beta < 1$ satisfy the following inequality for all $k \in \mathbb{N}$:

$$
\mathbb{E}[F(x_{k+1}) - F^*] \leq (1 - \frac{2\alpha \mu_m^2}{L_m}) \mathbb{E}[F(x_k) - F^*] + \alpha \gamma_m \sigma
$$

$$
+ \alpha \gamma_m \sqrt{\gamma_m^2 + \sigma^2} \frac{1 - (\Lambda \beta)^k}{1 - \beta \Lambda} + \frac{L_m (1 - (\Lambda \beta)^{k+1}) \alpha^2 (\gamma_m^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
$$

where $\Lambda = \omega + (1 - \omega)\lambda_2$. Alternatively, we can have a tighter bound as follows:

$$
\mathbb{E}[F(x_{k+1}) - F^*] \leq (1 - \frac{2\alpha \mu_m^2}{L_m}) \mathbb{E}[F(x_k) - F^*] + \alpha \gamma_m \sigma
$$

$$
+ \alpha \gamma_m \sqrt{\gamma_m^2 + \sigma^2} \frac{1 - (\Lambda \beta)^k}{1 - \beta \Lambda} + \frac{L_m (1 - (\Lambda \beta)^{k+1}) \alpha^2 (\gamma_m^2 + \sigma^2)}{2(1 - \beta \Lambda)^2}.
$$

**Corollary 2.** Let Assumptions 4 and 5 hold. The iterates generated by DMSGD Option II with $0 < \alpha \leq \frac{L_m}{2\mu_m^2}$ and $0 \leq \beta < 1$ satisfy the following relationship $\forall k \in \mathbb{N}$:

$$
\mathbb{E}[F(x_k) - F^*] \leq \frac{RL_m}{2\alpha \mu_m^2}
$$

$$
+ (1 - \frac{2\alpha \mu_m^2}{L_m})^{k-1} (F(x_1) - F^*) - \frac{RL_m}{2\alpha \mu_m^2},
$$

where $R = \alpha \gamma_m \sigma + \alpha \gamma_m \sqrt{\gamma_m^2 + \sigma^2} \frac{1 - (\Lambda \beta)^k}{1 - \beta \Lambda} + \frac{L_m \alpha^2 (\gamma_m^2 + \sigma^2)}{2(1 - \beta \Lambda)^2}$.

We next present the analysis for the objective functions satisfying the PL condition (quasi-convexity).
Next we present the convergence analysis for non-convex functions with DMSGD Option II. Before that, we give an auxiliary technical lemma to simplify the proof for the main theorem.

**Lemma 7.** Let Assumptions 4 and 5 hold. The iterates generated by DMSGD Option II with $0 < \alpha \leq \frac{1}{2\gamma_m}$ and $0 \leq \beta < 1$ satisfy the following inequality for all $k \in \mathbb{N}$, $\exists \alpha > 0$,

$$
\mathbb{E}[F(x_{k+1}) - F(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[\|\nabla F(x_k)\|^2] + \frac{(L_m\alpha^2 - \alpha)(\gamma_m^2 + \sigma^2)}{2(1 - \beta \Lambda)^2} + \frac{\alpha \hat{\sigma}^2}{2} + \frac{\alpha \hat{\sigma} \sqrt{\gamma_m^2 + \sigma^2}}{2(1 - \beta \Lambda)^2} + \frac{\alpha \hat{\sigma} \sqrt{(\gamma_m^2 + \sigma^2)}(1 - (\beta \Lambda)^k)}{2(1 - \beta \Lambda)^2},
$$

where $\Lambda = \omega + (1 - \omega)\lambda_2$. Alternatively, we have a tighter bound as

$$
\mathbb{E}[F(x_{k+1}) - F(x_k)] \leq -\frac{\alpha}{2} \mathbb{E}[\|\nabla F(x_k)\|^2] + \frac{L_m\alpha^2 - \alpha}{2} + \frac{\alpha \hat{\sigma}^2}{2} + \frac{\alpha \hat{\sigma} \sqrt{(\gamma_m^2 + \sigma^2)}}{2(1 - \beta \Lambda)^2} + \frac{\alpha \hat{\sigma} \sqrt{(1 - (\beta \Lambda)^k)}(\gamma_m^2 + \sigma^2)}{2(1 - \beta \Lambda)^2},
$$

By the above lemma, we can obtain the main result for the non-convex functions.

**Theorem 8.** *(Non-convex case for DMSGD Option II, with a simpler step size)* Let Assumptions 4 and 5 hold. The iterates generated by DMSGD with Option II with $0 \leq \beta < 1$ satisfy the following inequality, when selecting

$$
\alpha = \frac{\gamma_m^2 + \hat{\sigma}^2}{L_m(\gamma_m^2 + \sigma^2)} \frac{2(L_m \gamma_m^2 + \sigma^2)}{L_m(\gamma_m^2 + \sigma^2)} - 2(1 - \beta \Lambda)\beta \Lambda \sqrt{(\gamma_m^2 + \sigma^2)} + \frac{1}{k+1} \sum_{m=0}^{k} \mathbb{E}[\|\nabla F(x_m)\|^2] \leq \frac{2(F(x_0) - F^*)}{\alpha(k+1)}.
$$

Alternatively, without variance, when selecting $\alpha = \frac{1 - (\beta \Lambda)^2}{L_m}$, the last inequality still holds.

