

# Decentralized Sum-of-Nonconvex Optimization

Zhuanghua Liu<sup>1, 2</sup>, Bryan Kian Hsiang Low<sup>1</sup>

<sup>1</sup>Department of Computer Science, National University of Singapore

<sup>2</sup>CNRS@CREATE LTD, 1 Create Way, #08-01 CREATE Tower, Singapore 138602

liuzhuanghua9@gmail.com, lowkh@comp.nus.edu.sg

## Abstract

We consider the optimization problem of minimizing the sum-of-nonconvex function, i.e., a convex function that is the average of nonconvex components. The existing stochastic algorithms for such a problem only focus on a single machine and the centralized scenario. In this paper, we study the sum-of-nonconvex optimization in the decentralized setting. We present a new theoretical analysis of the PMGT-SVRG algorithm for this problem and prove the linear convergence of their approach. However, the convergence rate of the PMGT-SVRG algorithm has a linear dependency on the condition number, which is undesirable for the ill-conditioned problem. To remedy this issue, we propose an accelerated stochastic decentralized first-order algorithm by incorporating the techniques of acceleration, gradient tracking, and multi-consensus mixing into the SVRG algorithm. The convergence rate of the proposed method has a square-root dependency on the condition number. The numerical experiments validate the theoretical guarantee of our proposed algorithms on both synthetic and real-world datasets.

## 1 Introduction

The exponential growth of data in the past decades has sparked substantial interest in developing algorithms distributed over multiple agents. A common scenario is that each agent within some network topology owns a disjoint subset of data, and they collaborate to tackle a global optimization objective. The network topology in which each agent resides can be classified into two categories: client-server vs. decentralized settings. For the former setting, a central parameter server communicates with all the workers and aggregates the information collected from them (Li et al. 2014). When there is a large volume of data on each agent, the central server becomes the bottleneck in the whole network. For the latter setting, each agent only communicates with its direct neighbors to exchange their information and finish the global task (Lian et al. 2017).

This paper focuses on stochastic optimization for minimizing the sum-of-nonconvex objective function in the decentralized setting. We formulate our problem as a convex optimization problem collaboratively solved by  $m$  agents in the network. Consider the following composite optimization

objective function:

$$\min_{x \in \mathbb{R}^d} F(x) := f(x) + \psi(x), \quad f(x) := \frac{1}{m} \sum_{i=1}^m f_i(x) \quad (1)$$

where  $f(x)$  is convex and smooth, and  $\psi(\cdot)$  is convex but possibly non-smooth (e.g.,  $\ell_1$ -regularization term). We suppose there are  $m$  agents and the  $i$ -th agent stores the local objective function  $f_i(x)$  which can be written as a finite-sum form:

$$f_i(x) := \frac{1}{n} \sum_{j=1}^n f_{i,j}(x)$$

where  $n$  is the number of components and each  $f_{i,j}(x)$  is smooth but possibly non-convex. The sum-of-nonconvex optimization is common in real-world applications, including 1-PCA (Saad 2011),  $k$ -PCA (Allen-Zhu and Li 2016), online eigenvector problem (Allen-Zhu and Li 2017), and general nonconvex optimization (Agarwal et al. 2016; Carmon et al. 2018).

Existing decentralized first-order optimization algorithms suffer several limitations in solving Problem (1): (i) The decentralized deterministic algorithms, such as EXTRA (Shi et al. 2015a), Exact-Diffusion (Yuan et al. 2018), P2D2 (Alghunaim, Yuan, and Sayed 2019), and SONATA (Sun, Daneshmand, and Scutari 2019), need access to the full gradient at each round. The computational cost of each iteration is prohibitively expensive on massive datasets. (ii) While existing decentralized stochastic methods achieve a cheaper per-iteration cost by sampling a minibatch of samples, the theoretical analysis of these approaches is not specialized in the sum-of-nonconvex optimization problem. One class of methods (Shi et al. 2015a; Xin, Khan, and Kar 2020; Ye, Xiong, and Zhang 2020) assumes that all component functions  $f_{i,j}(\cdot)$  are convex such that the global objective function  $f(\cdot)$  is convex<sup>1</sup>. Convergence analysis of these works does not apply to Problem (1) due to the mismatch of problem assumptions. The other class of methods (Li, Li, and Chi 2022; Luo and Ye 2022; Xin, Khan, and Kar 2022) assumes that component functions  $f_{i,j}(\cdot)$  are nonconvex and

<sup>1</sup>Ye, Xiong, and Zhang (2020) have claimed that each component function can be possibly nonconvex. However, Assumption 1 in their work requires each component function to be both  $L$ -smooth and convex, otherwise Eq. (3) cannot be satisfied.

the global objective function  $f(\cdot)$  is also possibly nonconvex. Consequently, the rate achieved by these methods is not optimal for Problem (1).

In this paper, we intend to design communication- and computation-efficient optimization algorithms for the sum-of-nonconvex problem in the decentralized setting. We start by presenting a new theoretical analysis of the PMGT-SVRG proposed by Ye, Xiong, and Zhang (2020) for objective function (1). It achieves the stochastic first-order oracle (SFO) complexity of  $\mathcal{O}((n + \sqrt{n\kappa}) \log(1/\epsilon))$  and communication complexity of  $\tilde{\mathcal{O}}((\sqrt{n} + \kappa) \xi \log(1/\epsilon))$  where  $\kappa$  is the condition number and  $\xi$  is some constant depending on the underlying network structure. Notice that both the computational and communication complexities of PMGT-SVRG have a linear dependency on the condition number  $\kappa$ , which can be exceptionally expensive when the objective function is ill-conditioned.

To remedy this issue, we propose an accelerated stochastic variance-reduced proximal-gradient optimization method called PMGT-KatyushaX for Problem (1) to improve the dependency of complexities on the condition number. Specifically, the vanilla KatyushaX algorithm proposed by Allen-Zhu (2018) achieves the SFO complexity with a square-root dependency on the condition number on a single machine. To extend the KatyushaX to the decentralized setting, we incorporate the powerful ideas of acceleration (Allen-Zhu and Orecchia 2014), gradient tracking (Di Lorenzo and Scutari 2016; Qu and Li 2017), and multi-consensus mixing (Liu and Morse 2011) into the SVRG algorithm. The resulting PMGT-KatyushaX achieves the stochastic first-order oracle (SFO) complexity of  $\mathcal{O}((n + n^{\frac{3}{4}} \sqrt{\kappa}) \xi \log(1/\epsilon))$  and the communication complexity of  $\tilde{\mathcal{O}}((\sqrt{n} + n^{\frac{1}{4}} \sqrt{\kappa}) \xi \log(1/\epsilon))$ . It is worth noting that the SFO complexity of our proposed algorithm matches the best-known result (Allen-Zhu 2018) for a single machine.

