ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447202

ON IMPROVED DISTRIBUTED RANDOM RESHUFFLING OVER NETWORKS

Pranay Sharma⋆ *Jiarui Li*⋆ *Gauri Joshi*⋆

⋆Carnegie Mellon University, Pittsburgh, USA, {pranaysh, jiarui3, gaurij}@andrew.cmu.edu

ABSTRACT

In this paper, we consider a distributed optimization problem. A network of n agents, each with its own local loss function, aims to collaboratively minimize the global average loss. We prove improved convergence results for two recently proposed random reshuffling (RR) based algorithms, D-RR and GT-RR, for smooth strongly-convex and nonconvex problems, respectively. In particular, we prove an additional speedup with increasing n in both cases. Our experiments show that these methods can provide further communication savings by carrying multiple gradient steps between successive communications while also outperforming decentralized SGD. Our experiments also reveal a gap in the theoretical understanding of these methods in the nonconvex case.

Index Terms— distributed optimization, gradient tracking, random reshuffling, stochastic gradient methods

1. INTRODUCTION

This paper considers the problem of collaboratively optimizing the average of n *local* cost functions, each owned by individual *agents* connected in a network. The local function at each agent depends on the corresponding local dataset. Mathematically, the problem is as follows:

$$\min_{x\in\mathbb{R}^d} f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} f_{i,j}(x)}_{f_i(x)}, \tag{1}$$

where n is the number of agents, f_i is the local loss of agent $i \in [n] \triangleq \{1, 2, \ldots, n\}$, m is the local dataset size, and $f_{i,j}$ is the loss corresponding to the j-th sample at the i-th agent. This problem has applications in signal processing and machine learning (ML), and has been studied for decades [1, 2, 3]. However, modern applications also face an explosion in the amount of data available at the edge devices. This additional challenge precludes the usage of classical algorithms like gradient descent [4], that require full gradient computation at each step. In this situation, distributed stochastic gradient methods emerge as simple yet powerful alternatives.

Stochastic gradient descent (SGD) is one of the most popular methods in modern ML. Consequently, its decentralized versions have also been extensively studied in the literature [5, 6]. When minimizing smooth objective functions, decentralized SGD (D-SGD) *eventually* achieves the same convergence as that of centralized SGD, implying *network independence* of the convergence error [7]. Subsequent work has focused on proposing more sophisticated algorithms to further improve the performance of D-SGD. Such methods include gradient tracking (GT) [8] and exact diffusion (ED) [9, 10].

The theoretical analysis of vanilla SGD assumes *with-replacement* sampling at each step to compute the gradient estimate [11]. However, in practice, *without replacement* sampling is observed to perform better. A commonly used SGD variant, called *Random Reshuffling* (RR), is used in deep-learning packages like PyTorch and TensorFlow. RR permutes the dataset at the beginning of each epoch, and computes gradient estimates using mini-batches sampled from the permuted sequence. Recent work [12, 13] has theoretically shown the benefits of RR compared to SGD. Specifically, given the dataset size N[1] and number of epochs T, for smooth strongly-convex problems RR achieves (for large enough T) a convergence of $\mathcal{O}(1/(NT^2))$, compared to $\mathcal{O}(1/(NT))$ for SGD. For smooth nonconvex objectives, RR achieves $\mathcal{O}(1/(NT^2)^{1/3})$, compared to $\mathcal{O}(1/\sqrt{NT})$ for SGD. These benefits of RR over SGD naturally call for exploring RR in the decentralized setting.

The initial works that studied RR in the decentralized setting [14, 15] do not show any superiority over decentralized SGD. Subsequent work in [16, 17] proposed decentralized algorithms to solve smooth strongly-convex and nonconvex problems. Ignoring network dependence terms, the achieved convergence rates are $\mathcal{O}(1/(mT^2))$ in the strongly-convex, and $\mathcal{O}(1/(mT^2)^{1/3})$ in the nonconvex case. These outperform D-SGD in certain parameter regimes (see Table 1). However, two crucial questions remain unanswered.

1. Can decentralized RR methods (D-RR and GT-RR) achieve convergence speedup with increasing network size n?

2. How to design RR-based algorithms that achieve network independent asymptotic convergence?

This work was supported in part by NSF grants CCF 2045694, CNS-2112471, CPS-2111751, SHF-2107024 and ONR N00014-23-1-2149.

[1]These results are in the centralized setting. Comparing with the distributed setting in (1), $N = mn$.

In this paper, we answer the first question in the affirmative, and highlight some interesting empirical observations that might help answering the second question.

Table 1: Comparison of the convergence rates of different decentralized algorithms. We omit the higher-order terms. $\mathcal{O}_\lambda$ indicates that the dominant terms have network dependence that has been omitted for simplicity.

