Revisiting Optimal Convergence Rate for Smooth and Non-convex Stochastic Decentralized Optimization

Kun Yuan[1,3]*, **Xinmeng Huang**[2*], **Yiming Chen**[1,4*], **Xiaohan Zhang**[2], **Yingya Zhang**[1], **Pan Pan**[1]

[1]DAMO Academy, Alibaba Group [2]University of Pennsylvania
[3]Peking University [4]MetaCarbon

{kun.yuan, yingya.zyy, panpan.pp}@alibaba-inc.com
xinmengh@sas.upenn.edu yiming@metacarbon.vip zxiaohan@seas.upenn.edu

Abstract

Decentralized optimization is effective to save communication in large-scale machine learning. Although numerous algorithms have been proposed with theoretical guarantees and empirical successes, the performance limits in decentralized optimization, especially the influence of network topology and its associated weight matrix on the optimal convergence rate, have not been fully understood. While Lu and Sa [44] have recently provided an optimal rate for non-convex stochastic decentralized optimization with weight matrices defined over linear graphs, the optimal rate with *general* weight matrices remains unclear.

This paper revisits non-convex stochastic decentralized optimization and establishes an optimal convergence rate with general weight matrices. In addition, we also establish the optimal rate when non-convex loss functions further satisfy the Polyak-Lojasiewicz (PL) condition. Following existing lines of analysis in literature cannot achieve these results. Instead, we leverage the Ring-Lattice graph to admit general weight matrices while maintaining the optimal relation between the graph diameter and weight matrix connectivity. Lastly, we develop a new decentralized algorithm to nearly attain the above two optimal rates under additional mild conditions.

1 Introduction

1.1 Motivation. Decentralized optimization is an emerging paradigm for large-scale machine learning. By letting each node average with its neighbors, decentralized algorithms save great communication overhead compared to traditional approaches with a central server. While numerous effective decentralized algorithms (see, e.g., [51, 11, 62, 39, 5, 41, 78]) have been proposed in literature showing both theoretical guarantees and empirical successes, the performance limits in decentralized optimization have not been fully clarified. This paper provides further understandings in optimal convergence rate in decentralized optimization and develops algorithms to achieve these rates.

Weight matrix and connectivity measure. Assume n computing nodes are connected by some network topology (graph). We associate the topology with a weight matrix $W = [w_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ in which $w_{ij} \in (0,1)$ if node j is connected to node i otherwise $w_{ij} = 0$. We introduce $\beta := \|W - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\| \in (0,1)$, where $\mathbb{1}_n = (1,\ldots,1)^T \in \mathbb{R}^n$ has all entries being 1, as the *connectivity measure* to gauge how well the network topology is connected. Quantity $\beta \to 0$ (which implies $W \to \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$) indicates a well-connected topology while $\beta \to 1$ (which implies $W \to I$) indicates a badly-connected topology. It is the weight matrix W and its connectivity measure β that bring major challenges to convergence analysis in decentralized optimization.

[*] Equal Contribution. Corresponding Author: Kun Yuan

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-2636

Table 1: Rate comparison between different algorithms in smooth and non-convex stochastic decentralized optimization. Parameter n denotes the number of all computing nodes, $\beta \in [0, 1)$ denotes the connectivity measure of the weight matrix, σ^2 measures the gradient noise, b^2 denotes data heterogeneity, and T is the number of iterations. Other constants such as the initialization $f(x^{(0)}) - f^\star$ and smoothness constant L are omitted for clarity. Detailed complexities with constant L are listed in Table 6 in Appendix B. The definition of transient iteration complexity can be found in Remark 6 (the smaller the better). "LB" is lower bound while "UB" is upper bound. Notation $\tilde{O}(\cdot)$ hides all logarithm factors.

	References	Gossip matrix	Convergence rate	Tran. iters.
LB	[44]	$\beta = \cos(\pi/n)$	$\Omega\big(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T(1-\beta)^{\frac{1}{2}}}\big)$	$O(\frac{n}{(1-\beta)\sigma^2})$
	Theorem 2	$\beta \in [0, \cos(\pi/n)]$	$\Omega\big(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T(1-\beta)^{\frac{1}{2}}}\big)$	$O(\frac{n}{(1-\beta)\sigma^2})$
UB	DSGD [30]	$\beta \in [0, 1)$	$O\big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{1}{3}}} + \frac{b^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{2}{3}}}\big)$	$O\big(\frac{n^3}{(1-\beta)^2\sigma^2}\big)^\dagger$
	D^2/ED [66]	$\beta \in [0, 1)$	$O\big(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T(1-\beta)^3}\big)$	$O(\frac{n}{(1-\beta)^6\sigma^2})$
	DSGT [29]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma}{\sqrt{nT}} + \frac{\sigma^{\frac{2}{3}}}{T^{\frac{2}{3}}(1-\beta)^{\frac{1}{3}}}\big)$	$\tilde{O}(\frac{n^3}{(1-\beta)^2\sigma^2})$
	DeTAG [44]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T(1-\beta)^{\frac{1}{2}}}\big)$	$\tilde{O}(\frac{n}{(1-\beta)\sigma^2})$
	MG-DSGD	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma}{\sqrt{nT}} + \frac{1}{T(1-\beta)^{\frac{1}{2}}}\big)$	$\tilde{O}(\frac{n}{(1-\beta)\sigma^2})$

† The complete complexity is $O(\frac{n^3}{(1-\beta)^2 \min\{\sigma^2,(1-\beta)^2 b^2\}})$, which reduces to $O(\frac{n^3}{(1-\beta)^2\sigma^2})$ for small σ^2.

Decentralized optimization. Decentralized approaches are built upon the partial averaging $x_i^+ = \sum_{j\in\mathcal{N}_i} w_{ij}x_j$ (or $x^+ = Wx$ in a more compact manner) where $\mathcal{N}_i$ is the set of neighbors of node i (including node i itself). The sparsity and connectivity of W significantly influence the efficiency and effectiveness of the partial averaging. Decentralized gradient descent [51, 79], diffusion [11, 59] and dual averaging [19] are well-known decentralized methods. Other advanced algorithms, which can correct the bias caused by heterogeneous data distributions, include decentralized ADMM [45, 63], explicit bias-correction [62, 80, 38], gradient tracking [50, 17, 54, 74], and dual acceleration [60, 68].

