frontiers
in Artificial Intelligence

ORIGINAL RESEARCH
published: 14 September 2021
doi: 10.3389/frai.2021.573731

On Consensus-Optimality Trade-offs in Collaborative Deep Learning

Zhanhong Jiang[1]*, *Aditya Balu*[1], *Chinmay Hegde*[2] *and Soumik Sarkar*[1]

[1]*Self-aware Complex Systems Lab, Department of Mechaical Engineering, Iowa State University, Ames, IA, Unitd States,*
[2]*Tandon School of Engineering, New York University, New York, NY, United States*

In distributed machine learning, where agents collaboratively learn from diverse private data sets, there is a fundamental tension between *consensus* and *optimality*. In this paper, we build on recent algorithmic progresses in distributed deep learning to explore various consensus-optimality trade-offs over a fixed communication topology. First, we propose the *incremental consensus*-based distributed stochastic gradient descent (i-CDSGD) algorithm, which involves multiple consensus steps (where each agent communicates information with its neighbors) within each SGD iteration. Second, we propose the *generalized consensus*-based distributed SGD (g-CDSGD) algorithm that enables us to navigate the full spectrum from complete consensus (all agents agree) to complete disagreement (each agent converges to individual model parameters). We analytically establish convergence of the proposed algorithms for strongly convex and nonconvex objective functions; we also analyze the momentum variants of the algorithms for the strongly convex case. We support our algorithms via numerical experiments, and demonstrate significant improvements over existing methods for collaborative deep learning.

Keywords: distributed optimization, consensus-optimality, collaborative deep learning, sgd, convergence

OPEN ACCESS

Edited by:
Fabrizio Riguzzi,
University of Ferrara, Italy

Reviewed by:
Arnaud Fadja Nguembang,
University of Ferrara, Italy
Claudio Gallicchio,
University of Pisa, Italy

****Correspondence:***
Zhanhong Jiang
starkjiang@gmail.com

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence,
a section of the journal Frontiers in Artificial Intelligence

Received: *18 June 2020*
Accepted: *19 August 2021*
Published: *14 September 2021*

Citation:
Jiang Z, Balu A, Hegde C and Sarkar S (2021) On Consensus-Optimality Trade-offs in Collaborative Deep Learning.
Front. Artif. Intell. 4:573731.
doi: 10.3389/frai.2021.573731

1 INTRODUCTION

1.1 Motivation

Scaling up deep learning algorithms in a distributed setting (Recht et al., 2011; LeCun et al., 2015; Jin et al., 2016) is becoming increasingly critical, impacting several applications such as learning in robotic networks (Lenz et al., 2015; Fang et al., 2019), the Internet of Things (IoT) (Gubbi et al., 2013; Lane et al., 2015; Hu et al., 2020), mobile device networks (Lane and Georgiev, 2015; Kang et al., 2020), and sensor networks (Ge et al., 2019; He et al., 2020). For instance, with the development of wireless communication and distributed computing technologies, intelligent sensor network has been emerging as a kind of large-scale distributed network systems, which request more advanced sensor fusion techniques that enable data privacy preservation (Jiang et al., 2017a; He et al., 2019), dynamic optimization (Yang et al., 2016), and intelligent learning (Tan, 2020). This paper aims at developing novel algorithms to facilitate collaborative deep learning in distributed settings such as distributed sensor networks (Lesser et al., 2012). Several distributed deep learning approaches have been proposed to address issues such as model parallelism (Dean et al., 2012), data parallelism (Dean et al., 2012; Jiang et al., 2017a), and the role of communication and computation (Li et al., 2014; Das et al., 2016).

We focus on the constrained communication topology setting where the data is distributed (so that each agent has its own estimate of the deep model) and where information exchange among the learning agents are constrained along the edges of a given communication graph (Jiang et al., 2017a; Lian et al., 2017). In this context, two key aspects arise: *consensus* and *optimality*. We refer the reader to **Figure 1** for an illustration involving three agents. With sufficient information exchange, the learned model parameters corresponding to each agent, $\theta_k^j, j = 1, 2, 3$ could converge to $\hat{\theta}$, in which case they achieve consensus but not optimality (here, θ_* is the optimal model estimate if all the data were centralized). On the other hand, if no communication happens, the agents may approach their individual model estimates (θ_*^i) while being far from consensus. The question is whether this trade-off between consensus and optimality can be balanced so that *all* agents collectively agree upon a model estimate close to θ_*.

1.2 Our Contributions

In this paper, we propose, analyze, and empirically evaluate two new algorithmic frameworks for distributed deep learning that enable us to explore fundamental trade-offs between consensus and optimality. The first approach is called *incremental consensus*-based distributed stochastic gradient descent (i-CDSGD), which is a stochastic extension of the descent-style algorithm proposed in (Berahas et al., 2018). This involves running multiple consensus steps where each agent exchanges information with its neighbors within each SGD iteration. The second approach is called *generalized consensus*-based distributed SGD (g-CDSGD), based on the concept of generalized gossip (Jiang et al., 2017b). This involves a tuning parameter that explicitly controls the trade-off between consensus and optimality. Specifically, we:

- (**Algorithmic**) propose the i-CDSGD and g-CDSGD algorithms (along with their momentum variants).

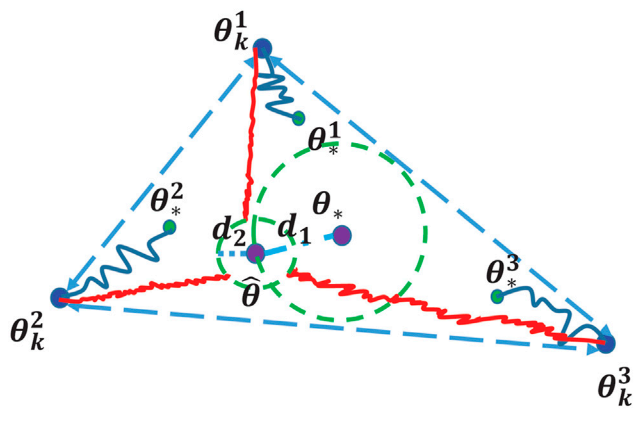

FIGURE 1 | A closer look at the optimization updates in distributed deep learning: Blue dots represent the current states (i.e., learned model parameters) of the agents; green dots represent the individual local optima (θ_*^i), that agents converge to without sufficient consensus; the purple dot (θ_*) represents the ideal optimal point for the entire agent population; another purple dot $\hat{\theta}$ represents a possible consensus point for the agents which is far from optimal; blue and red curves signify the convergence trajectories with different step sizes; the green dashed circles indicate the neighborhoods of θ_* and $\hat{\theta}$, respectively; d_2 represents the consensus bound/error and d_1 represents the optimality bound/error; ideally, both of these bounds should be small.

- (**Theoretical**) prove the convergence of g-CDSGD (Theorems 1 and 3) and i-CDSGD (Theorems 2 and 4) for strongly convex and non-convex objective functions;
- (**Theoretical**) prove the convergence of the momentum variants of g-CDSGD (Theorem 5) and i-CDSGD (Theorem 6) for strongly convex objective functions;
- (**Practical**) empirically demonstrate that i-CDMSGD (the momentum variant of i-CDSGD) can achieve similar (global) accuracy as the state-of-the-art with lower fluctuation across epochs as well as better consensus;
- (**Practical**) empirically demonstrate that g-CDMSGD (the momentum variant of g-CDSGD) can achieve similar (global) accuracy as the state-of-the-art with lower fluctuation, smaller generalization error and better consensus.

We use both balanced and unbalanced datasets (i.e., equal or unequal distributions of training samples among the agents) for the numerical experiments with benchmark deep learning data sets. Please see **Table 1** for a detailed comparison with existing algorithms.

1.3 Related Work

A large literature has emerged that studies distributed deep learning in both centralized and decentralized settings (Dean et al., 2012; Zhang et al., 2015; Blot et al., 2016; Jin et al., 2016; McMahan et al., 2016; Xu et al., 2017; Zhang et al., 2017; Zheng et al., 2017; Esfandiari et al., 2021), and we only attempt to summarize the most recent work (Wangni et al., 2017). proposed a gradient sparsification approach for communication-efficient distributed learning, while (Wen et al., 2017) proposed the concept of ternary gradients to reduce communication costs (Scaman et al., 2017). proposed a multi-step dual accelerated method using a gossip protocol to provide an optimal decentralized optimization algorithm for smooth and strongly convex loss functions. Decentralized parallel stochastic gradient descent (Lian et al., 2017) has also been proposed. In (Duchi et al., 2012), the authors developed a distributed averaging method for convex (possibly nonsmooth) objective functions; additionally (Mokhtari and Ribeiro, 2016), proposed a decentralized double stochastic averaging gradient algorithm. However, non-convex functions were not taken into account in either of the above works. Dual approaches (Uribe et al., 2020; Dvinskikh et al., 2019) were also proposed to address the convergence issues in the distributed optimization over networks while extra parameters need to be updated for obtaining the optimal solutions, which in return could increase the difficulty of solving the problem and the computational complexity. Again, convex problems were the main focus that might not enable the proposed schemes to generalize well for non-convex problems. Another category of approaches, the primal-dual gradient algorithms developed in (Hong et al., 2018; Dvinskikh and Gasnikov, 2019) were not evaluated by real-world datasets and were only originally specific for homogeneous networks where data was assumed independently identically distributed (i.i.d.).

Perhaps most closely related to this paper is the work of (Berahas et al., 2018), who presented a distributed optimization method (called DGD^τ) to enable consensus when the cost of

TABLE 1 | Comparisons between different optimization approaches.