Numerical experiments on several synthetic and real-world datasets demonstrate significant improvement of our proposed PMGT-KatyushaX over existing baseline methods.

**Paper Organization** A review of related literature on decentralized stochastic first-order methods and stochastic sum-of-nonconvex optimization is presented in Section 2. In Section 3, we introduce the notations and problem setting of decentralized sum-of-nonconvex optimization. We present the theoretical result of PMGT-SVRG on Problem (1) in Section 4. We formally present our proposed algorithm PMGT-KatyushaX with the main theorem in Section 5. A proof sketch is provided for the main theorem in Section 6. Numerical results are presented in Section 7. Finally, we conclude this paper with a summary of our results in Section 8.

## 2 Related Work

In this section, we review related literature on decentralized stochastic first-order algorithms. In addition, we summarize existing works of stochastic optimization for the sum-of-nonconvex problem on a single machine.

### 2.1 Decentralized Stochastic First-Order Methods

We review related work about decentralized stochastic first-order methods for objective functions when each local function  $f_i(\cdot)$  has the finite-sum structure. These methods can be divided into two categories based on the convexity of the component function.

**$f_{i,j}(\cdot)$  is convex** The first decentralized variance-reduced method called DSA was proposed by Mokhtari and Ribeiro (2016), and it is a combination of EXTRA (Shi et al. 2015a) and SAGA (Defazio, Bach, and Lacoste-Julien 2014). DB-SA/ADFS (Hendrikx, Bach, and Massoulié 2019; Hendrikx, Bach, and Massoulié 2021; Shen et al. 2018) attempted to accelerate DSA with proximal mapping and variance reduction. Although several works (Hendrikx, Bach, and Massoulié 2020; Xin, Khan, and Kar 2020) have proposed proximal mapping-free algorithms, the computation and communication complexities of these methods are worse than DBSA and ADFS. Li et al. (2020); Ye, Xiong, and Zhang (2020) proposed decentralized stochastic algorithms that achieve a linear convergence rate by incorporating variance reduction, gradient tracking, and multi-consensus mixing.

**$f_{i,j}(\cdot)$  is nonconvex** All existing decentralized stochastic methods assume the global objective function  $f(\cdot)$  is possibly nonconvex if the component function  $f_{i,j}(\cdot)$  is nonconvex. Although the analysis of these approaches can be applied to our setting, the resulting convergence rate may not be optimal for the problem studied in this paper. Sun, Lu, and Hong (2020) provided the first decentralized stochastic algorithm, D-GET, combining variance reduction and gradient tracking. Li, Li, and Chi (2022); Xin, Khan, and Kar (2022) further proposed algorithms with improved complexity bound. Recently, DEAREST (Luo and Ye 2022) is the first decentralized stochastic algorithm that achieves both optimal computation and communication complexity. Due to the assumption that the global objective is nonconvex, all these approaches can obtain at most sublinear convergence rates.

A comparison between our work and related works is summarized in Table 1.

### 2.2 Stochastic Sum-of-Nonconvex Optimization

Stochastic optimization on the sum-of-nonconvex optimization problem is a commonly used technique for analyzing offline Principle Component Analysis (PCA) problems. Garber et al. (2016) reduced 1-PCA subproblems to the sum-of-nonconvex problem, and they leveraged the conventional accelerated stochastic optimization scheme to accelerate the convergence. For the  $k$ -PCA problem, Allen-Zhu and Li (2016) reduced the  $k$ -PCA problem to the sum-of-nonconvex problem, and they apply the accelerated stochastic technique to improve the convergence of  $k$ -PCA problem. Allen-Zhu (2018) further improved the convergence by accelerating the stochastic optimization of the sum-of-nonconvex problem with the linear coupling technique (Allen-Zhu and Orecchia 2014).

Methods	Problem	Stochastic Gradient Calls	Communication Complexity
GT-SVRG (Xin, Khan, and Kar 2020)	Sum-of-Convex Non-Composite	$\mathcal{O}\left(\left(n + \frac{\kappa^2 \log \kappa}{(1-\lambda_2(W))^2}\right) \log \frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\left(n + \frac{\kappa^2 \log \kappa}{(1-\lambda_2(W))^2}\right) \log \frac{1}{\epsilon}\right)$
GT-SAGA (Xin, Khan, and Kar 2020)	Sum-of-Convex Non-Composite	$\mathcal{O}\left(\left(n + \frac{\kappa^2}{(1-\lambda_2(W))^2}\right) \log \frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\left(n + \frac{\kappa^2}{(1-\lambda_2(W))^2}\right) \log \frac{1}{\epsilon}\right)$
PG-EXTRA (Shi et al. 2015b)	Sum-of-Convex Composite	$\mathcal{O}\left(\frac{n\kappa}{1-\lambda_2(W)} \log \frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\frac{n\kappa}{1-\lambda_2(W)} \log \frac{1}{\epsilon}\right)$
NIDS (Li, Shi, and Yan 2019)	Sum-of-Convex Composite	$\mathcal{O}\left(\left(n\left(\kappa + \frac{1}{1-\lambda_2(W)}\right)\right) \log \frac{1}{\epsilon}\right)$	$\mathcal{O}\left(\left(\kappa + \frac{1}{1-\lambda_2(W)}\right) \log \frac{1}{\epsilon}\right)$
PMGT-SVRG (Ye, Xiong, and Zhang 2020)	Sum-of-Convex Composite	$\mathcal{O}\left((n + \kappa) \log \frac{1}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{n+\kappa}{\sqrt{1-\lambda_2(W)}} \log \frac{1}{\epsilon}\right)$
PMGT-SAGA (Ye, Xiong, and Zhang 2020)	Sum-of-Convex Composite	$\mathcal{O}\left((n + \kappa) \log \frac{1}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{n+\kappa}{\sqrt{1-\lambda_2(W)}} \log \frac{1}{\epsilon}\right)$
<b>PMGT-SVRG (This paper)</b>	Sum-of-Nonconvex Composite	$\mathcal{O}\left((n + \sqrt{n\kappa}) \log \frac{1}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{n}+\kappa}{\sqrt{1-\lambda_2(W)}} \log \frac{1}{\epsilon}\right)$
<b>PMGT-KatyushaX (This paper)</b>	Sum-of-Nonconvex Composite	$\mathcal{O}\left(\left(n + n^{\frac{3}{4}}\sqrt{\kappa}\right) \log \frac{1}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{n}+\sqrt{\kappa}n^{\frac{1}{4}}}{\sqrt{1-\lambda_2(W)}} \log \frac{1}{\epsilon}\right)$

Table 1: We compare the proposed algorithms with related work on Problem (1) when the global objective function is strongly convex. We use notation  $\tilde{\mathcal{O}}$  to hide the logarithm factor in the complexity. Note that the condition number in this table does not consider the difference in smoothness parameters. We present our results by distinguishing  $L$ ,  $\ell_1$ , and  $\ell_2$  in Sections 4 and 5.2.