In the stochastic regime, decentralize SGD [11, 59, 39, 30] and its momentum and adaptive variants [41, 78, 47] have attracted a lot of attentions. After being extended to directed topologies [5, 75], time-varying topologies [30, 48, 69], asynchronous settings [40], and data-heterogeneous scenarios [66, 73, 43, 2, 29], decentralize SGD has much more applicable scenarios with significantly improved performance. Decentralized algorithms are also closely related to federated learning methods [46, 64, 76, 28, 37] when they admit weight matrices alternating between $\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T$ and the identity matrix.

Prior understandings in theoretical limits. A series of pioneering works have shed lights on the theoretical limits in decentralized optimization. These works establish the optimal convergence rate for convex or non-stochastic decentralized optimization [60, 61, 65, 32], and propose decentralized algorithms that (nearly) match these optimal bounds [60, 61, 68, 65, 33, 32]. However, there are few studies on the theoretical limits in non-convex stochastic decentralized optimization, which is quite common in large-scale deep learning, even when the costs are smooth.

A recent novel work [44] establishes the optimal complexity for smooth and non-convex stochastic decentralized optimization. While this optimal bound is inspiring, it is valid for a restrictive family of weight matrix W associated with the linear graph whose connectivity measure $\beta = \cos(\pi/n)$[2]. However, the linear graph is just one of the commonly used topologies. A more difficult but fundamental question still remains open:

What is the optimal convergence rate of smooth and non-convex stochastic decentralized optimization with a general weight matrix (that does not necessarily satisfy $\beta = \cos(\pi/n)$)?

It is not known whether the lower bound in [44] still holds for other commonly-used weight matrices resulted from grids, toruses, hypercubes, etc. whose connectivity measure $\beta \neq \cos(\pi/n)$.

Challenges. Following existing lines of analysis, to our knowledge, cannot provide an assuring answer. It is known that the communication complexity in decentralized optimization is typically

[2] See Corollary 1 in the latest arXiv version of [44] uploaded in Jan. 2022: arXiv:2006.08085v4.

Table 2: Rate comparison between different algorithms in smooth and non-convex stochastic decentralized optimization under the PL condition. Constants such as the initialization $f(x^{(0)}) - f^\star$ and smoothness constant L are omitted for clarity. Detailed complexities with constants L and μ are listed in Table 6 in Appendix B.

	References	**Gossip matrix**	**Convergence rate**	**Tran. iters. complexity**
LB	Theorem 3	$\beta \in [0, \cos(\pi/n)]$	$\Omega\big(\frac{\sigma^2}{nT} + e^{-T(1-\beta)^{\frac{1}{2}}}\big)$	$\tilde{O}(\frac{1}{(1-\beta)^{1/2}})$
UB	DSGD [30][†]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + \frac{\sigma^2}{T^2(1-\beta)} + \frac{b^2}{T^2(1-\beta)^2}\big)$	$\tilde{O}(\frac{n}{(1-\beta)\min\{1,(1-\beta)b^2\}})$
	DAGD[57][†‡]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + e^{-T(1-\beta)^{\frac{1}{2}}}\big)$	$\tilde{O}(\frac{1}{(1-\beta)^{1/2}})$
	D^2/ED [2, 77]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + \frac{\sigma^2}{T^2(1-\beta)} + e^{-T(1-\beta)}\big)$	$\tilde{O}(\frac{n}{1-\beta})$
	DSGT [2]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + \frac{\sigma^2}{T^2(1-\beta)} + e^{-T(1-\beta)}\big)$	$\tilde{O}(\frac{n}{1-\beta})$
	DSGT [73]	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + \frac{\sigma^2}{T^2(1-\beta)^3} + e^{-T(1-\beta)}\big)$	$\tilde{O}(\frac{n}{(1-\beta)^3})$
	MG-DSGD	$\beta \in [0, 1)$	$\tilde{O}\big(\frac{\sigma^2}{nT} + e^{-T(1-\beta)^{\frac{1}{2}}}\big)$	$\tilde{O}(\frac{1}{(1-\beta)^{1/2}})$

[†] This rate is derived under the strongly-convex assumption.

[‡] This rate is achieved by utilizing increasing (non-constant) mini-batch sizes.

proportional to the diameter D of the network topology [60]. Establishing the relation between D and β is key to clarifying the influence of the weight matrix on optimal convergence rate. Existing works [60, 61, 44] utilize the linear graph to establish the optimal relation as $D = \Theta(1/\sqrt{1-\beta})$. However, weight matrices associated with linear graph are quite *limited*. Given a fixed network size n (which is a prerequisite when discussing complexities in distributed stochastic optimization [44, 24, 31, 21]), they have to satisfy $\beta = \cos(\pi/n)$. Weight matrices with other β cannot be implied from the linear graph. To examine the lower bounds with a more general weight matrix W, [65] proposes a wider class of networks named linear-star graph, which, however, leads to *suboptimal* relation between D and β. As a result, a network topology that admits general weight matrices W while maintaining the optimal relation between D and β is critical to clarify the above fundamental question.

1.2 Main results. This paper successfully identifies the desired network topology which facilitates the theoretical limits over the general weight matrices. Our contributions are summarized as follows.

- We discover that ring-lattice graph maintains the optimal relation between D and β for a much broader class of weight matrices. Given any fixed n and $\beta \in [0, \cos(\pi/n)]$, we can always construct a weight matrix $W \in \mathbb{R}^{n\times n}$ associated with some ring-lattice graph, whose connectivity measure is β, that satisfies $D = \Theta(1/\sqrt{1-\beta})$. Since $\cos(\pi/n)$ approaches to 1 when n is large, the ring-lattice graph admits far broader weight matrices than linear graph used in [60, 61, 44].
- With the above results, we provide the optimal convergence rate for smooth and non-convex stochastic decentralized optimization that holds for any weight matrix with $\beta \in [0, \cos(\pi/n)]$. In contrast, the optimal rate in [44] only applies to weight matrix satisfying $\beta = \cos(\pi/n)$. We also provide the optimal convergence rate when the non-convex cost functions further satisfy the Polyak-Lojasiewicz (PL) condition [27] for any weight matrix with $\beta \in [0, \cos(\pi/n)]$.
- We prove that the above two optimal complexities can be nearly-attained (under mild assumptions and up to the condition number L/μ as well as other logarithmic factors) by simply integrating multiple gossip communication [42, 58] and gradient accumulation [60, 57, 44] to the vanilla decentralized SGD (D-SGD) algorithm. Compared to DeTAG [44], the extended D-SGD avoids the gradient tracking steps and saves half of the communication overheads. Compared to DAGD [57], the extended D-SGD achieves the near-optimality using constant mini-batch sizes and has theoretical guarantees in the non-convex (as well as the PL) scenario.