Method	*F*	Con.Bou	Opt.Bou	Con.Rate	Mom.Ana	CC.T.	Sto
FedAvg McMahan et al. (2016)	Nonconvex	N/A	N/A	N/A	No	No	Yes
DGD$^{\tau}$ Berahas et al. (2018)	Str-con	$\mathcal{O}(\frac{\alpha}{1-\lambda_2^{\tau}})$	$\mathcal{O}(\frac{\alpha}{1-\lambda_2^{\tau}})$	$\mathcal{O}(\epsilon^k)$	No	Yes	No
MSDA Scaman et al. (2017)	Str-con	N/A	N/A	$\mathcal{O}(\epsilon^k)$	Yes	Yes	No
	—	—	—	—	—	—	—
	—	—	—	—	—	—	—
CDSGD Jiang et al. (2017a)	Str-con	$\mathcal{O}(\frac{\alpha}{1-\lambda_2})$	$\mathcal{O}(\frac{\alpha\gamma+1}{H+\alpha^{-1}(1-\lambda_2)})$	$\mathcal{O}(\epsilon^k)$	No	Yes	Yes
	Nonconvex	$\mathcal{O}(\frac{\alpha}{1-\lambda_2})$	$\mathcal{O}(\alpha\gamma+1-\lambda_N)$	N/A			
	—	—	—	—	—	—	—
	—	—	—	—	—	—	—
Acc-DNGD-SC Qu and Li (2017)	Str-con	$\mathcal{O}(\frac{\alpha^{\frac{1}{3}}}{(1-\lambda_2)\lambda_2^{\frac{2}{3}}})$	N/A	$\mathcal{O}(\epsilon^k)$	Yes	Yes	No
	—	—	—	—	—	—	—
	—	—	—	—	—	—	—
i-CDSGD [This paper]	Str-con	$\mathcal{O}(\frac{\alpha}{1-\lambda_2^{\tau}})$	$\mathcal{O}(\frac{\alpha\gamma+1}{H+\alpha^{-1}(1-\lambda_2^{\tau})})$	$\mathcal{O}(\epsilon^k)$	Yes	Yes	Yes
	Nonconvex	$\mathcal{O}(\frac{\alpha}{1-\lambda_2^{\tau}})$	Theorem 4	$\mathcal{O}(k^{-1})$	—	—	—
	—	—	—	—	—	—	—
	—	—	—	—	—	—	—
g-CDSGD [This paper]	Str-con	$\mathcal{O}(\frac{\omega\alpha}{1-\lambda_2})$	$\mathcal{O}(\frac{\alpha\gamma-1+\omega^{-1}}{H})$	$\mathcal{O}(\epsilon^k)$	Yes	Yes	Yes
	Nonconvex	$\mathcal{O}(\frac{\omega\alpha}{1-\lambda_2})$	Theorem 3	$\mathcal{O}(k^{-1})$	—	—	—

Con.Bou.: consensus bound. Opt.Bou.: optimality bound. Con.Rate: convergence rate. Str-con: strongly convex. Mom.Ana.: momentum analysis. α: step size. $\lambda_2 \in (0, 1)$: the second largest eigen-value of a stochastic matrix. $\tau \in \mathbb{N}$: positive constant. $\omega \in (0, 1]$: a positive constant. $\epsilon \in (0, 1)$: a positive constant, and it signifies the representative meaning. They are not exactly the same in different methods. Sto.: stochastic. C.C.T.: constrained communication topology. c_1, $c_2 > 0$: condition numbers. H: strong convexity constant. $\gamma > 0$: smoothness constant. The optimality bounds for i-CDSGD and g-CDSGD with nonconvex functions refer to the constant error bounds in Theorems 4 and 3.

communication is cheap. However, the authors only considered convex optimization problems, and only study deterministic gradient updates. Also (Qu and Li, 2017), proposed a class of (deterministic) accelerated distributed Nesterov gradient descent methods to achieve linear convergence rate, for the special case of strongly convex objective functions. In (Tsianos and Rabbat, 2012), both deterministic and stochastic distributed were discussed while the algorithm had no acceleration techniques. To our knowledge, none of these previous works have explicitly studied the trade-off between consensus and optimality. It should also be noted that the proposed approaches guarantee the convergence to the first-order stationary points for non-convex analysis and the avoidance of local maxima and saddle points is out of scope.

Outline: **Section 2** presents the problem and several mathematical preliminaries. In **Section 3**, we present our two algorithmic frameworks, along with their analysis in **Section 4**. For validating the proposed schemes, several experimental results based on benchmark datasets are presented in **Section 5**. Concluding remarks are presented in **Section 6**.

2 PROBLEM FORMULATION

We consider the standard unconstrained empirical risk minimization (ERM) problem typically used in machine learning problems (such as deep learning):

$$\min \frac{1}{n}\sum_{i=1}^{n} f^i(\theta), \qquad (1)$$

where $\theta \in \mathbb{R}^d$ denotes the parameter vector of interest, $f: \mathbb{R}^d \to \mathbb{R}$ denotes a given loss function, and f^i is the function value corresponding to a data point i. Our focus is to investigate the case where the ERM problem is solved collaboratively among a number of computational *agents*. In this paper, we are interested in problems where the agents exhibit *data parallelism*, i.e., they only have access to their own respective training datasets. However, we assume that the agents can communicate over a static undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a vertex set (with nodes corresponding to agents) and $\mathcal{E}$ is an edge set. Throughout this paper we assume that the graph $\mathcal{G}$ is *connected*.

In this work, we primarily consider the spectrum between consensus and optimality and investigate thoroughly what effect such trade-offs have on the decentralized learning paradigm. Specifically, we analyze the theoretical properties of the proposed algorithms and show the empirical findings over benchmark datasets. However, in realistic scenarios, the graph may be subject to changes, such as the addition of new agents, and robust decentralized learning algorithms need to be developed for tackling such an issue, which is out of the scope and will definitely be one of our future research directions beyond this work.

Let $\mathcal{D}_j$, $j = 1, \ldots, n$ denote the subset of the training data (comprising n_j samples) corresponding to the j^{th} agent such that $\sum_{j=1}^{N} n_j = n$, where N is the total number of agents. With this formulation, and since $f(\theta) = \sum_{j=1}^{N} f_j(\theta)$, we have the following (constrained) reformulation of **Eq. 1**:

$$\min \sum_{j=1}^{N}\sum_{i\in\mathcal{D}_j} f_j^i(\theta^j), \text{ s.t. } \theta^j = \theta^l \quad \forall (j,l) \in \mathcal{E}, \qquad (2)$$

Note that in this context, θ^j for all $j = 1, 2, \ldots, N$ is the local copy of θ, which means the model architecture for each agent is typically the same. In another line of works where agents own different models, personalized federated/decentralized learning (Fallah et al., 2020) or meta-learning approaches (Fallah et al., 2021) have been developed correspondingly.

Equivalently, the concatenated form of the above equation is as follows:

$$\min \mathcal{F}(\Theta) \coloneqq \sum_{j=1}^{N} \sum_{i \in \mathcal{D}_j} f_j^i(\theta^j), \text{ s.t. } (\Pi \otimes I_d)\Theta = \Theta, \tag{3}$$

where $\Theta \coloneqq [\theta^1; \theta^2; \ldots; \theta^N]^T \in \mathbb{R}^{dN}$, $\Pi \in \mathbb{R}^{N \times N}$ is the agent interaction matrix with its entries π_{jl} indicating the link between agents j and l, I_d is the identity matrix of dimension $d \times d$, and $\otimes$ represents the Kronecker product. Each element value in Π signifies the connection probability between two agents such that $\pi_{jl} \in [0, 1]$. One assumption is imposed for Π in the sequel to show the properties of any graph associated with the networked system.

We now introduce several key definitions and assumptions that characterize the above problem.

Definition 1. *A function $f: \mathbb{R}^d \to \mathbb{R}$ is said to be H-strongly convex, if for all $x, y \in \mathbb{R}^d$, we have $f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{H}{2}\|y - x\|^2$. It is said to be γ-smooth if $f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{\gamma}{2}\|y - x\|^2$. Here, $\| \cdot \|$ represents the Euclidean norm.*

Definition 2. *A function c 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 γ_j-smooth; b) each f_j is proper (not everywhere infinite) and coercive; c) each f_j is Lipschitz continuous.*

Assumption 2. *The interaction matrix Π is normalized to be doubly stochastic; the second largest eigenvalue of Π is strictly less than 1, i.e., $\lambda_2(\Pi)<1$, where $\lambda_2(\Pi)$ is the second largest eigenvalue of Π. If $(j, l) \notin \mathcal{E}$, then $\pi_{jl} = 0$.*For convenience, we use λ_2 to represent $\lambda_2(\Pi)$ and similar λ_N for $\lambda_N(\Pi)$, which signifies the N-largest eigenvalue of Π.

We will solve **Eq. 2** in a distributed and stochastic manner.

For solving stochastic optimization problems, variants of the well-known stochastic gradient descent (SGD) have been commonly employed. For the formulation in **Eq. 2**, one of the state-of-the-art algorithms is a method called *consensus distributed SGD*, or CDSGD, recently proposed in (Jiang et al., 2017a). This method estimates θ according to the update equation:

$$\theta_{k+1}^j = \sum_{l \in Nb(j)} \pi_{jl}\theta_k^l - \alpha g_j(\theta_k^j) \tag{4}$$

where $Nb(j)$ indicates the neighborhood of agent j, α is the step size, $g_j(\theta_k^j)$ is the (stochastic) gradient of f_j at θ_k^j, implemented by drawing a minibatch of sampled data points. More precisely, $g_j(\theta_k^j) = \frac{1}{b'}\sum_{q' \in \mathcal{D}'} \nabla f_j^{q'}(\theta_k^j)$, where b' is the size of the minibatch $\mathcal{D}'$ selected uniformly at random from the data subset $\mathcal{D}_j$ available to Agent j.

3 PROPOSED ALGORITHMS

State-of-the-art algorithms such as CDSGD alternate between the *gradient update* and *consensus* steps. We propose two natural extensions where one can control the emphasis on *consensus* relative to the *gradient update* and hence, leads to interesting trade-offs between consensus and optimality.