Work	Strongly Convex	Nonconvex
D-SGD	$\mathcal{O}\left(\frac{1}{mnT}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{mnT}}\right)$
D-RR [16]	$\mathcal{O}_\lambda\left(\frac{1}{mT^2}\right)$	$\mathcal{O}_\lambda\left(\frac{1}{T^{2/3}}\right)$
GT-RR [17]	$\mathcal{O}_\lambda\left(\frac{1}{mT^2}\right)$	$\mathcal{O}_\lambda\left(\frac{1}{(mT^2)^{1/3}}\right)$
Our work	With D-RR $\mathcal{O}_\lambda\left(\frac{\frac{1}{n}+\frac{1}{m}}{mT^2}\right)$	With GT-RR $\mathcal{O}_\lambda\left(\frac{1}{(mnT^2)^{1/3}}+\frac{1}{(mT)^{2/3}}\right)$

Contributions

We perform a refined analysis of two existing decentralized random reshuffling based algorithms: D-RR [16] for smooth strongly-convex, and GT-RR [17] for smooth nonconvex problems. Compared to existing results in [16, 17], we achieve an additional speedup in terms of the number of agents n (see Table 1) in both the cases.

Further, our experiments reveal some interesting observations. First, D-RR (and GT-RR) achieves a smaller error than D-SGD even if the agents in the former communicate significantly less often, with agents taking multiple stochastic gradient steps between successive communications. Second, in the nonconvex case, we discover a gap in the existing theoretical analysis and the empirical results. The convergence results in this case (in [16, 17], as well as ours) bound $\|\nabla f(\bar{x})\|^2$, where $\bar{x}$ is the global average of iterates, and have network dependence (see [17, Table 1] and Theorem 2). However, our experiments in Figure 2 suggest that $\|\nabla f(\bar{x})\|^2$ is independent of the network across different topologies. On the other hand, the convergence of $\frac{1}{n}\sum_{i=1}^n \|\nabla f(x_i)\|^2$ (often bounded in the analysis of D-SGD and related methods) does indeed depend on the underlying network.

2. ALGORITHM AND THEORETICAL RESULTS

We reproduce below the D-RR algorithm from [16], and refer the reader to [17] for the GT-RR algorithm. Next, we introduce the assumptions needed in our theoretical results.

2.1. Assumptions

We assume the agents in the network are connected via a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = [n]$ denotes the set of agents, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of edges. We denote the set of neighbors of agent i by $\mathcal{N}_i = \{j \in \mathcal{N} : (i,j) \in \mathcal{E}\}$. The edges of $\mathcal{G}$ have associated weights $W = [w_{ij}] \in \mathbb{R}^{n\times n}$.

Algorithm 1 Distributed Random Reshuffling (D-RR) [16]

Input: initialization $x_{i,0}$ for agents $i \in [n]$, weight matrix $W = [w_{ij}] \in \mathbb{R}^{n\times n}$, step-size sequence $\{\alpha_t\}$
for Epoch $t = 0, 1, \ldots T-1$ **do**
  for Agent $i \in [n]$ in parallel **do**
    Independently sample permutation $\{\pi_0^i, \ldots, \pi_{m-1}^i\}$ of $[m]$
    Set $x_{i,t}^0 = x_{i,t}$
    for $j = 0, \ldots, m-1$ **do**
      Update $x_{i,t}^{j+\frac{1}{2}} = x_{i,t}^j - \alpha_t \nabla f_{i,\pi_j^i}(x_{i,t}^j)$
      Send $x_{i,t}^{j+\frac{1}{2}}$ to neighbors $k \in \mathcal{N}_i$. Receive $x_{k,t}^{j+\frac{1}{2}}$ from neighbors $k \in \mathcal{N}_i$.
      Update $x_{i,t}^{j+1} = \sum_{k\in\mathcal{N}_i} w_{ik} x_{k,t}^{j+\frac{1}{2}}$
    end for
    Set $x_{i,t+1} = x_{i,t}^m$
  end for
end for
Output $\{x_{i,T}\}$

Assumption 1 (Network Weight Matrix)**.** The network graph $\mathcal{G}$ is undirected and connected, i.e., there exists a path between any two nodes in $\mathcal{G}$. There is a direct link between nodes i and j $(i \neq j)$ if and only if $w_{ij} > 0$ and $w_{ji} > 0$; otherwise, $w_{ij} = w_{ji} = 0$. The mixing matrix is nonnegative, symmetric, and stochastic, i.e., $W = W^\top$ and $W\mathbf{1} = \mathbf{1}$.

We denote by λ the spectral norm of matrix $W - \mathbf{1}\mathbf{1}^\top/n$. By Assumption 1, $\lambda < 1$. Next, we discuss the assumptions on the loss functions in (1).