All established results in this paper as well as those of existing state-of-the-art decentralized algorithms are listed in Tables 1 and 2. The smoothness constant L and PL constant μ are omitted in these tables for clarity. The detailed complexities of these algorithms with constant L and μ are listed in Tables 6 and 7 in Appendix B.

1.3 Other related works. Some other related works are discussed as follows.

Lower bounds and optimal algorithms. The lower bounds and optimal algorithms for stochastic and convex optimization have been intensively studied in [35, 36, 55, 1, 18, 15]. References [9, 10] established the lower bounds to find stationary solutions for non-convex optimization, and [21, 3] proposed algorithms with near-optimal convergence rate for non-convex finite-sum minimization. In deterministic decentralized optimization, the optimal convergence rate was established for strongly convex optimization [60], non-smooth convex optimization [61], and non-convex optimization [65]. Furthermore, the lower bounds and optimal algorithms over time-varying topologies are studied in [33, 57, 58, 26]. In stochastic decentralized optimization, works [77] and [30] established the lower bounds for strongly-convex decentralized SGD. [44] proves the optimal complexities for non-convex problems when weight matrices satisfy $\beta = \cos(\pi/n)$.

Ring-lattice Graph. The ring-lattice graph was used in [70, 22, 12] as a basis model to generate the small world graph. However, its relation between graph diameter and connectivity measure is not clarified to our knowledge. A related work [72] examines the robustness of the ring-lattice graph, but does not discuss its connectivity measure. Furthermore, ring-lattice has never been utilized to establish the influence of the weight matrix on the optimal rate in decentralized optimization before.

2 Problem setup

Problem setup. Consider the following problem with a network of n computing nodes:

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad \text{where} \quad f_i(x) := \mathbb{E}_{\xi_i\sim D_i} F(x;\xi_i). \tag{1}$$

Function $f_i(x)$ is local to node i, and random variable ξ_i denotes the local data that follows distribution D_i. Each local data distribution D_i can be different across all nodes.

Assumptions. The optimal convergence rate is established under the following assumptions.

- **Function class.** We assume each local loss function is L-smooth, i.e.,

  $$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x-y\|, \quad \forall x, y \in \mathbb{R}^n, \tag{2}$$

  for some constant $L > 0$. We let the class $\mathcal{F}_L$ denote the set of all functions satisfying (2). We may also assume the global loss $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ satisfies the μ-PL condition:

  $$2\mu(f(x) - f^\star) \le \|\nabla f(x)\|^2, \quad \forall x \in \mathbb{R}^n, \tag{3}$$

  where $f^\star$ is the optimal function value in problem (1) and $\mu > 0$ is some constant. We let the function class $\mathcal{F}_{L,\mu}$ denote the set of all functions satisfying both (2) and (3).
- **Oracle class.** We assume each node i interacts with $f_i(x)$ via the stochastic gradient oracle $\tilde{g}_i(x,\xi)$ where ξ is a random variable. In addition, we assume oracle $\tilde{g}_i(x,\xi)$ satisfies

  $$\mathbb{E}_\xi[\tilde{g}_i(x,\xi)] = \nabla f_i(x) \quad \text{and} \quad \mathbb{E}_\xi\|\tilde{g}_i(x,\xi) - \nabla f_i(x)\|^2 \le \sigma^2. \tag{4}$$

  We let $\mathcal{O}_{\sigma^2}$ denote the class of all such stochastic gradient oracles.
- **Weight matrix class.** We let $\mathcal{W}_{n,\beta}$ denote the class of all gossip matrices $W \in \mathbb{R}^{n\times n}$ satisfying

  $$W\mathbb{1}_n = \mathbb{1}_n, \quad \mathbb{1}_n^T W = \mathbb{1}_n^T, \quad \|W - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\| = \beta \in [0,1), \tag{5}$$

  where $\mathbb{1}_n = (1,\ldots,1)^T \in \mathbb{R}^n$ with all entries being 1. If W is positive semi-definite, then connectivity measure β is its second largest eigenvalue.
- **Algorithm class.** We consider an algorithm A in which each node i accesses an unknown local function f_i via the stochastic gradient oracle $\tilde{g}_i(x;\xi_i) \in \mathcal{O}_{\sigma^2}$. Each node i running algorithm A will maintain a local model copy $x_i^{(k)}$ at iteration k. We assume A to follow the partial averaging policy, i.e., each node communicates via protocol

  $$z_i^{(k)} = \sum_{j\in\mathcal{N}_i} w_{ij} y_j^{(k)}, \quad \forall i \in [n]$$

  for some $W = [w_{ij}]_{i,j=1}^n \in \mathcal{W}_{n,\beta}$ where $[n] := \{1,\cdots,n\}$, and y and z are the input and output variables of the communication protocol. In addition, we assume A to follow the zero-respecting

https://doi.org/10.52202/068431-2636

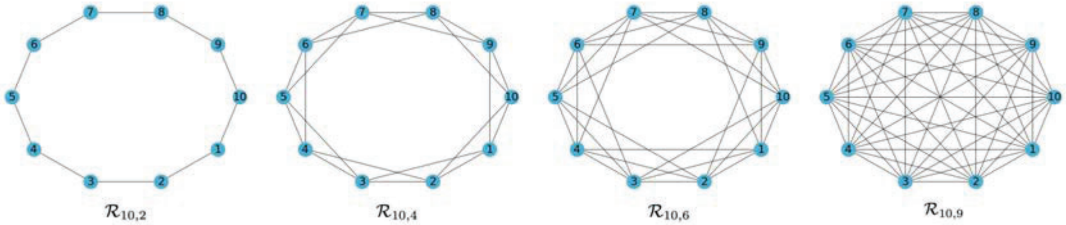

Figure 1: An illustration of the ring-lattice graph with size 10 and degrees $2, 4, 6, 9$. It is observed that $\mathcal{R}_{10,2}$ is a ring graph while $\mathcal{R}_{10,9}$ is a complete graph.

policy [9, 10, 25]. Informally speaking, the zero-respecting policy requires that the number of non-zero entries of local model copy $x_i^{(k)}$ can only be increased by sampling its own stochastic gradient oracle or interacting with the neighboring nodes. The zero-respecting policy is widely followed by all methods in Tables 1 and 2 and their momentum and adaptive variants. We let $\mathcal{A}_W$ be the set of all algorithms following the partial averaging and zero-respecting policies.