### 3 Preliminaries

In this section, we formalize our problem setting.

#### 3.1 Notations

We denote  $\|\cdot\|$  as the Euclidean norm for vectors and Frobenius norm for matrices, and we denote  $\|\cdot\|_2$  as the operator norm for matrix. We use lowercase non-bold letter  $x \in \mathcal{R}^d$  as a random variable of dimension  $d$  and lowercase bold letter

$$\mathbf{x} = (x_1, \dots, x_m)^\top \in \mathcal{R}^{m \times d}$$

as the aggregated variable collected from  $m$  machines. We denote all one vector of dimension  $m$  by  $\mathbf{1} \in \mathcal{R}^m$ . For simplicity, we write  $\mathbf{1}x \in \mathcal{R}^{m \times d}$  as the Kronecker product between the all one vector  $\mathbf{1}$  and some vector  $x$ . We use  $\bar{x}$  as the average of the aggregated variable  $\mathbf{x}$  such that  $\bar{x} = m^{-1}\mathbf{1}^\top \mathbf{x}$ . For the non-smooth function  $\psi(\cdot)$ , we define  $\Psi(\mathbf{x}) = m^{-1} \sum_{i=1}^m \psi(x_i)$  for the aggregated variable  $\mathbf{x} \in \mathcal{R}^{m \times d}$ . We also define the proximal operator for vector  $x$  as

$$\text{prox}_{\eta, \psi}(\mathbf{x}) = \arg \min_{z \in \mathcal{R}^d} \left( \psi(z) + \|z - \mathbf{x}\|^2 / (2\eta) \right)$$

and the proximal operator for aggregated variable  $\mathbf{x}$  as

$$\text{prox}_{m\eta, \Psi}(\mathbf{x}) = \arg \min_{z \in \mathcal{R}^{m \times d}} \left( \Psi(z) + \|z - \mathbf{x}\|^2 / (2m\eta) \right).$$

We use  $x^*$  to represent the optimal solution for  $F(\cdot)$  as

$$x^* = \arg \min_{x \in \mathcal{R}^d} F(x).$$

#### 3.2 Problem Formulation

We summarize some of the basic properties of convex and smooth functions below.

**Definition 3.1.** For a function  $f : \mathcal{R}^d \rightarrow \mathcal{R}$ , there exist some constants  $L, \ell_1, \ell_2 > 0$  and  $\sigma \geq 0$  such that

1.  $f$  is  $\sigma$ -strongly convex. That is, for any  $x, y \in \mathcal{R}^d$ ,

$$f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \frac{\sigma}{2} \|x - y\|^2;$$

2.  $f$  is  $L$ -Lipschitz smooth. That is, for any  $x, y \in \mathcal{R}^d$ ,

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|;$$

3.  $f$  is  $(\ell_1, \ell_2)$ -smooth. That is, for any  $x, y \in \mathcal{R}^d$ ,

$$-\frac{\ell_2}{2} \|x - y\|^2 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{\ell_1}{2} \|x - y\|^2.$$

Recall that the global objective function (1) can be decomposed into  $mn$  nonconvex functions  $f_{i,j}(\cdot)$ . The  $i$ th agent is given access to a disjoint subset of  $n$  functions  $f_{i,j}(\cdot)$ , for  $\forall j \in [n]$ . We assume the function  $f(\cdot)$  is convex and  $L$ -smooth, each  $f_{i,j}(\cdot)$  is  $(\ell_1, \ell_2)$ -smooth with  $\ell_2 \geq \ell_1$  and  $\psi(\cdot)$  is a proper convex function. We further assume  $f$  is  $\sigma_f$ -strongly convex and  $\psi$  is  $\sigma_\psi$ -strongly convex with  $\sigma_f \geq 0$  and  $\sigma_\psi \geq 0$ . We define  $\sigma = \sigma_f + \sigma_\psi$  such that  $\sigma > 0$ .

We focus on decentralized optimization on a network in which each agent only communicates with its neighbors. The topology of the network is characterized by the gossip matrix  $W$ . We let  $W_{i,j} \geq 0$  if nodes  $i$  and  $j$  are direct neighbour in  $\mathcal{G}$ ; and  $W_{i,j} = 0$  if nodes  $i$  and  $j$  are not connected. Furthermore, we assume  $W$  is a doubly stochastic matrix, and it satisfies the following properties:

**Definition 3.2.** Let  $W$  be a doubly stochastic matrix. Then, (a)  $W$  is symmetric, (b)  $0 \preceq W \preceq I$  and  $W\mathbf{1} = \mathbf{1}$ , and (c)  $\text{null}(I - W) = \text{span}(\mathbf{1})$ .

### 4 Convergence Analysis of PMGT-SVRG

In this section, we show the convergence rate of the PMGT-SVRG (Ye, Xiong, and Zhang 2020) for the objective function (1). PMGT-SVRG achieves a linear convergence rate on

the sum-of-convex problem by integrating gradient tracking and multi-consensus mixing techniques into the SVRG algorithm. We remove the assumption that each function  $f_{i,j}(\cdot)$  is convex such that the inequality

$$\frac{1}{2L} \|\nabla f_{i,j}(x) - \nabla f_{i,j}(y)\|^2 \leq f_{i,j}(x) - f_{i,j}(y) - \langle \nabla f_{i,j}(y), x - y \rangle$$

for any  $x, y \in \mathbb{R}^d$  in Assumption 1 of (Ye, Xiong, and Zhang 2020) no longer holds. The following result shows that PMGT-SVRG can still achieve a linear convergence rate on the sum-of-nonconvex problem:

**Theorem 4.1.** *Assume function  $F(\cdot)$  defined in (1) is  $\sigma$ -strongly convex,  $f(\cdot)$  is  $L$ -smooth, and each component  $f_{i,j}$  is  $(\ell_1, \ell_2)$ -smooth. Additionally, we assume that the underlying network matrix  $W$  is doubly stochastic so it satisfies the properties in Definition 3.2. Under appropriate hyperparameter setting, to obtain  $F(\bar{y}^{k+1}) - F(x^*) \leq \epsilon$ , the algorithm PMGT-SVRG requires at most*

$$\mathcal{O} \left( \left( n + \frac{L + (\ell_1 \ell_2)^{\frac{1}{2}}}{\sigma} n^{\frac{1}{2}} \right) \log \frac{F(\bar{y}^0) - F(x^*)}{\epsilon} \right)$$

SFO calls and

$$\tilde{\mathcal{O}} \left( \frac{(\sqrt{n} + (L + \sqrt{\ell_1 \ell_2})/\sigma)}{(1 - \lambda_2(W))^{1/2}} \log \frac{F(\bar{y}^0) - F(x^*)}{\epsilon} \right)$$

rounds of communication.