Analogously, we also discuss in this context about the convergence with a simpler step size for DMSGD with option II. The following result suggests a sublinear convergence rate, but differs from that for DMSGD with option I in some constants.

**Theorem 7.** *(Non-convex case for DMSGD Option II)* Let Assumptions 4 and 5 hold. The iterates generated by DMSGD Option II with $0 \leq \beta < 1$ satisfy the following inequality, when selecting

$$
\alpha = \frac{(\gamma_m^2 + \hat{\sigma}^2)(1 - (\beta \Lambda)^2) - \hat{\sigma}^2(1 - \beta \Lambda)^2}{L_m(\gamma_m^2 + \sigma^2)} - 2(1 - \beta \Lambda)\beta \Lambda \sqrt{(\gamma_m^2 + \sigma^2)} + \frac{1}{k+1} \sum_{m=0}^{k} \mathbb{E}[\|\nabla F(x_m)\|^2] \leq \frac{2(F(x_0) - F^*)}{\alpha(k+1)}.
$$

A.3. Additional Experimental Results

In this section, we provide additional experimental results. All the experiments were run on Paszke et al. (2017) package. We use a simple logistic regression model and a simple convolutional neural network whose architecture is shown in Figure 2. All the experiments are performed on a cluster with several GPUs (mostly Titan X(Pascal) architecture and some with Tesla P40 architecture). The RAM in each node is a minimum of 128GB. The code for the implementation is made available along with this supplementary material.

We first present some additional visualization of weight space based on training and test loss, similar to those provided in the main paper. Figure 4 and Figure 5 illustrates the loss surface for MNIST and Semeion datasets using a simple logistic regression model. The agents converge towards the consensus point and also towards a lower loss value. These results are similar to the one shown in main paper Figure 1.
Next, to strengthen the claim of our proposed algorithm’s performance in a non-iid data simulation strategy, we ran experiments using the MNIST dataset with different model architectures. Figure 6 shows the performance of both DMSGD-I and DMSGD-II with different $\omega$ values, and as mentioned in the main text, DMSGD-I can perform better than DMSGD-II under the non-iid data simulation strategy. In fact, for $\omega = 0.25$ and adaptive setting, our proposed algorithm outperforms CDSGD algorithm. Similarly, we observe this trend in Figure 7, where again, DMSGD-I is performing way better under the non-iid data constraint. Regardless of the $\omega$ value, DMSGD-I closely resembles the convergence rate of the CDSGD algorithm. Besides these, we also present experiments with the logistic regression model, which has higher accuracy as compared to the CNN model(a more complicated model). We observe the same trend in this scenario, as well.

We extend our empirical analysis of the proposed algorithm by running some experiments with different communication topologies. We select the agent interaction matrices for these topologies such that it is doubly stochastic and the second-largest eigenvalue($\lambda_2$) are well separated to understand the difference between the topologies. In Figure 8, we visualize the performance of our algorithm with different communication topologies using simple data like MNIST and a simple model like logistic regression. The performance for both DMSGD-I and DMSGD-II are almost similar across different topologies, but DMSGD-I has much superior performance in the non-iid setting than DMSGD-II. Further, as we increase the model complexity, we observe the impact of communication topology on the model performance, as shown in Figure 3 and Figure 9, where the performance of DMSGD-I deteriorates as the sparsity in the communication topology increases. From the figure, we also notice that the bipartite topology(most sparse) is the first to fail, followed by the ring topology, while the fully-connected topology still performs up to 300 epochs.

We then run the same experiments on the CIFAR10 dataset. In Figure 10 and Figure 11, we provide a comparison between the performance of iid and non-iid data simulation strategies with different topologies. Under the iid setting and with an adaptive $\omega$, we confirm the claim that DMSGD-II is performing better than DMSGD-I, but again, the different topology does not result in a significant difference in term of performance. On the other hand, in the non-iid data simulation strategies, though the accuracy is still climbing and having a relatively large variance, the approximate accuracy achieved by DMSGD-I is already $\sim$5% higher compared to DMSGD-II.

Besides varying the topologies, we also ran multiple experiments to show how the algorithm performs with a different number of agents. We observe that DMSGD-I performs similarly as the no. of agents increase. However, we observe that DMSGD-II performs worse with an increasing number of agents. Further, the lesser the number of agents, the convergence is faster(e.g. convergence is fastest with just 2 agents), which is intuitive. Although DMSGD-II has better convergence characteristics, the consensus characteristics are worse because of which, as the number of agents increase, the performance deteriorates. This phenomenon is parallel with the theory, as explained above in the main paper and supplementary material.