Lower bound metrics. With the above classes, we now introduce the measures for convergence rate's lower bound. Given loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_L$ (or $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{L,\mu}$), stochastic gradient oracles $\{\tilde{g}_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, a weight matrix $W \in \mathcal{W}_{n,\beta}$, an algorithm $A \in \mathcal{A}_W$, the number of gradient queries (or communications rounds) T, we let $\hat{x}_{A,\{f_i\}_{i=1}^n,\{\tilde{g}_i\}_{i=1}^n,W,T}$ (which will be denoted as $\hat{x}$ when there is no ambiguity) be the output of algorithm A with $\hat{x} \in \mathrm{Span}\big(\{\{x_j^{(t)}\}_{j=1}^n\}_{t=1}^T\big)$. The lower bound for smooth and non-convex stochastic decentralized optimization is defined as

$$\inf_{A\in\mathcal{A}_W} \sup_{W\in\mathcal{W}_{n,\beta}} \sup_{\{\tilde{g}_i\}_{i=1}^n\in\mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_L} \mathbb{E}\|\nabla f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{\tilde{g}_i\}_{i=1}^n,W,T})\|^2. \tag{6}$$

The lower bound under the PL condition is defined as

$$\inf_{A\in\mathcal{A}_W} \sup_{W\in\mathcal{W}_{n,\beta}} \sup_{\{\tilde{g}_i\}_{i=1}^n\in\mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_{L,\mu}} \mathbb{E}[f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{\tilde{g}_i\}_{i=1}^n,W,T}) - f^\star]. \tag{7}$$

Fixed network size n. This paper considers *stochastic* decentralized optimization where *the network size n is a fixed constant*. A fixed n is a prerequisite in distributed stochastic optimization which enables distributed algorithms to achieve the linear speedup in convergence rate $O(\sigma/\sqrt{nT})$, see the algorithms listed in Tables 1 and 2. In decentralized deterministic optimization, however, size n does not appear in the convergence rate. Thus, it does not need to be specified and can be varied freely when establishing the lower bounds [60, 61].

3 Ring-Lattice graph

The main challenge in analyzing theoretical limits in decentralized optimization lies in clarifying the influence of network topology and its associated weight matrix. It is intuitive that the graph diameter D affects the convergence rate. In the worst case, one node needs to take at least D communication steps to transmit its own gradient information to the other. If the relation between D and β is established, the influence of the weight matrix on lower bounds will become clear. However, existing results either establishes suboptimal relation between D and β [65] or admits restrictive weight matrices [60, 61, 44] given fixed n. Now we show that the ring-lattice graph maintains the optimal relation between D and β for a broad class of weight matrices. All proof details are in Appendix C.

Definition 1 (RING-LATTICE GRAPH [71])**.** *Given any positive integers n, k such that k is even and $2 \le k < n-1$, the k-regular ring-lattice graph over nodes $\{1, \cdots, n\}$, denoted by $\mathcal{R}_{n,k}$, is an undirected graph in which each node $i \in [n]$ is connected with the set of nodes $\{(i+\ell) \bmod n : 1 \le |\ell| \le k/2\}$ where $(i+\ell) \bmod n$ is defined as*

$$(i+\ell) \bmod n = \begin{cases} i+\ell & \textit{if } 1 \le i+\ell \le n; \\ i+\ell+n & \textit{if } i+\ell < 1; \\ i+\ell-n & \textit{if } i+\ell > n. \end{cases}$$

Apparently, each node in $\mathcal{R}_{n,k}$ has degree k.

Ring-lattice is a k-regular graph. When $k = 2$, ring-lattice will reduce to the ring graph. While Definition 1 excludes the case in which $k = n - 1$, the complete graph can be exceptionally viewed as the ring lattice with degree $k = n - 1$. Therefore, for general $2 \leq k \leq n - 1$, $\mathcal{R}_{n,k}$ can be regarded as an intermediate state between the ring and complete graphs, see the illustration in Fig. 1. The following lemma clarifies the diameter of $\mathcal{R}_{n,k}$.

Lemma 1. *The diameter of the ring-lattice graph $\mathcal{R}_{n,k}$ is $D = \Theta(n/k)$.*

It is derived from the above lemma that the diameters of $\mathcal{R}_{n,2}$ and $\mathcal{R}_{n,n}$ are $\Theta(n)$ and $\Theta(1)$, respectively, which are consistent with the ring and complete graphs.

The following fundamental theorem establishes the relation between D and connectivity measure β.

Theorem 1. *Given a fixed n and any $\beta \in [0, \cos(\pi/n)]$, we can always construct a ring-lattice graph $\mathcal{R}_{n,k}$ so that*

(1) it has an associated weight matrix $W \in \mathcal{W}_{n,\beta}$, i.e., $W \in \mathbb{R}^{n\times n}$ and $\|W - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^T\| = \beta$,

(2) its diameter D satisfies $D = \Theta(1/\sqrt{1-\beta})$.