**Remark 4.2.** Compared with the theoretical results of the PMGT-SVRG on the decentralized sum-of-convex problem in Table 1, the SFO complexity introduces an additional dependency on  $\sqrt{n}$ . It can be inferred that when the condition number  $\kappa := (L + \sqrt{\ell_1 \ell_2})/\sigma$  is larger than  $\sqrt{n}$ , the SFO complexity of PMGT-SVRG on the sum-of-nonconvex problem is worse than the sum-of-convex problem. Interestingly, the communication complexity of the PMGT-SVRG achieved by our analysis is better than that by Ye, Xiong, and Zhang (2020) even though our objective function is harder. The improvement comes from the introduction of the minibatch in Algorithm 3 in the appendix.

## 5 PMGT-KatyushaX

In this section, we propose the main idea behind PMGT-KatyushaX and present the convergence theorem of this algorithm.

### 5.1 The Algorithm

We present the main intuition of the PMGT-KatyushaX algorithm. The core design of the algorithm is to apply the acceleration scheme (Allen-Zhu and Orecchia 2014) on the stochastic variance reduced gradient (SVRG) (Johnson and Zhang 2013) method. Furthermore, we blend the powerful ideas of gradient tracking and multi-consensus mixing into the accelerated algorithm to develop the decentralized variant of the algorithm.

The backbone of our algorithm is the SVRG which adopts an outer-inner loop structure to reduce the inherent variance of stochastic gradients. Specifically, we construct a

full gradient snapshot  $\nabla f_i(w_i^0)$  for each agent  $i \in [m]$  at each epoch. At each iteration of the inner loop, a variance-reduced unbiased gradient estimator is updated as:

$$v_i^t = \nabla f_i(w_i^0) + b^{-1} \sum_{j_i \in B_i^t} (\nabla f_{i,j_i}(w_i^t) - \nabla f_{i,j_i}(w_i^0))$$

where  $B_i^t$  is a minibatch of  $b$  indices sampled uniformly from  $\{1, \dots, n\}$ . Recall that the global objective function (1) contains a convex, non-smooth function  $\psi(\cdot)$ , we apply the proximal mapping after executing one step of gradient descent:

$$w_i^{t+1} = \text{prox}_{\eta, \psi}(w_i^t - \eta v_i^t)$$

where  $\eta$  is the learning rate of the SVRG algorithm. The SVRG algorithm is known to achieve a linear rate of convergence for the strongly convex objective function.

**Acceleration** If we naively extend the SVRG algorithm to the decentralized setting, the convergence rate of the resulting algorithm has a linear dependency on the condition number as shown in Theorem 4.1. To obtain an improved dependency on the condition number, we employ the acceleration technique introduced by Allen-Zhu and Orecchia (2014). The acceleration is achieved by the linear coupling of the gradient descent step and mirror descent step. In particular, denote  $x_i^k := w_i^0$  and  $y_i^k := w_i^T$  as the first and the last iterate of the SVRG inner loop at Epoch  $k$ , then we apply one step of mirror descent as follows

$$q_i^k = \arg \min_{q \in \mathbb{R}^d} \left\{ \frac{1}{2} \|q - q_i^{k-1}\|^2 + \langle \frac{x_i^k - y_i^k}{2\tau}, q \rangle + \frac{\tau}{4} \|q - y_i^k\|^2 \right\},$$

which can be simplified as

$$q_i^k = \frac{q_i^{k-1} + \frac{\tau y_i^k}{2} - \frac{x_i^k - y_i^k}{2\tau}}{1 + \frac{\tau}{2}}.$$

The iterate  $x_i^{k+1}$  can be updated as a linear combination of  $y_i^k$  and  $q_i^k$ :

$$x_i^{k+1} = \tau q_i^k + (1 - \tau) y_i^k.$$

On top of the accelerated SVRG method, we also apply gradient tracking and multi-consensus mixing to extend the above algorithm to the decentralized setting.

**Gradient Tracking** Recall that our goal is to find the minima  $x^*$  of the global objective function (1). However, gradients collected from local neighbors have large variances due to the dissimilarity between distinct local objective functions  $f_i(\cdot)$ . To alleviate this issue, we adopt the gradient tracking technique that introduces a new variable  $s$  to track the difference between local gradient estimators:

$$s^{t+1} = \text{Mix}(s^t + v^{t+1} - v^t)$$

where  $\text{Mix}(\cdot)$  is some mixing protocol, and  $v^t$  is the aggregated variance-reduced gradient estimator of  $\{v_i^t\}_{i=1}^m$  at the  $t$ -th iteration of the inner loop. The intuition behind the technique is that while the variance of the local gradient estimators can be arbitrarily large in general, the variance of the differences between local gradient estimators will be small when the local variable approaches the global minima  $x^*$ . The technique also applies to the full gradient constructed in the outer loop.