As seen in the above experiments, choice is $\beta$ is very important to see good performance in the results. We run some experiments on different values of $\beta$ to understand the effect of $\beta$ on the consensus. For these and below experiments, we study only the adaptive $\omega$ in order to see a more generalized impact whereas, there might be a specific impact of $\beta$ on classic momentum and momentum of consensus variables. The results obtained are presented in Figure 8. We also run some experiments on different agent interaction matrices to understand its effect on consensus and convergence properties. The agent interaction matrices used are chosen such that they are doubly stochastic and the second-largest eigenvalue ($\lambda$) are well separated to understand the difference between the experiments. The results are similar to trends shown in (Jiang et al., 2017).

As seen in the above experiments, there are heavily non-linear dynamics involved with the choice of $\beta$, $\omega$, and the corresponding performance. The choice is $\beta$ is very important to see an excellent performance in the results. We run some experiments on different values of $\beta$ and $\omega$ under different data simulation strategies and datasets. Figures 17-20 are a result of several such studies. One common point to be observed among all these results is that as $\beta$ is close to 1.0, and as $\omega$ is close to 1.0, we observe that the algorithm fails severely. This observation validates the theoretical analysis we presented earlier. Further, we also observe that the adaptive $\omega$ is performing either equivalently to other $\omega$ values, or in some cases, it performs better than other $\omega$ values. This justifies the need for such an operator and further helps us in reducing one additional hyper-parameter.

In conclusion, we explore extensively on the parameters in the DMSGD algorithm with both the options. We observe merits for both the algorithms in different data simulation strategies and choice of topologies and number of agents. We see that DMSGD algorithm performs significantly better than other algorithms.
Momentum-Accelerated Consensus

Figure 4. Visualization of loss surface on hyperplane as explained in the main text for mnist data on a Logistic Regression(LR) Model

(a) MNIST dataset with LR model using training metrics (b) MNIST dataset with LR model using test metrics

Figure 5. Visualization of loss surface on hyperplane as explained in the main text for semeion data on a Logistic Regression(LR) Model
Figure 6. Performance of our proposed algorithm, DMSGD-(I & II) with different $\omega$ values and momentum $\beta = 0.95$, and its comparison with CDSGD. These experiments are done on mnist dataset using non-iid data simulation strategy and a sufficiently large CNN model.

Figure 7. Performance of our proposed algorithm, DMSGD-(I & II) with different $\omega$ values and momentum $\beta = 0.95$, and its comparison with CDSGD. These experiments are done on mnist dataset using non-iid data simulation strategy and a LR model.

Figure 8. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value and momentum $\beta = 0.95$, on different topologies. These experiments are done on mnist dataset using non-iid data simulation strategy and a LR model.
Momentum-Accelerated Consensus

Figure 9. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value and momentum $\beta = 0.95$, on different topologies. These experiments are done on mnist dataset using non-iid data simulation strategy and a sufficiently large CNN model.

Figure 10. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value and momentum $\beta = 0.95$, on different topologies. These experiments are done on cifar10 dataset using iid data simulation strategy and a sufficiently large CNN model.

Figure 11. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value and momentum $\beta = 0.95$, on different topologies. These experiments are done on cifar10 dataset using non-iid data simulation strategy and a sufficiently large CNN model.
Figure 12. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value and momentum $\beta = 0.95$, with various number of agents. These experiments are done on semeion dataset using iid data simulation strategy and a LR model.

Figure 13. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with $\omega \to 1$, momentum $\beta = 0.95$, on various number of agents. These experiments are done on mnist dataset using iid data simulation strategy and a LR model.

Figure 14. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value, momentum $\beta = 0.95$, on various number of agents. These experiments are done on mnist dataset using iid data simulation strategy and a LR model.
Figure 15. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value, momentum $\beta = 0.95$, on various number of agents. These experiments are done on mnist dataset using iid data simulation strategy and a sufficiently large CNN model.

Figure 16. Performance of our proposed algorithm, DMSGD-I (left) & DMSGD-II (right) with adaptive $\omega$ value, momentum $\beta = 0.95$, on various number of agents. These experiments are done on cifar10 dataset using iid data simulation strategy and a sufficiently large CNN model.

Figure 17. Performance of our proposed algorithm, DMSGD-I & II with various $\omega$ and momentum values, on a fully-connected topology. These experiments are done on mnist dataset using iid data simulation strategy and a sufficiently large CNN model.
Figure 18. Performance of our proposed algorithm, DMSGD-(I & II) with various $\omega$ and momentum values, on a fully-connected topology. These experiments are done on cifar10 dataset using iid data simulation strategy and a sufficiently large CNN model.

Figure 19. Performance of our proposed algorithm, DMSGD-(I & II) with various $\omega$ and momentum values, on a ring topology. These experiments are done on cifar10 dataset using iid data simulation strategy and a sufficiently large CNN model.

Figure 20. Performance of our proposed algorithm, DMSGD-(I & II) with various $\omega$ and momentum values, on a bipartite topology. These experiments are done on cifar10 dataset using iid data simulation strategy and a sufficiently large CNN model.
Figure 21. CNN model used in experiments.