Remark 1. *The works [60, 61, 44] utilize the linear graph to establish the same relation $D = \Theta(1/\sqrt{1-\beta})$. However, given fixed network size n, the linear graph admits limited weight matrices with $\beta = \cos(\pi/n)$.*

Remark 2. *Sun and Hong [65] propose a slightly wider class of networks named linear-star graph, which, however, leads to a suboptimal relation $D = \Omega(1/[n(1-\beta)])$ [65, Proof of Corollary 3.1].*

Remark 3. *It is worth noting that for deterministic decentralized optimization where n may vary, one can tune n to make $\beta = \cos(\pi/n)$ lie in the full interval $(0, 1)$ as shown in [60, 61]. In other words, [60, 61] proved that $D = \Theta(1/\sqrt{1-\beta})$ for any $\beta \in (0, 1)$ if n is a varying parameter. However, this result is not valid in stochastic scenarios where n is required to be fixed.*

4 Lower bounds with general weight matrices

This section will establish the lower bounds for smooth and non-convex decentralized stochastic optimization. All proof details are in Appendix D. When the cost function does not satisfy the PL condition (3), we will derive that the convergence rate is lower bounded by $\Omega(\frac{\sigma}{\sqrt{nT}} + \frac{D}{T})$. This result together with Theorem 1 leads to:

Theorem 2. *For any $L > 0$, $n \geq 2$, $\beta \in [0, \cos(\pi/n)]$, and $\sigma > 0$, there exists a set of loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_L$, a set of stochastic gradient oracles $\{\tilde{g}_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, and a weight matrix $W \in \mathcal{W}_{n,\beta}$, such that it holds for any $A \in \mathcal{A}_W$ starting form $x^{(0)}$ that*

$$\mathbb{E}\|\nabla f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{\tilde{g}_i\}_{i=1}^n,W,T})\|^2 = \Omega\left(\frac{\sigma\sqrt{L}}{\sqrt{nT}} + \frac{L}{T\sqrt{1-\beta}}\right). \tag{8}$$

Remark 4 (Comparison with existing bounds). *Comparing to the lower bound established in [44], Theorem 2 admits a much broader class of weight matrices with $\beta \in [0, \cos(\pi/n)]$. While the interval $[0, \cos(\pi/n)] \subset [0, 1)$, it approaches to $[0, 1)$ as n goes large. Such interval is broad enough to cover most weight matrices (generated through the Laplacian rule $W = I - L/d_{\max}$) resulted from common topologies such as line, ring, grid, torus, hypercube, exponential graph, complete graph, Erdos-Renyi graph, geometric random graph, etc. whose β lies in the interval $[0, \cos(\pi/n)]$ when n is sufficiently large, see details in Appendix D.*

Remark 5 (Linear speedup). *The first term $1/\sqrt{nT}$ dominates (8) when T is sufficiently large, which will require $T = O(1/(n\epsilon^2))$ (it is inversely proportional to n) iterations to reach accuracy ϵ. Therefore, if the term involving nT is dominating the rate for some T, we say the algorithm is in its linear-speedup stage. Linear speedup is a fundamental property that distributed algorithms enjoy.*

Remark 6 (Transient iteration complexity). *Due to the topology-incurred overhead in convergence rate, a decentralized algorithm has to experience a transient stage to achieve linear-speedup. Transient iterations are referred to those iterations before an algorithm reaches linear-speedup stage. To reach linear speedup in the lower bound (8), T has to satisfy $\frac{1}{T\sqrt{1-\beta}} \leq \frac{\sigma}{\sqrt{nT}}$ and hence $T = \mathcal{O}(\frac{n}{(1-\beta)\sigma^2})$, which is the transient iteration complexity shown in the first line in Table 1. The smaller the transient iterations are, the faster the algorithm can achieve linear speedup. Transient iteration complexity is an important metric to evaluate how sensitive the algorithm is to β.*

https://doi.org/10.52202/068431-2636

Algorithm 1: Decentralized SGD with multiple gossip steps (MG-DSGD)

```
Require: Initialize $x_i^{(0)} = 0$, the weight matrix $W$, and the rounds of gossip steps $R$.
for $k = 0, 1, 2, ..., K$, every node $i$ do
    Sample $\{\xi_i^{(k,r)}\}_{r=1}^R$ independently and let $g_i^{(k)} = \frac{1}{R}\sum_{r=1}^{R}\nabla F(x_i^{(k)}; \xi_i^{(k,r)})$;   ▷ grad. accumulation
    Update $\phi_i^{(k+1)} = x_i^{(k)} - \gamma g_i^{(k)}$;   ▷ local stochastic gradient descent
    Update $x_i^{(k+1)} = \textbf{FastGossipAverage}(\{\phi_i^{(k+1)}\}_{i=1}^n, W, R)$;   ▷ multiple gossips
```

Algorithm 2: $x_i = \textbf{FastGossipAverage}(\{\phi_i\}_{i=1}^n, W, R)$

```
Require: $\{\phi_i\}_{i=1}^n$, $W$, $R$, and $\beta$; let $z_i^{(0)} = z_i^{(-1)} = \phi_i$ and $\eta = \frac{1-\sqrt{1-\beta^2}}{1+\sqrt{1+\beta^2}}$.
for $r = 0, 1, 2, ..., R-1$, every node $i$ do
    Update $z_i^{(r+1)} = (1+\eta)\sum_{j\in\mathcal{N}_i} w_{ij} z_i^{(r)} - \eta z_i^{(r-1)}$;   ▷ fast gossip averaging
Output: $x_i = z_i^{(R)}$;
```

When the cost function satisfies PL condition (3), we prove the convergence rate is lower bounded by $\Omega(\frac{\sigma}{\sqrt{nT}} + \exp(-CT/D))$ where C is some constant. This result together with Theorem 1 leads to:

Theorem 3. *For any $L > 0$, $n \geq 2$, $\beta \in [0, \cos(\pi/n)]$, and $\sigma > 0$, there exists a set of loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{L,\mu}$, a set of stochastic gradient oracles $\{\tilde{g}_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, and a weight matrix $W \in \mathcal{W}_{n,\beta}$, such that it holds for any $A \in \mathcal{A}_W$ starting from $x^{(0)}$ that*

$$\mathbb{E}[f(\hat{x}_{A,\{f_i\}_{i=1}^n,\{\tilde{g}_i\}_{i=1}^n,W,T}) - f^\star] = \Omega\left(\frac{\sigma^2}{\mu n T} + \frac{\mu}{L}\exp(-T\sqrt{\mu/L}\sqrt{1-\beta})\right). \quad (9)$$

Rate (9) *results in an $\tilde{\mathcal{O}}((1-\beta)^{-\frac{1}{2}})$ transient complexity.*

Expression (9) is the first lower bound for smooth and non-convex stochastic decentralized optimization under the PL condition to our knowledge. Please note that the above lower bound also applies to strongly-convex stochastic decentralized optimization, see the construction details in Appendix D.