3.1 Increasing Consensus

Observe that the concatenated form of the CDSGD updates, **Eq. 4**, can be expressed as

$$\Theta_{k+1} = (\Pi \otimes I_d)\Theta_k - \alpha \mathbf{g}(\Theta_k),$$

If we perform τ consensus steps interlaced with each gradient update, we can obtain the following concatenated form of the iterations of the parameter estimates:

$$\Theta_{k+1} = (\Pi^\tau \otimes I_d)\Theta_k - \alpha \mathbf{g}(\Theta_k) \tag{5}$$

where, $\mathbf{g}(\Theta_k) = [g_1^T(\theta_k^1), g_2^T(\theta_k^2), \ldots, g_N^T(\theta_k^N)]^T$. We call this variant *incremental* consensus-based distributed SGD (i-CDSGD) which is detailed in **Algorithm 1**. Note, in a distributed setting, that this algorithm incurs an additional factor τ in communication complexity.

In this context, i-CDSGD not only extends traditional decentralized (stochastic) gradient algorithms but also leverages the consensus among agents in a graph, particularly when agents are heterogeneous. Most traditional decentralized algorithms have been properly designed for homogeneous scenarios where agents share common properties such data sampling distributions and cannot be directly applied to heterogeneous networks as only one step of consensus may not be enough to enable agents to converge to the same solution due to the trade-off between the consensus-optimality. Therefore, the analysis presented in the paper for i-CDSGD provides a new perspective different from that most traditional decentralized algorithms delivered. A *different* and more direct approach to control the trade-off between consensus and gradient would be as follows:

$$\Theta_{k+1} = (1 - \omega)(\Pi \otimes I_d)\Theta_k + \omega(\Theta_k - \alpha \mathbf{g}(\Theta_k)) \tag{6}$$

where, $0<\omega \leq 1$ is a user-defined parameter. We call this algorithm *generalized* consensus-based distributed SGD (g-CDSGD), and the full procedure is detailed in **Algorithm 3**.

Algorithm 1 Incremental Consensus-based Distributed Stochastic Gradient Descent

```
Initialization: $\theta_0^j, v_0^j$, $j = 1, 2, \ldots, N$, $\alpha$, $N$, $\tau$, $m$, $\Pi$
Distribute the training data set to $N$ agents
for each agent do
    andomly shuffle each data subset
    for $k = 0$: $m$ do
        $t = 0$
        for $j = 1, \ldots, N$ do
            $\theta_t^j = \theta_k^j$ {Initialization before incremental consensus}
        end for
        while $t \leq \tau - 1$ do
            for $j = 1, \ldots, N$ do
                $\theta_{t+1}^j = \sum_{l \in Nb(j)} \pi_{jl}\theta_t^l$ {Incremental consensus}
            end for
            $t = t + 1$
        end while
        $\hat{\theta} = \theta^j_t$ {Update after incremental consensus}
        $\theta^j_{k+1} = \hat{\theta} - \alpha g_j(\theta^j_k)$ {Update for iteration}
    end for
end for
```

Algorithm 2 Incremental Consensus-based Distributed Stochastic Gradient Descent w/ Momentum

```
Initialization: $\theta^j_0, v^j_0,\ j = 1, 2, \ldots, N,\ \alpha,\ N,\ \tau,\ m,\ \Pi,\ \mu$
Distribute the Non-IID training data set to $N$ agents
for each agent do
    Randomly shuffle each data subset
    for $k = 0: m$ do
        $t = 0$
        for $j = 1, \ldots, N$ do
            $\theta^j_t = \theta^j_k$ {Initialization before incremental consensus}
            $v^j_t = v^j_k$ {Initialization of momentum before incremental consensus}
        end for
        while $t \leq \tau - 1$ do
            for $j = 1, \ldots, N$ do
                $\theta^j_{t+1} = \sum_{l \in Nb(j)} \pi_{jl} \theta^l_t$ {Incremental consensus for decision variable}
                $v^j_{t+1} = \sum_{l \in Nb(j)} \pi_{jl} v^l_t$ {Incremental consensus for momentum}
            end for
            $t = t + 1$
        end while
        $\hat{\theta} = \theta^j_t$ {Update of decision variable after incremental consensus}
        $\hat{v} = v^j_t$ {Update of momentum after incremental consensus}
        $v^j_{k+1} = \hat{\theta} - \theta^j_k + \mu \hat{v} - \alpha g_j(\theta^j_k)$ {Update of momentum for iteration}
        $\theta^j_{k+1} = \theta^j_k + v^j_{k+1}$ {Update of decision variable for iteration}
    end for
end for
```

By examining **Eq. 6**, we observe that when ω approaches 0, the update law boils down to a only consensus protocol, and that when ω approaches 1, the method reduces to standard stochastic gradient descent (for individual agents).

Next, we introduce the *Nesterov momentum* variants of our aforementioned algorithms. The momentum term is typically used for speeding up the convergence rate with high momentum constant close to **Eq. 1** (Sutskever et al., 2013). More details can be found in **Algorithms 2** and **4**.

Algorithm 3 Generalized Consensus-based Distributed Stochastic Gradient Descent

```
Initialization: $\omega,\ \theta^j_0, v^j_0,\ j = 1, 2, \ldots, N,\ \alpha,\ N,\ m,\ \Pi$
Distribute the training data set to $N$ agents
for each agent do
    Randomly shuffle each data subset
    for $k = 0: m$ do
        $\hat{\theta} = \sum_{l \in Nb(j)} \pi_{jl} \theta^l_k$ {Consensus update for decision variable only}
        $\theta^j_{k+1} = (1 - \omega)\hat{\theta} + \omega(\theta^j_k - \alpha g_j(\theta^j_k))$ {Generalized consensus}
    end for
end for
```

We provide a discussion on the trade-off between the consensus and optimality to conclude this section. The trade-off between consensus and optimality can vary from convex to non-convex optimization problems. For most convex distributed optimization problems, they are well defined and globally optimal solution are not empty (probably unique if strongly convex) so each agent can communicate with other agents in its neighborhood to reach consensus and their local gradient updates will guide them to an minimizer. Therefore, the trade-off can be perfectly balanced to get to good optimal solutions. However, for non-convex problems, there exist possibly numerous locally optimal solutions such that the trade-off plays a critical role in the distributed optimization. The consensus among agents may not be necessarily a good optimal solution since the local gradient update of an agent may "dominate" the solution searching process and allows for "bias". Hence, the investigation of such a trade-off is quite critical.

3.2 Tools for Convergence Analysis

We now analyze the convergence of the iterates $\{\theta^j_k\}$ generated by our algorithms. Specifically, we identify an appropriate Lyapunov function (that is bounded from below) for each algorithm that decreases with each iteration, thereby establishing convergence.

Algorithm 4 Generalized Consensus-based Distributed Stochastic Gradient Descent w/ Momentum

```
Initialization: $\omega,\ \theta^j_0, v^j_0,\ j = 1, 2, \ldots, N,\ \alpha,\ N,\ m,\ \Pi,\ \mu$
Distribute the Non-IID training data set to $N$ agents
for each agent do
    Randomly shuffle each data subset
    for $k = 0: m$ do
        $\hat{\theta} = \sum_{l \in Nb(j)} \pi_{jl} \theta^l_t$ {Consensus update for decision variable}
        $\hat{v} = \sum_{l \in Nb(j)} \pi_{jl} v^l_k$ {Consensus update for momentum}
        $v^j_{k+1} = (1 - \omega)(\hat{\theta} - \theta^j_k + \mu \hat{v}) + \omega \mu v^j_k - \omega \alpha g_j(\theta^j_k + \mu v^j_k)$ {Generalized consensus for momentum}
        $\theta^j_{k+1} = \theta^j_k + v^j_{k+1}$ {Update of decision variable for iteration}
    end for
end for
```

In our analysis, we use the concatenated (Kronecker) form of the updates. For simplicity, let $\mathbf{P} = \Pi \otimes I_d \in \mathbb{R}^{Nd \times Nd}$.

We begin the analysis for g-CDSGD by constructing a Lyapunov function that combines the true objective function with a regularization term involving a quadratic form of consensus as follows:

$$V(\Theta) := \omega \mathcal{F}(\Theta) + \frac{1 - \omega}{2\alpha} \Theta^T (I_{Nd} - \mathbf{P}) \Theta \tag{7}$$

It is easy to show that $\sum_{j=1}^N f_j(\theta^j)$ is $\gamma_m := \max_j \{\gamma^j\}$-smooth, and that $V(\Theta)$ is $\hat{\gamma}$-smooth with

$$\hat{\gamma} := \omega\gamma_m + (1-\omega)\alpha^{-1}\lambda_{\max}(I_{Nd} - \mathbf{P}) = \omega\gamma_m + (1-\omega)\alpha^{-1}(1-\lambda_N).$$

Likewise, it is easy to show that $\sum_{j=1}^{N} f_j(\theta^j)$ is $H_m := \min_j\{H_j\}$-strongly convex; therefore $V(\Theta)$ is $\hat{H}$-strongly convex with

$$\hat{H} := \omega H_m + (1-\omega)(2\alpha)^{-1}\lambda_{\min}(I_{Nd} - \mathbf{P}) = \omega H_m + (1-\omega)(2\alpha)^{-1}(1-\lambda_2).$$

We also assume that there exists a lower bound $V_{\inf}$ for the function value sequence $\{V(\Theta_k)\}$, $\forall k$. When the objective functions are strongly convex, we have $V_{\inf} = V(\Theta^*)$, where Θ^* is the optimizer. Due to Assumptions 1 and 2, it is straightforward to obtain an equivalence between the gradient of **Eq. 7** and the update law of g-CDSGD. Rewriting (6), we get:

$$\Theta_{k+1} = (1-\omega)\mathbf{P}\Theta_k + \omega(\Theta_k - \alpha\mathbf{g}(\Theta_k)) \tag{8}$$

Therefore, we obtain:

$$\begin{aligned}\Theta_{k+1} &= \Theta_k - \Theta_k + (1-\omega)\mathbf{P}\Theta_k + \omega(\Theta_k - \alpha\mathbf{g}(\Theta_k))\\ &= \Theta_k - \alpha\omega\Theta_k - (1-\omega)I_{Nd}\Theta_k + (1-\omega)\mathbf{P}\Theta_k\\ &= \Theta_k - \alpha\underbrace{\left(\omega\mathbf{g}(\Theta_k) + \frac{1}{\alpha}(1-\omega)(I_{Nd} - \mathbf{P})\Theta_k\right)}_{\text{Lyapunov Gradient}}\end{aligned} \tag{9}$$

The last term in **Eq. 9** is precisely the gradient of $V(\Theta)$. In the stochastic setting, $g(\Theta_k)$ can be approximated by sampling one data point (or a mini-batch of data points) and the stochastic Lyapunov gradient is denoted by $\mathcal{S}(\Theta_k), \forall k$.