**Algorithm 1: PMGT-KatyushaX**


---

**Input:**  $x_i^0, y_i^0, y_i^{-1}, q_i^0$  and  $S_i^0$   
**Parameter:** FastMix parameter  $M$ , functions  $\{f_i\}_{i=1}^m$  and  $\psi$ , initial point  $x^0$ , mini-batch size  $b$ , learning rate  $\eta \geq 0$ , momentum parameter  $\tau \in (0, 1]$ , number  $K$  of epochs  
**Output:**  $x_i^K, y_i^K, q_i^K$

---

- 1: Initialize  $x_i^0 = x^0, y_i^0 = x^0, y_i^{-1} = x^0, q_i^0 = x^0$  and  $\hat{s}_i^0 = \nabla F(x_0)$  for each  $i$  in parallel
- 2: **for**  $k = 0, \dots, K - 1$  **do**
- 3:  $x_i^{k+1} = \text{FastMix}(\tau \mathbf{q}^k + (1 - \tau) \mathbf{y}^k, M)_i$
- 4:  $\nu_i^{k+1} = \hat{s}_i^k + \nabla f_i(x_i^{k+1}) - \nabla f_i(x_i^k)$
- 5:  $\hat{s}_i^{k+1} = \text{FastMix}(\nu^{k+1}, M)_i$
- 6:  $s_i^{-1} = v_i^{-1} = \hat{s}_i^{k+1}$
- 7:  $\mu_i = \hat{s}_i^{k+1}, w_i^0 = x_i^{k+1}$
- 8:  $t_0 = \lceil n/b \rceil$  and sample  $T \sim \text{Geom}(1/t_0)$
- 9: **for**  $t = 0, \dots, T$  **do**
- 10: Let  $B_i^t$  be a batch of  $b$  indices sampled uniformly from  $[n]$  with replacement
- 11:  $v_i^t = \mu_i + b^{-1} \sum_{j_i \in B_i^t} (\nabla f_{i,j_i}^t(w_i^t) - \nabla f_{i,j_i}^t(w_i^0))$
- 12:  $s_i^t = \text{FastMix}(s^{t-1} + \mathbf{v}^t - \mathbf{v}^{t-1}, M)_i$
- 13:  $w_i^{t+1} = \text{FastMix}(\text{prox}_{m\eta, \Psi}(w^t - \eta s^t), M)_i$
- 14: **end for**
- 15:  $y_i^{k+1} = w_i^{T+1}$
- 16:  $q_i^{k+1} = \text{FastMix}(\frac{\mathbf{q}^k + \frac{\tau \mathbf{y}^{k+1} - \mathbf{x}^{k+1} - \mathbf{y}^{k+1}}{2\tau}}{1 + \frac{\tau}{2}}, M)_i$
- 17: **end for**

---

**Multi-Consensus Mixing** Under the decentralized learning scenario, each agent wishes to obtain the global average of their variables through communication with local neighbors. The agent can get a more faithful estimate of the global average through multiple rounds of local communication. Additionally, the mixing technique can be accelerated to achieve even faster consensus. The complete procedure of the Multi-Consensus Mixing can be found in Algo. 2. We apply the Mixing technique to both state variables and gradient tracking variables in both the inner loop and outer loop.

After fusing these techniques with the SVRG algorithm, the complete PMGT-KatyushaX for the decentralized setting is presented in Algo. 1.

## 5.2 Main Theorem

In this subsection, we characterize the linear convergence rate of PMGT-KatyushaX on the decentralized sum-of-nonconvex problems.

**Algorithm 2: FastMix( $\mathbf{x}^0, M, W$ )**


---

**Initilaize:**  $\mathbf{x}^{-1} = \mathbf{x}^0$  and  $\eta = 1/(1 + \sqrt{1 - \lambda_2^2(W)})$   
**Output:**  $W_K$

---

- 1: **for**  $k = 0, \dots, M$  **do**
- 2:  $\mathbf{x}^{k+1} = (1 + \eta) \mathbf{x}^k W - \eta \mathbf{x}^{k-1}$  (2)
- 3: **end for**

---

**Theorem 5.1.** Assume function  $F(\cdot)$  defined in (1) is  $\sigma$ -strongly convex,  $f(\cdot)$  is  $L$ -smooth, and each component  $f_{i,j}(\cdot)$  is  $(\ell_1, \ell_2)$ -smooth. We also assume that the underlying network matrix  $W$  is doubly stochastic so it satisfies the properties in Definition 3.2. Under appropriate parameter settings, the outputs of Algo. 1 has the following properties:

$$\mathbb{E}[F(\bar{y}^K) - F(x^*)] \leq \frac{3}{(1 + \tau/2)^K} (F(\bar{y}^0) - F(x^*)),$$

$$\mathbb{E}[\|\bar{x}^K - x^*\|^2] \leq \frac{128L}{\sigma(1 + \tau/2)^K} \|\bar{x}^0 - x^*\|^2,$$

and

$$\mathbb{E}[\|\bar{q}^K - x^*\|^2] \leq \frac{1750L}{\sigma\tau^4(1 + \tau/2)^K} \|\bar{x}^0 - x^*\|^2.$$

**Remark 5.2.** To obtain the  $\epsilon$ -approximate solution of the Problem (1), i.e.,  $F(\bar{y}^K) - F(x^*) \leq \epsilon$ , the outer loop of Algorithm PMGT-KatyushaX has to be executed for at least

$$K = \mathcal{O}\left(\left(1 + \frac{\sqrt{Lb}}{\sqrt{n\sigma}} + \frac{(\ell_1\ell_2)^{\frac{1}{4}}}{\sqrt{\sigma n^{\frac{1}{4}}}}\right) \log\left(\frac{F(\bar{y}^0) - F(x^*)}{\epsilon}\right)\right)$$

times. Recall that at each epoch, the algorithm makes one call of full gradient oracle and  $Tb$  SFO calls where  $\mathbb{E}[T] = n/b$ . Consequently, our algorithm makes an expectation of  $2n$  SFO calls. Additionally, as can be seen in Algo. 1, the multi-consensus mixing takes  $M$  rounds of communication when called. We can deduce that

$$M = \mathcal{O}\left(\log\left(\frac{L}{n\sigma} + \frac{\sqrt{\ell_1\ell_2}}{n\sigma}\right)\right).$$

To reach  $\epsilon$ -accuracy, Algo. 1 has to make  $\mathcal{O}(Kn)$  calls to the first-order oracle and  $\mathcal{O}(KnM/b)$  rounds of communication.

We can bound the computation complexity and communication complexity by setting batch size  $b = \sqrt{n}$  and the following corollary can be obtained:

**Corollary 5.3.** Under the setting of Theorem 5.1, to obtain an  $\epsilon$ -approximate solution  $\bar{y}^K$ , i.e.,  $F(\bar{y}^K) - F(x^*) \leq \epsilon$ , PMGT-KatyushaX requires at most

$$\mathcal{O}\left(\left(n + \frac{L^{\frac{1}{2}} + (\ell_1\ell_2)^{\frac{1}{4}}}{\sqrt{\sigma}} n^{\frac{3}{4}}\right) \log\left(\frac{F(\bar{y}^0) - F(x^*)}{\epsilon}\right)\right)$$

SFO calls and

$$\tilde{\mathcal{O}}\left(\left(\frac{\sqrt{n} + \frac{L^{\frac{1}{2}} + (\ell_1\ell_2)^{\frac{1}{4}}}{\sqrt{\sigma}} n^{\frac{1}{4}}}{\sqrt{1 - \lambda_2(W)}}\right) \log\left(\frac{F(\bar{y}^0) - F(x^*)}{\epsilon}\right)\right)$$

rounds of communication.