5 DSGD with multiple gossip communications

This section will present a decentralized algorithm that achieve the lower bounds established in Sec. 4 up to constants L and μ. The new algorithm is a direct extension of the vanilla decentralized SGD (DSGD) [11, 30]. Inspired by the algorithm development in [57, 44], we add two additional components to DSGD: gradient accumulation and multiple-gossip communication. The main recursions are listed in Algorithm 1 which utilizes the fast gossip average step [42] in Algorithm 2. We call the new algorithm as MG-DSGD where "MG" indicates "multiple gossips". All proofs are in Appendix E.

The DeTAG algorithm proposed in [44] applies the gradient accumulation and multiple-gossip communication techniques to gradient tracking, which achieves the optimal rate in non-convex (without PL condition) stochastic decentralized optimization. This paper finds in MG-DSGD that the multiple gossip technique can significantly reduce the influence of data heterogeneity, and the gradient tracking step can be saved to achieve the convergence optimality. Similar observations also appeared in [57] albeit in the strongly-convex scenario. Removing gradient tracking from algorithm updates brings two benefits. First, it saves half of the communication overhead and memory storage per iteration. Second, it enables numerous effective acceleration techniques designed exclusively for decentralized SGD such as Quasi-global momentum [41], DecentLaM [78], adaptive momentum [47], etc. which cannot be easily extended to gradient tracking.

Let T be the toal number of gradient queries (or decentralized communications) at each node. Since each node takes R gradient queries and R communications at round k, it holds that $T = KR$ when MG-DSGD finishes after K rounds. The following theorems clarify the convergence rate of MG-DSGD where $T = KR$. We omit the initialization constants below. Our analysis techniques relate to the classic SGD analysis but with inaccurate oracles in the nonconvex case [20].

Theorem 4. *Given $L > 0$, $n \geq 2$, $\beta \in [0,1)$, $\sigma > 0$, and let A denote Algorithm 1. Assuming that $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla f(x)\|^2 \leq b^2$ for any $x \in \mathbb{R}^n$ and the number of gossip rounds R is set as in Appendix E, the convergence of A can be bounded for any $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_L$ and any $W \in \mathcal{W}_{n,\beta}$ that*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla f(\bar{x}^{(k)})\|^2 = \tilde{O}\left(\frac{\sigma\sqrt{L}}{\sqrt{nT}} + \frac{L}{T\sqrt{1-\beta}}\right) \quad \textit{where} \quad \bar{x}^{(k)} = \frac{1}{n}\sum_{i=1}^{n} x_i^{(k)}. \tag{10}$$

Remark 7. *The rate* (10) *matches with the lower bound* (8) *up to logarithm factors. Furthermore, we remark that the gradient similarity assumption $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla f(x)\|^2 \leq b^2$ is not required to obtain the lower bound in Theorem 2. Due to these two restrictions, MG-DSGD is a nearly-optimal (not fully-optimal) algorithm. However, while the analysis in MG-DSGD needs to assume bounded gradient similarity b^2, it only affects the convergence rate as a logarithm term hidden in the $\tilde{\mathcal{O}}(\cdot)$ notation. If one wants to completely removes the gradient similarity assumption from Theorem 4, he/she can refer to the techniques in gradient tracking [44, 73, 43].*

The rate (10) matches with the lower bound (8) up to logarithm factors. The comparison between MG-DSGD with other state-of-the-art algorithms is listed in Table 1. MG-DSGD matches DeTAG in convergence rate and saves half of the communications, see experiments in Fig. 2. While the analysis in MG-DSGD needs to assume bounded data heterogeneity b^2, it only affects the convergence rate as a logarithm term hidden in the $\tilde{\mathcal{O}}(\cdot)$ notation.

Theorem 5. *Under the same assumptions as in Theorem 4 and setting R as in Appendix E, the convergence of A can be bounded for any loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{L,\mu}$, and any $W \in \mathcal{W}_{n,\beta}$ by*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[f(\bar{x}^{(k)}) - f^\star] = \tilde{O}\left(\frac{L\sigma^2}{\mu^2 nT} + \exp(-\mu T\sqrt{1-\beta}/L)\right). \tag{11}$$

Remark 8. *Comparing* (11) *and lower bound* (9)*, we find its dependence on σ^2, n, T, and $1-\beta$ is optimal while that on L and μ is not. One reason is that our algorithm does not involve any Nesterov acceleration [52] mechanism. The established upper bound in a recent work [57] is closer to the lower bound* (9) *by utilizing Nesterov acceleration, see Table 7 in Appendix B. However, it requires an increasingly large mini-batch of data per iteration, and only works for the strongly-convex scenario. It is not known whether the acceleration mechanisms in [57] can be extended to the non-convex and PL scenario. The bound* (11) *is state-of-the-art under the general PL condition to our knowledge.*

6 Experiments

This section will validate our theoretical results by empirically comparing different decentralized algorithms DSGD [39], D^2 [67], DSGT [82], DeTAG [44] and MG-DSGD in deep learning.

Implementation setup. We implement all decentralized algorithms with PyTorch [53] 1.6.0 using NCCL 2.8.3 (CUDA 10.1) as the communication backend, in which the partial averaging primitive is implemented by NCCL send/receive primitives. For parallel SGD, we used PyTorch's native Distributed Data Parallel (DDP) module. All the models and training scripts in this section run on servers with 8 NVIDIA V100 GPUs with each GPU treated as one node. Similar to existing works [44], we use Ring graph throughout all experiments.

Tasks and training details. A series of experiments are carried out with CIFAR-10 [34] and ImageNet [16] to compare the aforementioned methods. For CIFAR-10 dataset, it consists of 50,000 training images and 10,000 validation images in 10 classes. For ImageNet dataset, it consists of 1,281,167 training images and 50,000 validation images in 1000 classes. We utilize ResNet variants [23] on CIFAR-10 (ResNet-20 with 0.27M parameters and ResNet-18 with 11.17M parameters) and ResNet-50 on ImageNet. For training protocol, we train total 300 epochs and the learning rate is warmed up in the first 5 epochs and is decayed by a factor of 10 at 150 and 250-th epoch. For learning rate, we tuned a strong baseline in the PSGD setting (5e-3 for single node) and used the same setting in all decentralized methods. The batch size is set to 128 on each node. All CIFAR-10 experiments are repeated three times with different seeds. For fair comparison, we maintain the total number of gradient queries the same for algorithms in experiments, i.e., $T = KR$.