Assumption 2 (Smoothness)**.** Each local component function $f_{i,j}$ is bounded from below and L-smooth, i.e.,

$$\|\nabla f_{i,j}(x) - \nabla f_{i,j}(y)\| \leq L\|x-y\|, \text{ for all } x, y \in \mathbb{R}^d.$$

Assumption 3 (Strong Convexity)**.** Each local function $f_{i,j}$ is μ-strongly convex, i.e., for all i, j and $x, y \in \mathbb{R}^d$

$$\langle \nabla f_{i,j}(x) - \nabla f_{i,j}(y), x - y\rangle \geq \mu\|x-y\|^2.$$

The above assumption can be relaxed to PL condition on the global function f [17], but we retain it here for simplicity. Next, we state and discuss our convergence results.

2.2. Convergence Results

Theorem 1 (Strongly Convex Case)**.** *Suppose Assumptions 1, 2 and 3 hold. If we choose $\alpha_t = \frac{\theta}{m\mu(t+K)}$ with $\theta > 12$ and appropriately chosen K, we have*

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left\|x_{i,T}^0 - x^*\right\|^2\right] \leq \mathcal{O}\left(\frac{1}{(1-\lambda)mnT^2}\right) + \mathcal{O}\left(\frac{1}{(1-\lambda)^3 m^2 T^2}\right) + \textit{higher order terms} \tag{2}$$

Remark 1. Theorem 1 improves the bound $\mathcal{O}\left(\frac{1}{(1-\lambda)^3 m^2 T^2}\right)$ in [16, Theorem 1], with an addition speedup with respect to the number of agents n in the first term and improving the dependence on the number of component functions m in the second term. In settings with $n < m$ and well-connected underlying graphs (such that $\frac{1}{1-\lambda}$ is independent of n), the first term dominates, and the convergence becomes *almost* network independent, recovering the performance of centralized RR (C-RR) [18]. For example, in Figure 1c, with exponential graphs, both D-RR and C-RR show similar convergence.

The improved bound in Theorem 1 results from a tighter bound on the shuffling variance $\sigma^2_{\text{shuffle}}$, defined in [16, Definition 1], which we state next.

Lemma 2.1. *Under Assumption 2, we have*

$$\sigma^2_{\textit{shuffle}} \leq \frac{\alpha_t^2 L m}{4n} \tilde{\sigma}_*^2, \tag{3}$$

where $\tilde{\sigma}_*^2 \triangleq \frac{1}{nm}\sum_{i=1}^n \sum_{j=1}^m \left\|\nabla f_{i,j}(\mathbf{x}^*) - \nabla f_i(\mathbf{x}^*)\right\|^2$.

Remark 2. The bound in (3) has an extra $1/n$ factor that is missing in the corresponding bound in [16, Lemma 6], and results in the linear speedup in the number of agents n in Theorem 1. Note that $\tilde{\sigma}_*^2$ defined in Lemma 2.1 is different from σ_*^2 in [16], where $\sigma_*^2 \triangleq \frac{1}{nm}\sum_{i=1}^n \sum_{j=1}^m \left\|\nabla f_{i,j}(\mathbf{x}^*)\right\|^2$. However, it is easy to see that $\tilde{\sigma}_*^2 \leq 4\sigma_*^2$.

Next, we state the convergence of GT-RR, a gradient tracking-based method proposed in [17], for smooth nonconvex problems. It is proved in [17] that GT-RR has better network dependence compared to D-RR.

Theorem 2 (Nonconvex Case)**.** *Suppose Assumptions 1, 2, and [17, Assumption 2.1] hold. Then, with appropriately chosen step-sizes* $\{\alpha_t\}$*, iterates generated by GT-RR satisfy*

$$\min_{t=0,1,\ldots,T-1} \mathbb{E}\left\|\nabla f(\bar{x}_t^0)\right\|^2 \leq \mathcal{O}\left(\frac{1}{((1-\lambda)mnT^2)^{1/3}}\right) + \mathcal{O}\left(\frac{1}{((1-\lambda)m^2T^2)^{1/3}}\right) + \textit{higher order terms}, \tag{4}$$

where $\bar{x}_t^0 = \frac{1}{n}\sum_{i=1}^n x_{i,t}^0$.

Remark 3. Our bound above improves the corresponding bound $\mathcal{O}\left(\frac{1}{(mT^2(1-\lambda^2))^{1/3}}\right)$ in [17, Theorem 4.1], by again achieving an addition speedup with respect to the number of agents n in the first term, and improving the dependence on the number of component functions m in the second term.