## 6 Proof Sketch of Theorem 5.1

In this section, we provide a sketch of the proof for Theorem 5.1. Since Algo. 1 has a double loop structure, we prove the theorem in two stages. For the first stage, we present the analysis for one epoch of decentralized SVRG adapted for

the sum-of-nonconvex problem; For the second stage, our goal is to show that the consensus error and the convergence error of the outer loop decay linearly at the same rate.

We first define the vector of consensus errors in the inner loop as  $z^t = (\|\mathbf{w}^t - \mathbf{1}\bar{w}^t\|^2, \|\mathbf{s}^t - \mathbf{1}\bar{s}^t\|^2)^\top$ . We can build a system of linear inequalities with  $z^t$  such that the spectral norm of the coefficient matrix is less than 1 if the hyperparameters are chosen appropriately. We start by reviewing the following lemma which is essential for the analysis of the consensus error:

**Lemma 6.1** (Liu and Morse (2011)). *Let  $\mathbf{w}_0, \mathbf{w}_M \in \mathcal{R}^d$  be the input and output of the Algorithm 2 and  $\bar{w} = \frac{1}{m}\mathbf{1}^\top \mathbf{w}_0$ , then it satisfies that*

$$\|\mathbf{w}_M - \mathbf{1}\bar{w}\| \leq \rho \|\mathbf{w}_0 - \mathbf{1}\bar{w}\|,$$

where  $\rho = \sqrt{14} \left(1 - \left(1 - \frac{1}{\sqrt{2}}\right) \sqrt{1 - \lambda_2(L)}\right)^M$ .

After constructing the linear system of inequalities for  $z^t$ , we obtain the following lemma that bounds the consensus error accumulated in the inner loop:

**Lemma 6.2.** *Given  $T \sim \text{Geom}(p)$ , the expected consensus error at the end of the inner loop satisfies*

$$\mathbb{E}_T [\|z^T\|] \leq \|z^0\| + \frac{1-p}{p} 16\rho^2 \eta^2 \ell_1 \ell_2 \|\mathbf{w}^0 - \mathbf{1}\bar{w}^0\|^2 + \frac{2-p}{p} 16\rho^2 \eta^2 \ell_1 \ell_2 m \mathbb{E}_T [\|\bar{w}^T - \bar{w}^0\|^2].$$

Using the convexity and smoothness assumption, we can obtain the following result.

**Lemma 6.3.** *For any iteration  $t$  in the inner loop, the average variable is defined as  $\bar{w}^t = m^{-1}\mathbf{1}^\top \mathbf{w}^t$  where  $\mathbf{w}^{t+1} = \text{FastMix}(\text{prox}_{m\eta, \Psi}(\mathbf{w}^t - \eta \mathbf{s}^t), M)$ . For any  $u \in \mathbb{R}^d$ ,*

$$\begin{aligned} & \mathbb{E} [F(\bar{w}^{t+1}) - F(u)] \\ & \leq \mathbb{E} \left[ \frac{\eta}{1-\eta L} \|\bar{s}^t - \nabla f(\bar{w}^t)\|^2 + \frac{2-\eta\sigma_f}{4\eta} \|u - \bar{w}^t\|^2 \right. \\ & \quad - \frac{(1+\sigma_\psi)\eta \|u - \bar{w}^{t+1}\|^2}{2\eta} + \frac{1-\eta L + 2\eta}{2(1-\eta L)m} \|\mathbf{1}\bar{s}^t - \mathbf{s}^t\|^2 \\ & \quad \left. + \left( \frac{\ell_1 \ell_2}{\sigma_f m} + \frac{1+\eta}{2m\eta} \right) \|\mathbf{w}^t - \mathbf{1}\bar{w}^t\|^2 \right]. \end{aligned}$$

Moreover, one can show the following result by the  $(\ell_1, \ell_2)$ -smoothness of  $f_{i,j}$ :

**Lemma 6.4.** *Denote the variable  $\bar{s}^t = m^{-1} \sum_{i=1}^m \mathbf{s}_i^t$  with  $\mathbf{s}_i^t = \text{FastMix}(\mathbf{s}^{t-1} + \mathbf{v}^t - \mathbf{v}^{t-1}, M)_i$ . Then,*

$$\begin{aligned} & \mathbb{E} [\|\bar{s}^t - \nabla f(\bar{w}^t)\|^2] \\ & \leq \left( \frac{6\ell_1 \ell_2}{mb} + \frac{2\ell_1 \ell_2}{m} \right) \|\mathbf{w}^t - \mathbf{1}\bar{w}^t\|^2 \\ & \quad + \frac{6\ell_1 \ell_2}{b} \|\bar{w}^t - \bar{w}^0\|^2 + \frac{6\ell_1 \ell_2}{mb} \|\mathbf{w}^0 - \mathbf{1}\bar{w}^0\|^2. \end{aligned}$$

Combining the above three lemmas, we can prove the following main result for the inner loop:

**Lemma 6.5.** *If we choose the hyperparameters for Algorithm 1 such that  $\eta \leq \min \left( 1/(2L), \sqrt{b/(\ell_1 \ell_2 t_0)}/8 \right)$ , and*

$\rho \leq \min \left( (8bm\eta J t_0)^{-1/2}, (18(1+b))^{-1/2} \right)$  where  $J := J(\eta, b)$  is some positive constant. Then, for any  $u \in \mathbb{R}^d$ ,

$$\begin{aligned} & \mathbb{E} [F(\bar{w}^{T+1}) - F(u)] \\ & \leq \mathbb{E} \left[ -\frac{\|\bar{w}^{T+1} - \bar{w}^0\|^2}{4\eta t_0} + \frac{\langle \bar{w}^0 - u, \bar{w}^0 - \bar{w}^{T+1} \rangle}{t_0 \eta} \right. \\ & \quad \left. - \frac{\sigma_f + \sigma_\psi}{8} \|u - \bar{w}^{T+1}\|^2 + J\eta^2 \|\mathbf{s}^0 - \mathbf{1}\bar{s}^0\|^2 \right. \\ & \quad \left. + \left( J + \frac{1}{4\eta m} \right) \|\mathbf{w}^0 - \mathbf{1}\bar{w}^0\|^2 \right]. \end{aligned}$$

For the theoretical analysis of the outer loop, we define the following vector of consensus errors

$$(\|\mathbf{x}^t - \mathbf{1}\bar{x}^t\|^2, \|\mathbf{y}^t - \mathbf{1}\bar{y}^t\|^2, \|\mathbf{q}^t - \mathbf{1}\bar{q}^t\|^2, \eta^2 \|\mathbf{s}^t - \mathbf{1}\bar{s}^t\|^2)^\top.$$

We also construct a system of linear inequalities of the above quantity. Using a similar proof technique as the inner loop to bound the consensus error of the outer loop, we can prove Theorem 5.1 by induction. The complete theoretical analysis of PMGT-KatyushaX is presented in the Appendix.