Performance with homogeneous data. Table 3 compares the proposed MG-DSGD with centralized parallel SGD and other baselines with homogeneous data distributions. For DeTAG and MG-DSGD, the communication round R is set as 2. We also performed an ablation study on the effect of

https://doi.org/10.52202/068431-2636

Table 3: Accuracy comparison on CIFAR-10 with homogeneous data (8 nodes).

	With Momentum		Without Momentum	
Methods	ResNet18	ResNet20	ResNet18	ResNet20
PSGD	93.99 ± 0.52	91.62 ± 0.13	93.22 ± 0.19	89.06 ± 0.08
DSGD	94.54 ± 0.05	91.46 ± 0.06	93.67 ± 0.04	89.14 ± 0.08
DSGT	94.34 ± 0.04	91.64 ± 0.01	93.57 ± 0.02	88.85 ± 0.11
D^2	92.51 ± 0.07	89.83 ± 0.08	92.18 ± 0.08	88.67 ± 0.11
DeTAG	94.40 ± 0.13	**91.77 ± 0.25**	93.17 ± 0.18	88.97 ± 0.11
MG-DSGD	**94.57 ± 0.05**	**91.77 ± 0.09**	**93.75 ± 0.12**	**89.18 ± 0.08**

Table 4: Accuracy comparison on CIFAR-10 with heterogeneous data (ResNet-20).

Methods	DSGD	DSGT	D^2	DeTAG	MG-DSGD
$\alpha = 0.1$	58.14 ± 1.76	55.16 ± 0.42	53.85 ± 0.40	58.77 ± 1.30	**59.42 ± 3.36**
$\alpha = 1$	84.74 ± 0.97	84.74 ± 0.32	84.1 ± 0.64	84.82 ± 0.19	**84.91 ± 0.74**
$\alpha = 10$	90.19 ± 0.47	89.83 ± 0.12	89.80 ± 0.25	90.27 ± 0.08	**90.43 ± 0.04**

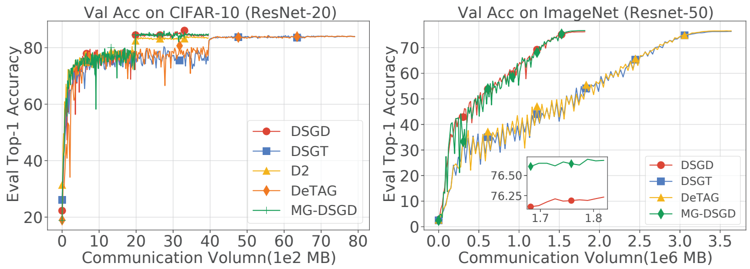

Figure 2: Accuracy in terms of the communication volumes. Left: CIFAR-10 with heterogeneous data ($\alpha = 1$); Right: ImageNet with homogeneous data.

momentum acceleration. Except for the ResNet-20 scenario with momentum in which MG-DSGD and DeTAG achieve the same performance, MG-DSGD consistently outperforms baseline methods in other scenarios. Moreover, MG-DSGD is more communication-efficient than DSGT and DeTAG.

Performance with heteogeneous data. Following previous works [81, 41], we simulate data heterogeneity among nodes with a Dirichlet distribution-based partitioning. A quantity α controls the degree of the data heterogeneity. The training data for a particular class tends to concentrate in a single node as $\alpha \to 0$, i.e. becoming more heterogeneous, while the homogeneous data distribution is achieved as $\alpha \to \infty$. We tested $\alpha = 0.1/1/10$ for all compared methods as corresponding to a setting with high/mild/low heterogeneity. As shown in Table 4, DeTAG and MG-DSGD outperform the other baselines by visible margins, and MG-DSGD is with slightly better accuracies. The left plot in Fig. 2 illustrates that MG-DSGD achieve a competitive accuracy to DeTAG using much less communication budgets on CIFAR-10 with $\alpha = 1$.

Large-batch training in ImageNet. To train ResNet-50 on ImageNet, utilization of the large-batch is necessary to speedup the training process. Since MG-DSGD is based on DSGD, we can extend the large-batch acceleration techniques introduced in [78, 41] to MG-DSGD to achieve better performance. These techniques are developed exclusively for DSGD and cannot be integrated to DeTAG easily. Following [78], we train total 120 epochs. The learning rate is warmed up in the first 20 epochs and is decayed in a cosine annealing scheduler. The results are listed in Table 5, showing that MG-DSGD has better accuracy than other approaches. Fig. 2(right) illustrates how different algorithms perform in terms of communication budgets used.

Table 5: Performance of different methods (ImageNet).

Methods	Acc.
DSGD	76.23 ± 0.06
DSGT	76.37 ± 0.07
DeTAG	76.57 ± 0.02
MG-DSGD	**76.64 ± 0.03**

More results. In Appendix F, we depict the accuracy curves in terms of epochs (rather than communication volumes). The ablation on the influence of inner loop R is also listed there.

7 Conclusion

This paper revisits the theoretical limits in smooth and non-convex stochastic decentralized optimization. Existing lower bound proved in [44] is only valid for special weight matrices with connectivity measure $\beta = \cos(\pi/n)$. It is not known what the lower bound is when using weight matrices with other β. To address this issue, this paper establishes the lower bound for any weight matrix with $\beta \in [0, \cos(\pi/n)]$. We also clarify the lower bound when loss functions satisfy the PL condition. A new algorithm is proposed to match these lower bounds up to logarithm factors.

Acknowledgements

The authors are grateful to all anonymous reviewers in NeurIPS 2022 for the very helpful discussions during the rebuttal process as well as useful references on accelerated decentralized gradient descent.

References

[1] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 22, 2009.

[2] Sulaiman A Alghunaim and Kun Yuan. A unified and refined convergence analysis for non-convex decentralized learning. *arXiv preprint arXiv:2110.09993*, 2021.

[3] Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[4] Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake E. Woodworth. Lower bounds for non-convex stochastic optimization. *ArXiv*, abs/1912.02365, 2019.