Similarly, the update laws for our proposed Nesterov momentum variants can be compactly analyzed using the above Lyapunov function. First, we rewrite the updates for g-CDMSGD as follows:

$$\mathbf{y}_{k+1} = \Theta_k + \mu(\Theta_k - \Theta_{k-1}) \tag{10a}$$

$$\Theta_{k+1} = (1-\omega)\mathbf{P}\mathbf{y}_{k+1} + \omega(\mathbf{y}_{k+1} - \alpha\mathbf{g}(\mathbf{y}_{k+1})) \tag{10b}$$

With a few algebraic manipulations, we get:

$$\begin{aligned}\Theta_{k+1} &= \mathbf{y}_{k+1} - \mathbf{y}_{k+1} + (1-\omega)\mathbf{P}\mathbf{y}_{k+1} + \omega(\mathbf{y}_{k+1} - \alpha\mathbf{g}(\mathbf{y}_{k+1}))\\ &= \mathbf{y}_{k+1} - \alpha\left(\omega\mathbf{g}(\mathbf{y}_{k+1}) + \frac{1-\omega}{\alpha}(I_{Nd} - \mathbf{P})\mathbf{y}_{k+1}\right)\end{aligned} \tag{11}$$

The above derivation simplifies the Nesterov momentum-based updates into a regular form which is more convenient for convergence analysis. For clarity, we separate this into two sub-equations. Let $\mathcal{S}(\mathbf{y}_{k+1}) = \omega\mathbf{g}(\mathbf{y}_{k+1}) + \frac{1-\omega}{\alpha}(I_{Nd} - \mathbf{P})\mathbf{y}_{k+1}$. Thus, the updates for g-CDMSGD can be expressed as

$$\mathbf{y}_{k+1} = \Theta_k + \mu(\Theta_k - \Theta_{k-1}) \tag{12a}$$

$$\Theta_{k+1} = \mathbf{y}_{k+1} - \alpha S(\mathbf{y}_{k+1}), \tag{12b}$$

Please find the similar transformation for i-CDMSGD in **Supplementary Section S1**.

For analysis, we require a bound on the variance of the stochastic Lyapunov gradient $\mathcal{S}(\Theta_k)$ such that the variance of the gradient noise[1] can be bounded from above. The variance of $\mathcal{S}(\Theta_k)$ is defined as:

$$Var[\mathcal{S}(\Theta_k)] := \mathbb{E}[\|\mathcal{S}(\Theta_k)\|^2] - \|\mathbb{E}[\mathcal{S}(\Theta_k)]\|^2.$$

The following assumption is standard in SGD convergence analysis, and is based on Bottou et al. (2018).

Assumption 3. *a) There exist scalars* $r_2 \geq r_1 > 0$ *such that* $\nabla V(\Theta_k)^T\mathbb{E}[\mathcal{S}(\Theta_k)] \geq r_1\|\nabla V(\Theta_k)\|^2$ *and* $\|\mathbb{E}[\mathcal{S}(\Theta_k)]\| \leq r_2\|\nabla V(\Theta_k)\|$ *for all* $k \in \mathbb{N}$*; b) There exist scalars* $B \geq 0$ *and* $B_V \geq 0$ *such that* $Var[\mathcal{S}(\Theta_k)] \leq B + B_V\|\nabla V(\Theta_k)\|^2$ *for all* $k \in \mathbb{N}$.

Remark 1. Assumption 3(a) guarantees sufficient descent of V in the direction of $-\mathcal{S}(\Theta_k)$; Assumption 3(b) states that the variance of $\mathcal{S}(\Theta_k)$ is bounded above by the second moment of $\nabla V(\Theta_k)$. The constant B can be regarded to represent the second moment of noise involving in the gradient $\mathcal{S}(\Theta_k)$. Therefore, the second moment of $\mathcal{S}(\Theta_k)$ can be bounded above as

$$\mathbb{E}[\|\mathcal{S}(\Theta_k)\|^2] \leq B + B_m\|\nabla V(\Theta_k)\|^2,$$

where $B_m := B_V + r_2^2 \geq r_1^2 \geq 0$. Note that this is slightly different from the conventional assumption in SGD analysis that the variance of stochastic gradients is bounded above by a single constant; in our context, we control the restriction of $\mathcal{S}(\Theta_k)$ via two scalar constants. However, our analysis technique is otherwise similar.

For convergence analysis, we assume:

Assumption 4. *There exists a constant* $G > 0$ *such that* $\|\nabla V(x)\| \leq G, \forall x \in \mathbb{R}^{dN}$.

We justify this assumption. As the Lyapunov function is a composite function with the true cost function which is Lipschitz continuous and the regularization term associated with consensus, it can be immediately obtained that $\|\nabla V(x)\|$ is bounded above by some positive constant.

Before turning to our main results, we present two auxiliary technical lemmas.

Lemma 1. *Let Assumptions* 1 *and* 2 *hold. The iterates of g-CDSGD (Algorithm* 3*) satisfy the following inequality* $\forall k \in \mathbb{N}$:

$$\mathbb{E}[V(\Theta_{k+1})] - V(\Theta_k) \leq -\alpha\nabla V(\Theta_k)^T\mathbb{E}[\mathcal{S}(\Theta_k)] + \frac{\hat{\gamma}}{2}\alpha^2\mathbb{E}[\|\mathcal{S}(\Theta_k)\|^2]. \tag{13}$$

Lemma 2. *Let Assumptions* 1, 2, *and* 3 *hold. The iterates of g-CDSGD (Algorithm* 3*) satisfy the following inequality* $\forall k \in \mathbb{N}$:

$$\mathbb{E}[V(\Theta_{k+1})] - V(\Theta_k) \leq -\left(r_1 - \frac{\hat{\gamma}}{2}\alpha B_m)\alpha\|\nabla V(\Theta_k\right)\|^2 + \frac{\hat{\gamma}}{2}\alpha^2 B. \tag{14}$$

We provide the proof of Lemmas 1 and 2 in the **Supplementary Section S1**. To guarantee that the first term on the right hand side is strictly negative, the step size α should be chosen such that

[1] As our proposed algorithm is a distributed variant of SGD, the noise in the performance is caused by the random sampling Song et al. (2015).

$$0 \le \alpha \le \frac{r_1 - (1-\omega)(1-\lambda_N^\tau)B_m}{\omega B_m \gamma_m}. \tag{15}$$

4 ANALYSIS AND MAIN RESULTS

This section presents the main results by analyzing the convergence properties of the proposed algorithms. Our main results are grouped as follows: 1) we provide rigorous convergence analysis for g-CDSGD and i-CDSGD for both strongly convex and non-convex objective functions. 2) we analyze their momentum variants only for strongly convex objective functions. It is noted that the proofs of theorems are provided in the main body while the proofs of lemmas and propositions are provided in the **Supplementary Section S1**.

4.1 Convergence Analysis for I-CDSGD and G-CDSGD

Our analysis will consist of two components: establishing an upper bound on how far away the estimates of the individual agents are with respect to their empirical mean (which we call the *consensus bound*), and establishing an upper bound on how far away the overall procedure is with respect to the optimum (which we call the *optimality bound*).

First, we obtain consensus bounds for the g-CDSGD and i-CDSGD as follows.

Proposition 1. *(Consensus with fixed step size, g-CDSGD) Let Assumptions* 1, 2, 4 *hold. The iterates of g-CDSGD (Algorithm* 3*) satisfy the following inequality* $\forall k \in \mathbb{N}$, *when* α *satisfies* **Eq. 15**,

$$\mathbb{E}[\|\theta_k^j - s_k\|] \le \frac{\omega\alpha\sqrt{B + B_m G^2}}{1 - \hat{\lambda}_2} \tag{16}$$

where $s_k = \frac{1}{N}\sum_{j=1}^N \theta_k^j$, $\hat{\lambda}_2$ *is the second-largest eigenvalue of the matrix* $\mathbf{Q} = (1-\omega)\mathbf{P} + \omega I_{Nd}$.

Proposition 2. *(Consensus with fixed step size, i-CDSGD) Let Assumptions* 1, 2, 4 *hold. The iterates of i-CDSGD (Algorithm* 1*) satisfy the following inequality* $\forall k \in \mathbb{N}$, *when* α *satisfies* $0 \le \alpha \le \frac{r_1 - (1-\lambda_N^\tau)B_m}{\gamma_m B_m}$,

$$\mathbb{E}[\|\theta_k^j - s_k\|] \le \frac{\alpha\sqrt{B + B_m G^2}}{1 - \lambda_2^\tau} \tag{17}$$

We provide a discussion on comparing the consensus bounds in the **Supplementary Section S1**. Next, we obtain optimality bounds for g-CDSGD and i-CDSGD.

Theorem 1. (Convergence of g-CDSGD in strongly convex case) Let Assumptions 1, 2, and 3 hold. When the step size satisfies **Eq. 15**, the iterates of g-CDSGD (Algorithm 3) satisfy the following inequality $\forall k \in \mathbb{N}$:

$$\mathbb{E}[D_k] \le C_1^{k-1} D_1 + C_2 \sum_{q=0}^{k-1} C_1^q \tag{18}$$

where $D_k = V(\Theta_k) - V^*$, $C_1 = 1 - (\omega\alpha H_m + \frac{1-\omega}{2}(1-\lambda_2))r_1$, $C_2 = \frac{(\alpha^2\gamma_m\omega + \alpha(1-\omega)(1-\lambda_N))B}{2}$.