## 7 Numerical Experiments

To demonstrate the efficiency of PMGT-KatyushaX, we evaluate the proposed method on the sub-problem of solving PCA by the shift-and-invert method. The corresponding sum-of-nonconvex optimizing problem has the form of

$$\min_{x \in \mathcal{R}^d} F(x) = \frac{1}{2} x^\top \left( \left( \lambda_1 + \frac{\lambda_1 - \lambda_2}{r} \right) I - A \right) x + b^\top x \quad (3)$$

where  $A = (mn)^{-1} \sum_{i=1}^m \sum_{j=1}^n a_{i,j} a_{i,j}^\top \in \mathbb{R}^{d \times d}$ ,  $\lambda_1$  and  $\lambda_2$  are the largest and the second largest eigenvalues of the matrix  $A$ ;  $r$  is a hyperparameter that controls the ratio for the eigengap.

We conduct our experiments on both synthetic and real-world datasets. For the synthetic dataset, we generate a Bernoulli matrix of size  $60,000 \times 50$  with entries in  $\{\pm 1\}$ ; For the real-world dataset, we use the Covtype downloaded from the LIBSVM website<sup>2</sup>. For the gossip matrix  $W$  underlying the decentralized network, we generate a matrix where each agent is randomly connected to two neighbors. The second largest eigenvalue of the resulting matrix is  $\lambda_2(W) = 0.97$ .

**Baselines** We compare the empirical performance of our proposed method with several baselines including PMGT-SVRG (Ye, Xiong, and Zhang 2020), PGEXTRA (Shi et al. 2015b), and NIDS (Li, Shi, and Yan 2019). Although theoretical results of PGEXTRA and NIDS are developed for the sum-of-convex objective functions, they still show convergence behavior on the sum-of-nonconvex functions empirically.

**Experiment Specifications** For all of the experiments, the y-axis represents the suboptimality of function value, i.e.,  $F(\bar{y}) - F(x^*)$  where  $F(x^*)$  is taken as the lowest value achieved among all the baselines. The left and the middle figure represents the suboptimality vs. gradient evaluations and communication rounds, respectively. The right

<sup>2</sup><https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets>

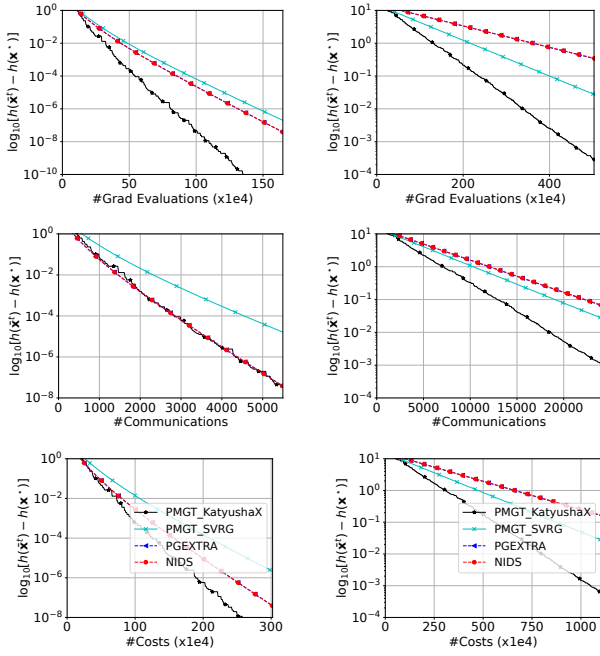


Figure 1: Performance comparison between PGEXTRA, NIDS, PMGT-SVRG, and PMGT-KatyushaX on the synthetic dataset. The left column represents results with the ratio  $r = 2$  and the right column represents results with the ratio  $r = 300$  defined in Problem (3). The plot of PGEXTRA and NIDS are overlapped as their performance are close to each other.

figure shows the suboptimality vs. the cost which is the weighted average between gradient evaluation and communication rounds.

**Comparison Results on the Synthetic Dataset** Figure 1 reports the performance comparison between our PMGT-KatyushaX and baselines when the ratio  $r = 2$  or  $r = 300$ . PMGT-KatyushaX outperforms other baselines for both settings, even when the ratio is small. We point out that PGEXTRA and NIDS have similar performance so their curves are overlapped in the performance comparison. For each setting, PMGT-KatyushaX makes the least number of gradient evaluations compared with other baselines. To our surprise, it even requires fewer communication rounds to converge than NIDS although the communication bound of PMGT-KatyushaX depends on the number of component functions  $\sqrt{n}$ . As a result, PMGT-KatyushaX achieves the best cost among all baselines.

**Comparison Results on the Real-world Dataset** Figure 2 reports the performance comparison between PMGT-KatyushaX and other baselines on the Covtype dataset when the ratio  $r = 2$  or  $r = 300$ . When  $r$  is small, PMGT-KatyushaX has fewer gradient evaluations than NIDS/PGEXTRA, but PMGT-KatyushaX requires more communication rounds than NIDS/PGEXTRA. When  $r$  is large, PMGT-KatyushaX outperforms other baselines in both the

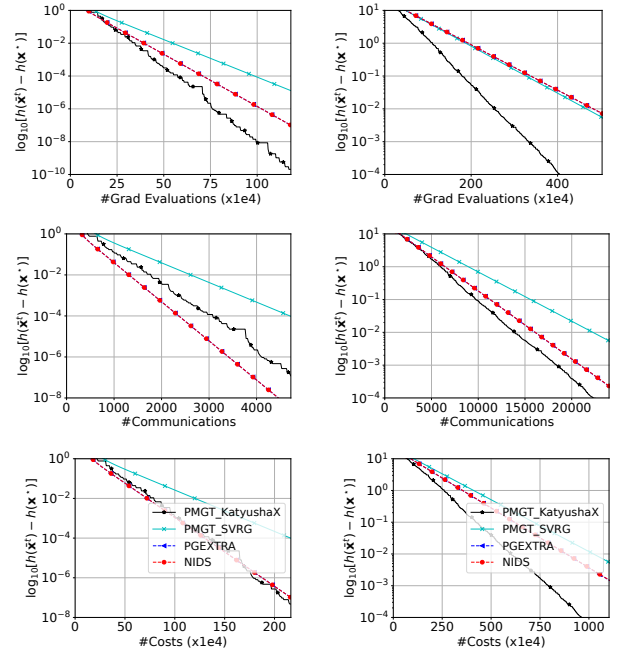


Figure 2: Performance comparison between PGEXTRA, NIDS, PMGT-SVRG, and PMGT-KatyushaX on the Covtype dataset. The left column represents results with the ratio  $r = 2$  and the right column represents results with the ratio  $r = 300$  defined in (3). The plot of PGEXTRA and NIDS are overlapped as their performance are close to each other.

number of gradient evaluations and communication rounds.