[5] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning (ICML)*, pages 344–353, 2019.

[6] Itai Benjamini, Gady Kozma, and Nicholas Wormald. The mixing time of the giant component of a random graph. *Random Structures & Algorithms*, 45(3):383–407, 2014.

[7] Stephen P Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Mixing times for random walks on geometric random graphs. In *ALENEX/ANALCO*, pages 240–249, 2005.

[8] Lucien Le Cam. Asymptotic methods in statistical decision theory. 1986.

[9] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020.

[10] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

[11] Jianshu Chen and Ali H Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[12] Yu Chen, Sanjeev Khanna, and Huan Li. On weighted graph sparsification by linear sketching. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

[13] Yiming Chen, Kun Yuan, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Accelerating gossip SGD with periodic global averaging. In *International Conference on Machine Learning (ICML)*, 2021.

[14] Imre Csiszár and János Körner. *Information Theory - Coding Theorems for Discrete Memoryless Systems, Second Edition*. Cambridge University Press, 2011.

[15] Anindya De, Sanjeev Khanna, Huan Li, and MohammadHesam NikpeySalekde. Approximate optimization of convex functions with outlier noise. *Advances in Neural Information Processing Systems*, 2021.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[17] P. Di Lorenzo and G. Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[18] Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. In *Conference on Learning Theory*, pages 1132–1157. PMLR, 2019.

[19] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

https://doi.org/10.52202/068431-2636

[20] Pavel Dvurechensky. Gradient method with inexact oracle for composite non-convex optimization, 2017.

[21] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

[22] Brian Hayes. Computing science: Graph theory in practice: Part ii. *American Scientist*, 88(2):104–109, 2000.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[24] Samuel Horvath, D. Kovalev, Konstantin Mishchenko, Sebastian U. Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv*, 2019.

[25] Xinmeng Huang, Yiming Chen, Wotao Yin, and Kun Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[26] Xinmeng Huang and Kun Yuan. Optimal complexity in non-convex decentralized learning over time-varying networks. *arXiv preprint arXiv:2211.00533*, 2022.

[27] Hamed Karimi, Julie Nutini, and Mark Schmidt. 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*, pages 795–811. Springer, 2016.

[28] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, pages 5132–5143. PMLR, 2020.

[29] Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[30] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning (ICML)*, pages 1–12, 2020.

[31] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, 2020.

[32] Dmitry Kovalev, Elnur Gasanov, Alexander Gasnikov, and Peter Richtarik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[33] Dmitry Kovalev, Adil Salim, and Peter Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18342–18352, 2020.

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[35] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

[36] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, 171(1):167–215, 2018.

[37] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[38] Z. Li, W. Shi, and M. Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, July 2019. early acces. Also available on arXiv:1704.07807.

[39] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5330–5340, 2017.

[40] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, pages 3043–3052, 2018.

https://doi.org/10.52202/068431-2636

[41] Tao Lin, Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *International Conference on Machine Learning*, 2021.

[42] Ji Liu and A Stephen Morse. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2):160–165, 2011.

[43] Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321. IEEE, 2019.

[44] Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning (ICML)*, pages 7111–7123. PMLR, 2021.

[45] Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B Giannakis. Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, 2010.

[46] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[47] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

[48] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

[49] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[50] A. Nedic, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[51] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[52] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.

[54] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

[55] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning (ICML)*, pages 1571–1578, 2012.

[56] Aaditya Ramdas and Aarti Singh. Optimal rates for stochastic convex optimization under tsybakov noise condition. In *International Conference on Machine Learning*, pages 365–373, 2013.

[57] Alexander Rogozin, Mikhail Mikhailovich Bochko, Pavel E. Dvurechensky, Alexander V. Gasnikov, and Vladislav Lukoshkin. An accelerated method for decentralized distributed stochastic optimization over time-varying graphs. *IEEE Conference on Decision and Control (CDC)*, 2021.

[58] Alexander Rogozin, Vladislav Lukoshkin, Alexander Gasnikov, Dmitry Kovalev, and Egor Shulgin. Towards accelerated rates for distributed optimization over time-varying networks. In *International Conference on Optimization and Applications*, 2021.

[59] Ali H Sayed. Adaptive networks. *Proceedings of the IEEE*, 102(4):460–497, 2014.

[60] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning (ICML)*, pages 3027–3036, 2017.

[61] Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2740–2749, 2018.

[62] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

https://doi.org/10.52202/068431-2636

[63] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[64] Sebastian Urban Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019.

[65] Haoran Sun and Mingyi Hong. Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Transactions on Signal processing*, 67(22):5912–5928, 2019.

[66] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856, 2018.

[67] Hanlin Tang, Chen Yu, Xiangru Lian, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pages 6155–6165. PMLR, 2019.

[68] César A Uribe, Soomin Lee, Alexander Gasnikov, and Angelia Nedić. A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pages 1–40, 2020.

[69] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. MATCHA: Speeding up decentralized SGD via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019.

[70] Duncan J Watts. Small worlds: the dynamics of networks between order and randomness, 1999.

[71] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’networks. *nature*, 393(6684):440–442, 1998.

[72] Jun Wu, Mauricio Barahona, Yue-jin Tan, and Hong-zhong Deng. Robustness of random graphs based on graph spectra. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):043101, 2012.

[73] Ran Xin, Usman A Khan, and Soummya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 2020.

[74] Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *IEEE Conference on Decision and Control (CDC)*, pages 2055–2060, Osaka, Japan, 2015.

[75] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

[76] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019.

[77] Kun Yuan, Sulaiman A Alghunaim, and Xinmeng Huang. Removing data heterogeneity influence enhances network topology dependence of decentralized sgd. *arXiv preprint arXiv:2105.08023*, 2021.

[78] Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. DecentLaM: Decentralized momentum SGD for large-batch deep training. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[79] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[80] Kun Yuan, Bicheng Ying, Xiaochuan Zhao, and Ali H. Sayed. Exact dffusion for distributed optimization and learning – Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3):708 – 723, 2019.

[81] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019.

[82] Jiaqi Zhang and Keyou You. Decentralized stochastic gradient tracking for non-convex empirical risk minimization. *arXiv preprint arXiv:1909.02712*, 2019.

Checklist

1. For all authors...

https://doi.org/10.52202/068431-2636

   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 6.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 6.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]