Proof. Recalling Lemma 2 and using Definition 1 yield:

$$\mathbb{E}\quad[V(\Theta_{k+1})] - V(\Theta_k) \le -\left(r_1 - \frac{\hat{\gamma}}{2}\alpha B_m\right)\alpha\|\nabla V(\Theta_k)\|^2 + \frac{\hat{\gamma}}{2}\alpha^2 B \le -\frac{1}{2}\alpha r_1\|\nabla(\Theta_k)\|^2 + \frac{\alpha^2\hat{\gamma}B}{2} \le -\alpha r_1 \hat{H}(V(\Theta_k) - V^*) + \frac{\alpha^2\hat{\gamma}B}{2}. \tag{19}$$

The second inequality follows from that $\alpha \le \frac{r_1}{\hat{\gamma}B_m}$, which is implied by **Eq. 15**. The expectation taken in the above inequalities is only related to θ_{k+1}. Hence, recursively taking the expectation and subtracting V^* from both sides, we get:

$$\mathbb{E}[V(\Theta_{k+1}) - V^*] \le (1 - \alpha\hat{H}r_1)\mathbb{E}[V(\Theta_k) - V^*] + \frac{\alpha^2\hat{\gamma}B}{2}. \tag{20}$$

As $0 \le \alpha\hat{H}r_1 \le \frac{\hat{H}r_1^2}{\hat{\gamma}B_m} \le \frac{\hat{H}r_1^2}{\hat{\gamma}r_1^2} = \frac{\hat{H}}{\hat{\gamma}} \le 1$, the conclusion follows by applying **Eq. 1** recursively through iteration $k \in \mathbb{N}$ and letting $D_k = V(\Theta_k) - V^*, C_1 = 1 - (\omega\alpha H_m + \frac{1-\omega}{2}(1-\lambda_2))r_1, C_2 = (\alpha^2\gamma_m\omega + \alpha\,\frac{(1-\omega)(1-\lambda_N))B}{2}$.

Theorem 2. (Convergence of i-CDSGD in strongly convex case) Let Assumptions 1, 2, and 3 hold. When the step size satisfies $0 \le \alpha \le \frac{r_1 - (1-\lambda_N^\tau)B_m}{\gamma_m B_m}$, the iterates of i-CDSGD (Algorithm 1) satisfy the following inequality $\forall k \in \mathbb{N}$:

$$\mathbb{E}[D_k] \le C_3^{k-1} D_1 + C_4 \sum_{q=0}^{k-1} C_3^q \tag{21}$$

where $D_k = V(\Theta_k) - V^\star$, $C_3 = 1 - (\alpha H_m + \frac{1}{2}(1-\lambda_2^\tau))r_1$, $C_4 = \frac{(\alpha^2\gamma_m + \alpha(1-\lambda_N^\tau))B}{2}$.

Proof. We omit the proof here and one can easily get it following the proof techniques shown for Theorem 1. The desired result is obtained by replacing C_1 with C_3 and C_2 with C_4, respectively.

Although we show the convergence for strongly convex objectives, we note that objective functions are highly non-convex for most deep learning applications. While convergence to a global minimum in such cases is extremely difficult to establish, we prove that g-CDSGD and i-CDSGD still exhibits weaker (but meaningful) notions of convergence.

Theorem 3. *(Convergence to the first-order stationary point for non-convex case of g-CDSGD) Let Assumptions* 1, 2, *and* 3 *hold. When the step size satisfies* **Eq. 15**, *the iterates of g-CDSGD (Algorithm* 3*) satisfy the following inequality* $\forall K \in \mathbb{N}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^K \|\nabla V(\Theta_k)\|^2\right] \le \frac{(\omega\gamma_m\alpha + (1-\omega)(1-\lambda_N))B}{r_1} + \frac{2(V(\Theta_1) - V_{\inf})}{Kr_1\alpha} \tag{22}$$

Proof. Recalling Lemma 2 and taking expectations on both sides lead to the following relation:

$$\mathbb{E}[V(\Theta_{k+1})] - \mathbb{E}[V(\Theta_k)] \le -\left(r_1 - \frac{\hat{\gamma}\alpha B_m}{2}\right)\alpha\mathbb{E}[\|\nabla V(\Theta_k)\|^2] + \frac{\hat{\gamma}\alpha^2 B}{2}. \tag{23}$$

If the step size is such that $\alpha \le \frac{r_1}{\hat{\gamma}B_m}$, we get:

$$\mathbb{E}[V(\Theta_{k+1})] - \mathbb{E}[V(\Theta_k)] \le -\frac{r_1\alpha}{2}\mathbb{E}[\|\nabla V(\Theta_k)\|^2] + \frac{\alpha^2\hat{\gamma}B}{2}. \quad (24)$$

Applying the above inequality from 1 to K and summing them up can give the following relation

$$V_{\inf} - V(\Theta_1) \le \mathbb{E}[V(\Theta_{k+1})] - V(\Theta_1) \le -\frac{r_1\alpha}{2}\sum_{k=1}^{m}\mathbb{E}[\|\nabla V(\Theta_k)\|^2] + \frac{m\alpha^2\hat{\gamma}B}{2}. \quad (25)$$

The last inequality follows from the Assumption 3. Rearrangement of the above inequality, substituting $\hat{\gamma} = \omega\gamma_m + \alpha^{-1}(1-\omega)(1-\lambda_N)$, and dividing by K yields the desired result.

Theorem 4. *(Convergence to the first-order stationary point for non-convex case of i-CDSGD) Let Assumptions* 1, 2, *and* 3 *hold. When the step size satisfies* $0 \le \alpha \le \frac{r_1 - (1-\lambda_N^\tau)B_m}{\gamma_m B_m}$, *the iterates of i-CDSGD (Algorithm* 1*) satisfy the following inequality* $\forall K \in \mathbb{N}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla V(\Theta_k)\|^2\right] \le \frac{(\gamma_m\alpha + (1-\lambda_N^\tau))B}{r_1} + \frac{2(V(\Theta_1) - V_{\inf})}{Kr_1\alpha}. \quad (26)$$

Proof. The proof for this theorem is rather similar to the one provided for Theorem 3 above, and we omit the details. □

Remark 2. In the literature, to eliminate the negative effect of "noise" caused by the stochastic gradients, a diminishing step size is used. However, in our context, we observe from Theorems 1 and 2 that a constant step size itself can result in convergence, up to a neighborhood, of the local minimum. In fact, using a constant step size can lead to a linear convergence rate instead of a sub-linear convergence rate. To summarize, one can speed up the convergence rate at the cost of solution accuracy, which has also been reported in the recent work Pu and Nedić (2018).

In our context, we have shown the convergence of the function value sequence $\{V(\Theta_k)\}$ to a neighborhood of $V^\star$. As $\mathbb{E}[\omega\mathcal{F}(\Theta_k)] \le \mathbb{E}[V(\Theta_k)]$, then we have $\mathbb{E}[\mathcal{F}(\Theta_k)] \le \frac{1}{\omega}\mathbb{E}[V(\Theta_k)]$. Since $\nabla V(\Theta^*) = \nabla\mathcal{F}(\Theta^*) = 0$, which leads to that $\mathcal{F}^* = V^*$. It can be obtained that $\mathbb{E}[\mathcal{F}(\Theta_k) - \mathcal{F}^*] \le \frac{1}{\omega}\mathbb{E}[V(\Theta_k) - V^*]$. Then using Theorems 1 and 2 can establish analogous convergence rate for the true value function sequence $\{\mathcal{F}(\Theta_k)\}$.

Remark 3. Let us discuss the rates of convergence suggested by Theorems 1 and 3. We observe that when the objective function is strongly convex, the function value sequence $\{V(\Theta_k)\}$ can *linearly* converge to within a fixed radius of convergence, which can be calculated as follows:

$$\lim_{k\to\infty}\mathbb{E}[V(\Theta_k) - V^\star] \le \frac{\mathrm{B}[\omega\alpha\gamma_\mathrm{m} + (1-\omega)(1-\lambda_\mathrm{N})]}{2\mathrm{r}_1(\omega\mathrm{H}_\mathrm{m} + \alpha^{-1}(1-\omega)(1-\lambda_2))}.$$

When the objective function is non-convex, we cannot claim linear convergence. However, Theorem 3 asserts that the average of the second moment of the Lyapunov gradient is bounded from above. Recall that the parameter B bounds the variance of the "noise" due to the stochasticity of the gradient, and if $B = 0$, Theorem 3 implies that $\{\Theta_k\}$ asymptotically converges to a first-order stationary point.

Remark 4. For g-CDSGD, let us investigate the corner cases where $\omega \to 0$ or $\omega \to 1$. For the strongly convex case, when $\omega \to 1$, we have $\frac{\alpha cB}{2r_1}$, where $c = \frac{\gamma_m}{H_m}$ is the condition number. This suggests that if consensus is not a concern, then each iterate $\{\theta_k^j\}$ converges to its own respective $\theta_\star^j$, as depicted in **Figure 1**. On the other hand, when $\omega \to 0$, the upper bound converges to $\frac{\alpha B(1-\lambda_N)}{2r_1(1-\lambda_2)}$. In such a scenario, each agent sufficiently communicates its own information with other agents to arrive at an agreement. In this case, the upper bound depends on the topology of the communication network. If $\lambda_N \approx 0$, this results in:

$$\lim_{k\to\infty}\mathbb{E}[V(\Theta_k) - V^*] \le \frac{B\alpha}{2r_1(1-\lambda_2)}.$$

For the non-convex case, when $\omega \to 1$, the upper bound suggested by Theorem 3 is $\frac{\alpha\gamma_m B}{r_1}$, while $\omega \to 0$ leads to $\frac{(1-\lambda_N)B}{r_1}$, which is roughly $\frac{B}{r_1}$ if $\lambda_N \approx 0$.
We also compare i-CDSGD and CDSGD with g-CDSGD in terms of the optimality upper bounds to arrive at a suitable lower bound for ω. However, due to the space limit, the analysis is presented in the **Supplementary Section S1**.