## 8 Conclusions and Future Work

This paper presented a new theoretical analysis of PMGT-SVRG to the decentralized sum-of-nonconvex problem, and it has a linear dependency on the condition number. To achieve a better dependency, we proposed the first accelerated stochastic first-order algorithm for the decentralized sum-of-nonconvex problem and showed it enjoys a linear convergence rate with a square-root dependency on the condition number. The empirical evidence validates the advantages of our algorithm on both synthetic and real-world datasets.

Although the computational complexity of the proposed algorithm has matched the best-known algorithm for the centralized scenario, the communication upper bound still looks unsatisfying due to the inclusion of the factor  $\sqrt{n}$ . It is interesting to study how to design a more communication-efficient algorithm for decentralized sum-of-nonconvex optimization.

## Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2023-08-043T-J). This research is part of the programme DesCartes and is supported

by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

## References

- Agarwal, N.; Allen-Zhu, Z.; Bullins, B.; Hazan, E.; and Ma, T. 2016. Finding approximate local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*.
- Alghunaim, S.; Yuan, K.; and Sayed, A. H. 2019. A linearly convergent proximal gradient algorithm for decentralized optimization. *Advances in Neural Information Processing Systems*, 32.
- Allen-Zhu, Z. 2018. Katyusha X: Practical momentum method for stochastic sum-of-nonconvex optimization. *arXiv preprint arXiv:1802.03866*.
- Allen-Zhu, Z.; and Li, Y. 2016. LazySVD: Even faster SVD decomposition yet without agonizing pain. *Advances in Neural Information Processing Systems*, 29.
- Allen-Zhu, Z.; and Li, Y. 2017. Follow the compressed leader: Faster online learning of eigenvectors and faster MMWU. In *International Conference on Machine Learning*, 116–125. PMLR.
- Allen-Zhu, Z.; and Orecchia, L. 2014. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*.
- Carmon, Y.; Duchi, J. C.; Hinder, O.; and Sidford, A. 2018. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2): 1751–1772.
- Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems*, 27.
- Di Lorenzo, P.; and Scutari, G. 2016. Next: In-network non-convex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2): 120–136.
- Garber, D.; Hazan, E.; Jin, C.; Kakade, S. M.; Musco, C.; Netrapalli, P.; and Sidford, A. 2016. Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation. In *International Conference on Machine Learning*.
- Hendriks, H.; Bach, F.; and Massoulié, L. 2019. An accelerated decentralized stochastic proximal algorithm for finite sums. *Advances in Neural Information Processing Systems*, 32.
- Hendriks, H.; Bach, F.; and Massoulié, L. 2020. Dual-free stochastic decentralized optimization with variance reduction. *Advances in Neural Information Processing Systems*, 33: 19455–19466.
- Hendriks, H.; Bach, F.; and Massoulié, L. 2021. An optimal algorithm for decentralized finite-sum optimization. *SIAM Journal on Optimization*, 31(4): 2753–2783.
- Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26.
- Li, B.; Cen, S.; Chen, Y.; and Chi, Y. 2020. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. In *International Conference on Artificial Intelligence and Statistics*, 1662–1672. PMLR.
- Li, B.; Li, Z.; and Chi, Y. 2022. DESTRESS: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization. *SIAM Journal on Mathematics of Data Science*, 4(3): 1031–1051.
- Li, M.; Andersen, D. G.; Park, J. W.; Smola, A. J.; Ahmed, A.; Josifovski, V.; Long, J.; Shekita, E. J.; and Su, B.-Y. 2014. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*, 583–598.
- Li, Z.; Shi, W.; and Yan, M. 2019. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17): 4494–4506.
- 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. *Advances in Neural Information Processing Systems*, 30.
- Liu, J.; and Morse, A. S. 2011. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35(2): 160–165.
- Luo, L.; and Ye, H. 2022. An Optimal Stochastic Algorithm for Decentralized Nonconvex Finite-sum Optimization. *arXiv preprint arXiv:2210.13931*.
- Mokhtari, A.; and Ribeiro, A. 2016. DSA: Decentralized double stochastic averaging gradient algorithm. *The Journal of Machine Learning Research*, 17(1): 2165–2199.
- Qu, G.; and Li, N. 2017. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3): 1245–1260.
- Saad, Y. 2011. *Numerical methods for large eigenvalue problems: revised edition*. SIAM.
- Shen, Z.; Mokhtari, A.; Zhou, T.; Zhao, P.; and Qian, H. 2018. Towards more efficient stochastic decentralized learning: Faster convergence and sparse communication. In *International Conference on Machine Learning*, 4624–4633. PMLR.
- Shi, W.; Ling, Q.; Wu, G.; and Yin, W. 2015a. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2): 944–966.
- Shi, W.; Ling, Q.; Wu, G.; and Yin, W. 2015b. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22): 6013–6023.
- Sun, H.; Lu, S.; and Hong, M. 2020. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, 9217–9228. PMLR.
- Sun, Y.; Daneshmand, A.; and Scutari, G. 2019. Convergence rate of distributed optimization algorithms based on gradient tracking. *arXiv preprint arXiv:1905.02637*.



- Xin, R.; Khan, U. A.; and Kar, S. 2020. Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68: 6255–6271.
- Xin, R.; Khan, U. A.; and Kar, S. 2022. Fast decentralized nonconvex finite-sum optimization with recursive variance reduction. *SIAM Journal on Optimization*, 32(1): 1–28.
- Ye, H.; Xiong, W.; and Zhang, T. 2020. PMGT-VR: A decentralized proximal-gradient algorithmic framework with variance reduction. *arXiv preprint arXiv:2012.15010*.
- Yuan, K.; Ying, B.; Zhao, X.; and Sayed, A. H. 2018. Exact diffusion for distributed optimization and learning-Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3): 708–723.