4.2 Convergence Analysis for Momentum Variants

We next provide a convergence analysis for the g-CDMSGD algorithm, summarized in the update laws given in **Eq. 12**. A similar analysis can be applied to i-CDMSGD. The proof techniques are developed on top of the estimate sequence method that has been applied to the centralized version (Nesterov, 2013). In the following analysis, we focus on the basic variant where $\mu = \frac{1-\sqrt{\hat{H}\alpha}}{1+\sqrt{\hat{H}\alpha}}$. Before stating the main result, we define the sequence $\phi_k(\Theta)$, $k = 1, 2, \ldots$ as:

$$\phi_1(\Theta) = V(\Theta_1) + \frac{\hat{H}}{2}\|\Theta - \Theta_1\|^2, \text{ and}$$

$$\phi_{k+1} = (1-\sqrt{\hat{H}\alpha})\phi_k(\Theta) + \sqrt{\hat{H}\alpha}(\hat{V}(\mathbf{y}_k) + (\mathcal{S}_k, \Theta - \mathbf{y}_k) + \frac{\hat{H}}{2}\|\Theta - \mathbf{y}_k\|^2 \quad (27)$$

where $\hat{V}$ represents the average of the objective function values of a mini-batch. We define ϕ_k^* as follows

$$\phi_k^* = \min_{\Theta\in\mathbb{R}^{Nd}}\phi_k(\Theta)$$

Further, from Assumption 3, we see that $Var[\mathcal{S}(\mathbf{y}_k)] \le B + B_V\|\nabla V(\mathbf{y}_k)\|^2$. Combining Assumption 4 and $\mathrm{Var}[\mathcal{S}(\mathbf{y}_k)] := \mathbb{E}[\|\mathcal{S}(\mathbf{y}_k) - \nabla V(\mathbf{y}_k)\|^2]$, we have $\mathbb{E}[\|\mathcal{S}(\mathbf{y}_k) - \nabla V(\mathbf{y}_k)\|^2] \le B + B_V G^2$.

We now state our main result, which characterizes the performance of g-CDMSGD. To our knowledge, this is the first theoretical result for momentum-based versions of consensus-distributed SGD.

Theorem 5. *(Convergence of g-CDMSGD, strongly convex case) Let Assumptions* 1, 2, 3, *and* 4 *hold. If the step size satisfies* $\alpha \le \min\{\frac{r_1-(1-\omega)(1-\lambda_N)B_m}{\omega B_m\gamma_m}, \frac{1}{\hat{H}}, \frac{1}{2\hat{\gamma}}\}$, *we have:*

$$\mathbb{E}[V(\Theta_k) - V^*] \leq (1 - \sqrt{\hat{H}\alpha})^{k-1}(\phi_1^* - V^*) + \sqrt{\frac{\alpha}{\hat{H}}}(B + B_V G^2). \tag{28}$$

PROOF. From Lemma 5 in **Supplementary Section S1.**, it can be obtained that

$$\mathbb{E}[V(\Theta_k)] \leq \mathbb{E}\left[\phi_k^* + \sum_{p=1}^{k-1}(1 - \sqrt{\hat{H}\alpha}\,)^{k-1-p}\{\alpha\|\mathcal{S}(\mathbf{y}_k) - \nabla V(\mathbf{y}_k)\|^2\}\right] \tag{29}$$

The last inequality follows from that the coefficient $-\frac{\hat{H}}{2}\frac{1-\sqrt{\hat{H}\alpha}}{\sqrt{\hat{H}\alpha}} \leq 0$. Recalling **Eq. S20** of Lemma 5 in **Supplementary Section S1** and letting $\Theta = \Theta^*$, and combining **Eq. 29**, we have

$$\begin{aligned}\mathbb{E}[V(\Theta_k)] &\leq \mathbb{E}[V^* + (1 - \sqrt{\hat{H}\alpha}\,)^{k-1}(\phi_1^* - V^*)] \\ &+ \mathbb{E}\left[\sum_{p=1}^{k-1}(1 - \sqrt{\hat{H}\alpha}\,)^{k-1-p}\{\alpha\|\mathcal{S}(\mathbf{y}_k) - \nabla V(\mathbf{y}_k)\|^2\}\right]\end{aligned} \tag{30}$$

As $\mathbb{E}[\|\mathcal{S}(\mathbf{y}_k) - \nabla V(\mathbf{y}_k)\|^2] \leq B + B_V G^2$, therefore, the following inequality can be acquired

$$\begin{aligned}\mathbb{E}[V(\Theta_k) - V^*] &\leq (1 - \sqrt{\hat{H}\alpha}\,)^{k-1}(\phi_1^* - V^*) \\ &+ \mathbb{E}\left[\sum_{p=1}^{k-1}(1 - \sqrt{\hat{H}\alpha}\,)^{k-1-p}(B + B_V G^2)\right]\end{aligned} \tag{31}$$

Using $\sum_{p=1}^{k-1}(1 - \sqrt{\hat{H}\alpha}\,)^{k-1-p} \leq \sum_{t=0}^{\infty}(1 - \sqrt{\hat{H}\alpha}\,)^t = \frac{1}{\sqrt{\hat{H}\alpha}}$ completes the proof.

Theorem 6. *(Convergence of i-CDMSGD, strongly convex case) Let Assumptions* 1, 2, 3, *and* 4 *hold. If the step size satisfies* $\alpha \leq \min\{\frac{r_1 - (1-\lambda_N^\tau)B_m}{B_m\gamma_m}, \frac{1}{\hat{H}}, \frac{1}{2\hat{\gamma}}\}$, *we have:*

$$\mathbb{E}[V(\Theta_k) - V^*] \leq (1 - \sqrt{\hat{H}\alpha})^{k-1}(\phi_1^* - V^*) + \sqrt{\frac{\alpha}{\hat{H}}}(B + B_V G^2). \tag{32}$$

Note, although the theorem statements look the same for g-CDMSGD and i-CDMSGD, the constants $\hat{H}$ are significantly different from each other. Theorem 5 suggests that with a sufficiently small step size, using Nesterov acceleration results in a linear convergence (with parameter $1 - \sqrt{\hat{H}\alpha}$) up to a neighborhood of V^* of radius $\sqrt{\frac{\alpha}{\hat{H}}}(B + B_V G^2)$. When $k \to \infty$, the first term on the right hand side vanishes and substituting $\hat{H} = \omega H_m + (1-\omega)(2\alpha)^{-1}(1-\lambda_2)$ into $\sqrt{\frac{\alpha}{\hat{H}}}(B + B_V G^2)$, we have

$$\sqrt{\frac{\alpha}{\omega H_m + (1-\omega)(2\alpha)^{-1}(1-\lambda_2)}}(B + B_V G^2),$$

which implies that the upper bound is related to the spectral gap $1 - \lambda_2$ of the network; hence, a similar conclusion as Theorem 1 can be deduced. When $\omega \to 0$, the upper bound becomes $\alpha\sqrt{\frac{1}{2(1-\lambda)}}(B + B_V G^2)$. However, $\omega \to 1$ leads to $\sqrt{\frac{\alpha}{H_m}}(B + B_V G^2)$. These two scenarios demonstrates that the "gradient noise" cased by the stochastic sampling negatively affects the convergence. One can use ω to trade-off the consensus and optimality updates. Evidently, when compared with the non-momentum version, the upper bound is looser due to the $B_V G^2$ even when $B = 0$. However, it should be noted that B+ $B_V G^2$ represents the upper bound of variance of $\mathcal{S}(\Theta_k)$. The analysis below shows the faster convergence rate with the cost solution accuracy.

Next, we discuss the upper bounds obtained when $k \to \infty$ for g-CDSGD and g-CDMSGD. 1) $\omega \to 0$: When B_V is sufficiently small and $r_1 \approx \frac{1}{2\sqrt{2}}$, it can be observed that the optimality bound for the Nesterov momentum variant is smaller than that for g-CDSGD as $\frac{B\alpha}{1-\lambda_2} \geq \frac{B\alpha}{\sqrt{1-\lambda_2}}$; 2) $\omega \to 1$: When γ_m and r_1 are carefully selected such that $\frac{\gamma_m}{2r_1} \approx 1$, we have $B\sqrt{\frac{\alpha}{H_m}} \leq \frac{B\alpha}{H_m}$ when $\frac{\alpha}{H_m} > 1$. Therefore, introducing the momentum can speed up the convergence rate with appropriately chosen hyperparameters.

5 EXPERIMENTAL RESULTS

We validate our algorithms with several experimental results using the CIFAR-10 image recognition dataset (with standard training and testing sets). We have performed the experiments with different network architectures and hyperparameters. Out of several offline hyperparameters chosen, for brevity, we present results obtained with the LeNet architecture (LeCun et al., 1998). However, we note that, the behavior for other networks remain the same. Please see Supplement for results on other hyperparameters. The LeNet architecture is a convolutional neural network (CNN) (with ReLU activations) which includes 2 convolutional layers with 32 filters each followed by a max pooling layer, then 2 more convolutional layers with 64 filters each followed by another max pooling layer, and a dense layer with 512 units. The mini-batch size is set to 512, and step size is set to 0.01 in all experiments. All experiments were performed using Keras with TensorFlow (Chollet, 2015; Abadi et al., 2016).

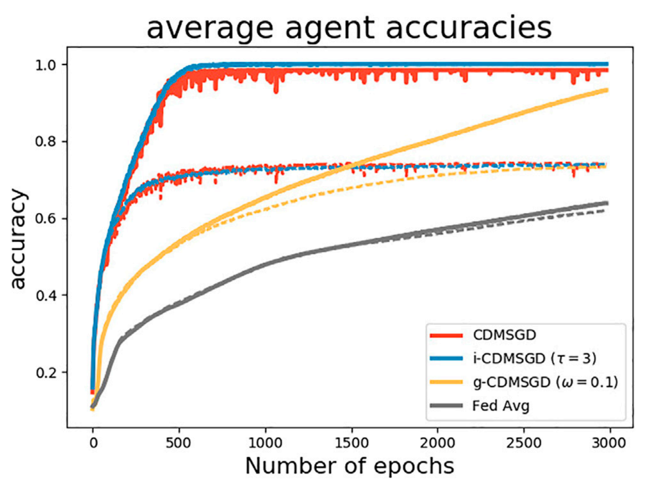

FIGURE 2 | Performance of different algorithms with unbalanced sample distribution among agents (Dashed lines represent test accuracy and solid lines represent training accuracy.)

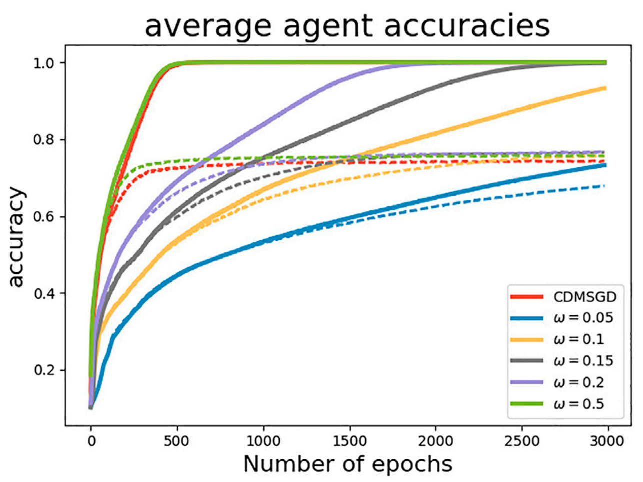

FIGURE 3 | Performance of g-CDMSGD for different ω values (Dashed lines represent test accuracy and solid lines represent training accuracy.)

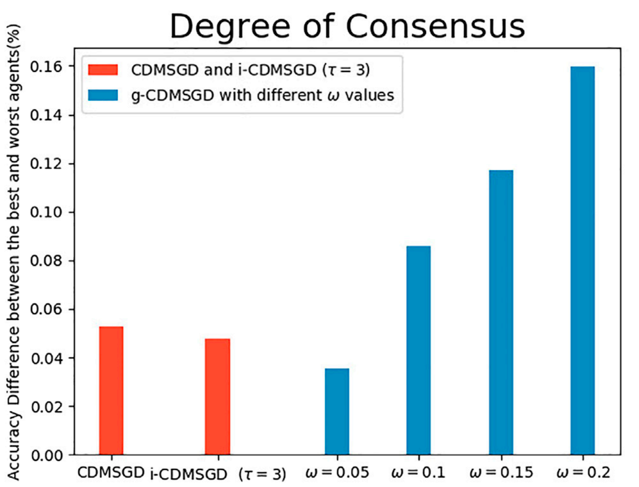

FIGURE 5 | The accuracy percentage difference between the best and the worst agents with balanced sample distribution for CDMSGD, i-CDMSGD and g-CDMSGD (varying ω.).

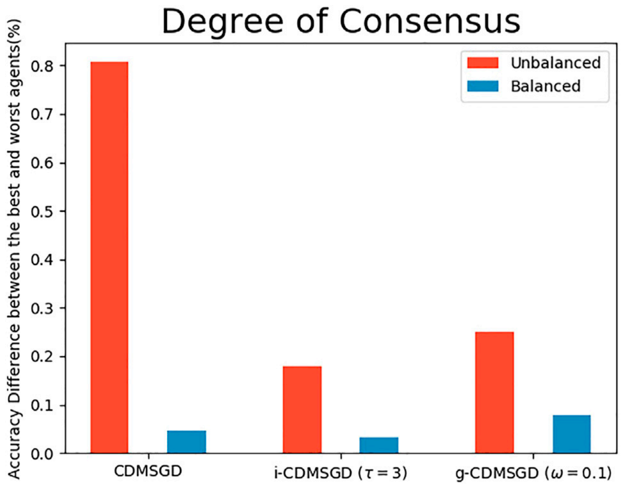

FIGURE 4 | The accuracy percentage difference between the best and the worst agents for different algorithms with unbalanced and balanced sample distribution among agents.

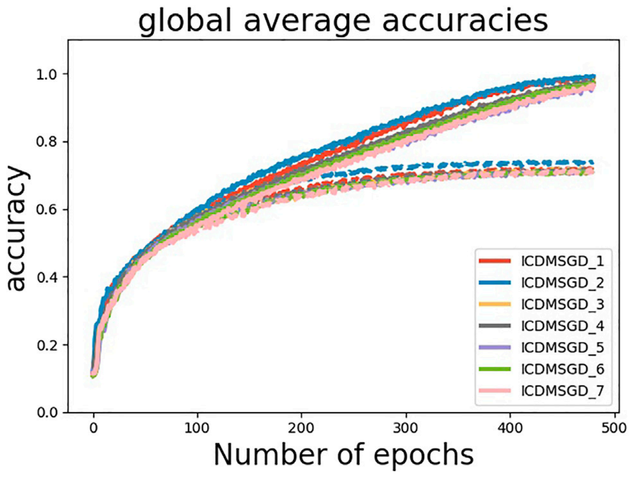

FIGURE 6 | Performance of i-CDMSGD for different τ values (Dash lines represent test accuracy and solid lines represent training accuracy).

We use a sparse network topology with five agents. We use both balanced and unbalanced data sets for our experiments. In the balanced case, agents have an equal share of the entire training set. However, in the unbalanced case, agents have (randomly selected) unequal parts of the training set while making sure that each agent has at least half of the equal share amount of examples. We summarize our key experimental results in this section, with more details and results provided in the **Supplementary Section 2**.

Performance of algorithms. In **Figure 2**, we compare the performance of the momentum variants of our proposed algorithms, i-CDMSGD and g-CDMSGD (with ω = 0.1) with state-of-the art techniques such as CDMSGD and Federated Averaging using an unbalanced data set. All algorithms were run for 3,000 epochs. Observing the average accuracy over all the agents for both training and test data, we note that i-CDMSGD can converge as fast as CDMSGD with lesser fluctuation in the performance across epochs. While being slower in convergence, g-CDMSGD achieves similar performance (with test data) with less fluctuation as well as smaller generalization gap (i.e., difference between training and testing accuracy). All algorithms significantly outperform Federated Averaging in terms of average accuracy. We also vary the tuning parameter ω for g-CDMSGD to show (in **Figure 3**) that it is able to achieve similar (or better) convergence rate as CDMSGD using higher ω values with some sacrifice in terms of the generalization gap.

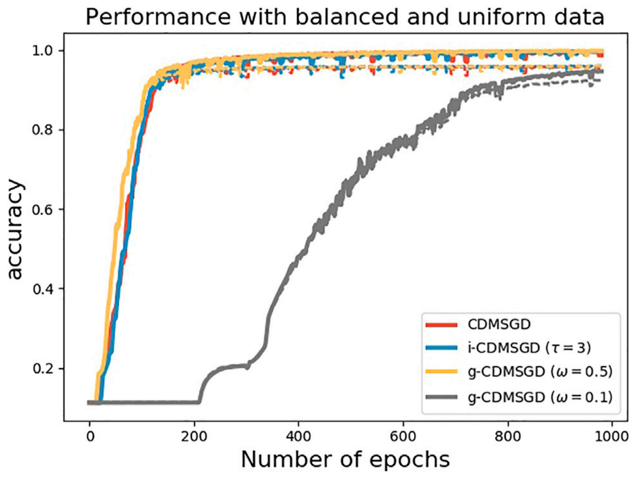

FIGURE 7 | Performance of different algorithms with a balanced data distribution on MNIST dataset.

Degree of Consensus. One of the main contribution of our paper is to show that one can control the degree of consensus while maintaining average accuracy in distributed deep learning. We demonstrate this by observing the accuracy difference between the best and the worst performing agents (identified by computing the mean accuracy for the last 100 epochs). As shown in **Figure 4**, the degree of consensus is similar for all three algorithms for balanced data set, with i-CDMSGD performing slightly better than the rest. However, for an unbalanced set, both i-CDMSGD and g-CDMSGD perform significantly better compared to CDMSGD. Note, the degree of consensus can be further improved for g-CDMSGD using lower values of ω as shown in **Figure 5**. However, the convergence becomes relatively slower as shown in **Figure 3**. We do not compare these results with the Federated Averaging algorithm as it performs a brute force consensus at every epoch using centralized parameter server. In **Figure 6**, we show the performance of i-CDMSGD with different τ values. We observe that while there is better performance by increasing the value of τ, we see that the performance degrades after a while and then quickly stabilizes to similar performance. This is because the doubly stochastic agent interaction matrix for the small agent population becomes stationary very quickly with a very small value of τ. However, this will be explored in our future work with significantly bigger networks.

Finally, we also compare our proposed algorithms to CDMSGD on another benchmark dataset—MNIST. The performance of the algorithms is shown in **Figure 7** which follows similar trend as observed for CIFAR-10.

6 CONCLUSION AND FUTURE WORK

For investigating the trade-off between consensus and optimality in distributed deep learning with constrained communication topology, this paper presents two new algorithms, called i-CDSGD and g-CDSGD and their momentum variants. We show the convergence properties for the proposed algorithms and the relationships between the hyperparameters and the consensus and optimality bounds. Theoretical and experimental comparison with the state-of-the-art algorithm called CDSGD, shows that i-CDSGD, and g-CDSGD can improve the degree of consensus among the agents while maintaining the average accuracy especially when there is data imbalance among the agents. Future research directions include learning with non-uniform data distributions among agents and time-varying networks.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here MNIST: http://yann.lecun.com/exdb/mnist/, CIFAR: https://www.cs.toronto.edu/~kriz/cifar.html.

AUTHOR CONTRIBUTIONS

ZJ: Writing the whole paper, give the analytical results, and help implement the experiments AB: Help write the paper, mainly implement the experiments and present the numerical results, help finalize the paper CH: Check the writing, analytical and experimental results, and help finalize the paper SS: Check the writing, analytical and experimental results, and help finalize the paper.

FUNDING

The funder information is: National Science Foundation grant CNS-1954556; National Science Foundation under the CAREER grant CNS-1845969; National Science Foundation under the CAREER grant CCF-1750920/2005804.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2021.573731/full#supplementary-material

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "Tensorflow: A System for Large-Scale Machine Learning," in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI-16), Savannah, GA, November 2–4, 2016, 265–283.

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

Blot, M., Picard, D., Cord, M., and Thome, N. (2016). *Gossip Training for Deep Learning*. Barcelona, Spain: arXiv. preprint arXiv:1611.09726.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization Methods for Large-Scale Machine Learning. *Siam Review* 60(2), 223–311.

Chollet, F. (2015). Keras. [Dataset].

Das, D., Avancha, S., Mudigere, D., Vaidynathan, K., Sridharan, S., Kalamkar, D., et al. (2016). *Distributed Deep Learning Using Synchronous Stochastic Gradient Descent*. arXiv. preprint arXiv:1602.06709.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., et al. (2012). "Large Scale Distributed Deep Networks," in Advances in neural information processing systems, Lake Tahoe, NV, December 3–8, 2012, 1223–1231.

Duchi, J. C., Agarwal, A., and Wainwright, M. J. (2012). Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Trans. Automat. Contr.* 57, 592–606. doi:10.1109/tac.2011.2161027

Dvinskikh, D., and Gasnikov, A. (2021). Decentralized and Parallelized Primal and Dual Accelerated Methods for Stochastic Convex Programming Problems. *Journal of Inverse and Ill-posed Problems* 29 (3), 385–405.

Dvinskikh, D., Gorbunov, E., Gasnikov, A., Dvurechensky, P., and Uribe, C. A. (2019). *On Dual Approach for Distributed Stochastic Convex Optimization over Networks*. Nice, France: arXiv. preprint arXiv:1903.09844.

Esfandiari, Y., Tan, S. Y., Jiang, Z., Balu, A., Herron, E., Hegde, C., et al. (2021). "Cross-Gradient Aggregation for Decentralized Learning from Non-IID Data," in Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research 139, 3036–3046.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2021). *Generalization of Model-Agnostic Meta-Learning Algorithms: Recurring and Unseen Tasks*. arXiv. preprint arXiv:2102.03832.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2020). *Personalized federated learning: A meta-learning approach. in Advances in Neural Information Processing Systems*. Virtual-only Conference, December 6–12, 2020.

Fang, X., Pang, D., Xi, J., and Le, X. (2019). Distributed Optimization for the Multi-Robot System Using a Neurodynamic Approach. *Neurocomputing* 367, 103–113. doi:10.1016/j.neucom.2019.08.032

Ge, X., Han, Q.-L., Zhang, X.-M., Ding, L., and Yang, F. (2019). Distributed Event-Triggered Estimation over Sensor Networks: A Survey. *IEEE Trans. cybernetics*.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (Iot): A Vision, Architectural Elements, and Future Directions. *Future Gen. Comput. Syst.* 29, 1645–1660. doi:10.1016/j.future.2013.01.010

He, J., Cai, L., Cheng, P., Pan, J., and Shi, L. (2019). Consensus-based Data-Privacy Preserving Data Aggregation. *IEEE Trans. Automat. Contr.* 64, 5222–5229. doi:10.1109/tac.2019.2910171

He, S., Shin, H.-S., Xu, S., and Tsourdos, A. (2020). Distributed Estimation over a Low-Cost Sensor Network: A Review of State-Of-The-Art. *Inf. Fusion* 54, 21–43. doi:10.1016/j.inffus.2019.06.026

Hong, M., Lee, J. D., and Razaviyayn, M. (2018). *Gradient primal-dual algorithm converges to second-order stationary solution for nonconvex distributed optimization over networks*. International Conference on Machine Learning (pp. 2009-2018). PMLR. preprint arXiv:1802.08941.

Hu, R., Guo, Y., Ratazzi, E. P., and Gong, Y. (2020). *Differentially Private Federated Learning for Resource-Constrained Internet of Things*. arXiv. preprint arXiv:2003.12705.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. (2017a). "Collaborative Deep Learning in Fixed Topology Networks," in Neural Information Processing Systems (NIPS), Long Beach, CA, December 4–9, 2017.

Jiang, Z., Mukherjee, K., and Sarkar, S. (2017b). Generalised Gossip-Based Subgradient Method for Distributed Optimisation. *Int. J. Control.*, 1–17. doi:10.1080/00207179.2017.1387288

Jin, P. H., Yuan, Q., Iandola, F., and Keutzer, K. (2016). *How to Scale Distributed Deep Learning?* arXiv. preprint arXiv:1611.04581.

Kang, J., Xiong, Z., Niyato, D., Zou, Y., Zhang, Y., and Guizani, M. (2020). Reliable Federated Learning for mobile Networks. *IEEE Wireless Commun.* 27, 72–80. doi:10.1109/mwc.001.1900119

Lane, N. D., Bhattacharya, S., Georgiev, P., Forlivesi, C., and Kawsar, F. (2015). "An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-Of-Things Devices," in Proceedings of the 2015 International Workshop on Internet of Things towards Applications (ACM), Seoul, South Korea, November 1, 2015, 7–12. doi:10.1145/2820975.2820980

Lane, N. D., and Georgiev, P. (2015). "Can Deep Learning Revolutionize mobile Sensing?," in Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (ACM), Santa Fe, New Mexico, February 12–13, 2015, 117–122.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature* 521, 436–444. doi:10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* 86, 2278–2324. doi:10.1109/5.726791

Lenz, I., Lee, H., and Saxena, A. (2015). Deep Learning for Detecting Robotic Grasps. *Int. J. Robot. Res.* 34, 705–724. doi:10.1177/0278364914549607

Lesser, V., Ortiz, C. L., Jr, and Tambe, M. (2012). *Distributed Sensor Networks: A Multiagent Perspective*, 9. Springer Science & Business Media.

Li, M., Andersen, D. G., Smola, A. J., and Yu, K. (2014). "Communication Efficient Distributed Machine Learning with the Parameter Server," in Advances in Neural Information Processing Systems, Montreal, Canada, December 8–13, 2014, 19–27.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. (2017). "Can Decentralized Algorithms Outperform Centralized Algorithms? a Case Study for Decentralized Parallel Stochastic Gradient Descent," in Advances in Neural Information Processing Systems, Long Beach, CA, December 4–9, 2017, 5336–5346.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-efficient Learning of Deep Networks from Decentralized Data. *In Artificial intelligence and statistics*, 1273–1282.

Mokhtari, A., and Ribeiro, A. (2016). Dsa: Decentralized Double Stochastic Averaging Gradient Algorithm. *J. Machine Learn. Res.* 17, 1–35.

Nesterov, Y. (2013). *Introductory Lectures on Convex Optimization: A Basic Course*, 87. Springer Science & Business Media.

Pu, S., and Nedić, A. (2018). Distributed Stochastic Gradient Tracking Methods. *Mathematical Programming* 187 (1), 409–457.

Qu, G., and Li, N. (2017). Accelerated distributed Nesterov gradient descent. *IEEE Transactions on Automatic Control* 65 (6), 2566–2581.

Recht, B., Re, C., Wright, S., and Niu, F. (2011). "Hogwild: A Lock-free Approach to Parallelizing Stochastic Gradient Descent," in Advances in Neural Information Processing Systems, Granada, Spain, December 12–17, 2011, 693–701.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. (2017). "Optimal Algorithms for Smooth and Strongly Convex Distributed Optimization in Networks," in International Conference on Machine Learning, Sydney, Australia, Auguset 6–11, 2017, 3027–3036.

Song, S., Chaudhuri, K., and Sarwate, A. D. (2015). Learning from Data with Heterogeneous Noise Using Sgd. *In Artificial Intelligence and Statistics*, 894–902.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). "On the Importance of Initialization and Momentum in Deep Learning," in International conference on machine learning, Atlanta, GA, June 16–21, 2013, 1139–1147.

Tan, F. (2020). The Algorithms of Distributed Learning and Distributed Estimation about Intelligent Wireless Sensor Network. *Sensors* 20 (5), 1302. doi:10.3390/s20051302

Tsianos, K. I., and Rabbat, M. G. (2012). "Distributed Strongly Convex Optimization," in Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on (IEEE), Allerton Park and Retreat Center, IL, October 1–5, 2012, 593–600. doi:10.1109/allerton.2012.6483272

Uribe, C. A., Lee, S., Gasnikov, A., and Nedić, A. (2020). A Dual Approach for Optimal Algorithms in Distributed Optimization over Networks. In 2020 Information Theory and Applications Workshop (ITA). IEEE, 1–37.

Wangni, J., Wang, J., Liu, J., and Zhang, T. (2017). *Gradient Sparsification for Communication-Efficient Distributed Optimization*. Montreal, Canada: arXiv. preprint arXiv:1710.09854.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., et al. (2017). "Terngrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning," in Advances in Neural Information Processing Systems, Long Beach, CA, December 4–9, 2017, 1508–1518.

Xu, Z., Taylor, G., Li, H., Figueiredo, M., Yuan, X., and Goldstein, T. (2017). "Adaptive Consensus Admm for Distributed Optimization," in International Conference on Machine Learning, Sydney, Australia, August 6–11, 2017, 3841–3850.

Yang, S., Tahir, Y., Chen, P.-y., Marshall, A., and McCann, J. (2016). "Distributed Optimization in Energy Harvesting Sensor Networks with Dynamic In-Network Data Processing," in IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications (IEEE), San Francisco, CA, April 10–15, 2016, 1–9. doi:10.1109/infocom.2016.7524475

Zhang, S., Choromanska, A. E., and LeCun, Y. (2015). "Deep Learning with Elastic Averaging Sgd," in Advances in Neural Information Processing Systems, Montreal, Canada, December 7–12, 2015, 685–693.

Zhang, W., Zhao, P., Zhu, W., Hoi, S. C., and Zhang, T. (2017). "Projection-free Distributed Online Learning in Networks," in International Conference on Machine Learning, Sydney, Australia, August 6–11, 2017, 4054–4062.

Zheng, S., Meng, Q., Wang, T., Chen, W., Yu, N., Ma, Z.-M., et al. (2017). "Asynchronous Stochastic Gradient Descent with Delay Compensation for Distributed Deep Learning," in International Conference on Machine Learning, Sydney, Australia, August 6–11, 2017, 4120–4129.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.