Remark 4. We state the result in Theorem 2 in terms of $\mathbb{E}\left\|\nabla f(\bar{x}_t^0)\right\|^2$ for direct comparison with the corresponding results in [16, 17]. However, there are two limitations of the current bound. First, the bound does not explicitly quantify how different the individual iterates $\{x_{i,t}^0\}_{i=1}^n$ are. Therefore, many existing works [6, 10] bound $\frac{1}{n}\sum_{i=1}^n \left\|\nabla f(x_{i,t}^0)\right\|^2$. To see the benefit of the latter quantity, note that

$$\frac{1}{n}\sum_{i=1}^n \left\|\nabla f(x_{i,t}^0)\right\|^2 \leq 2\left\|\nabla f(\bar{x}_t^0)\right\|^2 + \frac{2L^2}{n}\sum_{i=1}^n \left\|x_{i,t}^0 - \bar{x}_t^0\right\|^2,$$

which follows from Assumption 2. Second, our experiments suggest (see Figure 2 and the accompanying discussion) that $\left\|\nabla f(\bar{x}_t^0)\right\|^2$ for both GT-RR and D-RR is independent of the network topology, and matches the performance of centralized RR. This suggests that the $\frac{1}{1-\lambda}$ factor in (4) might be an artifact of the analysis. On the other hand, Figure 2 shows that $\left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(x_{i,t}^0)\right\|^2$ is indeed network dependent.

3. EXPERIMENT RESULTS

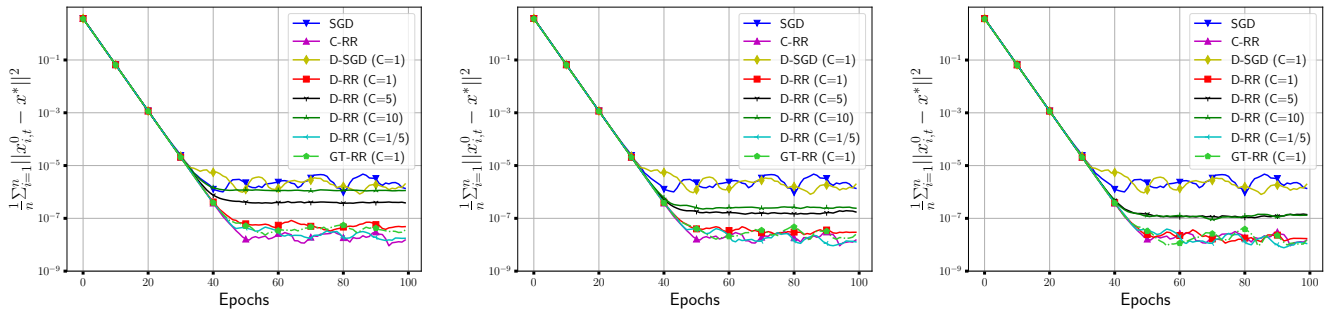
We evaluate the numerical performance of D-RR and GT-RR algorithms on the same binary classification problems on CIFAR10 dataset [19], as those considered in [17], but uncover some interesting observations. Centralized-RR and SGD serve as the baselines in all our experiments. All the curves have values averaged over 5 independent trials. The per agent batch-size for decentralized algorithms is 10 (for the centralized algorithms, it is $10 \times n$). In the strongly-convex case, we solve the following optimization problem.

$$\begin{aligned} &\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x), \\ &f_i(x) \triangleq \frac{1}{m}\sum_{j\in\mathcal{D}_i} \log\left(1+\exp(-x^\top u_j v_j)\right) + \frac{\rho}{2}\left\|x\right\|^2, \end{aligned} \tag{5}$$

where $\mathcal{D}_i (i = 1, \ldots, n)$ denotes the local dataset for agent i, such that $m = |\mathcal{D}_i|$ for all i. ρ is set as 0.2.

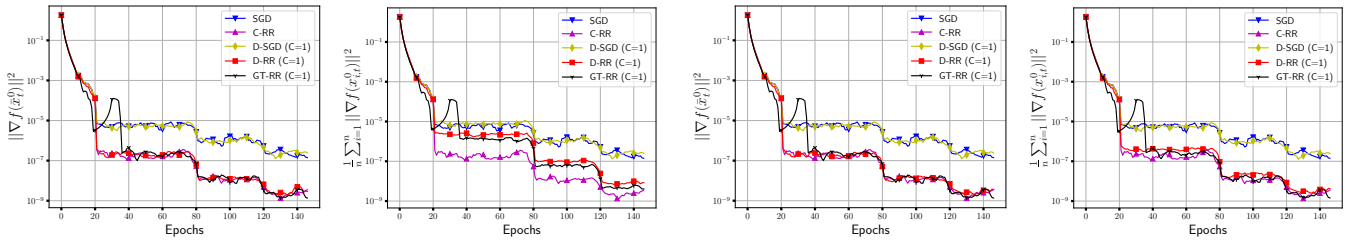
In Figure 1, we compare the performance of D-RR and GT-RR with that of D-SGD over a network of $n = 16$ agents connected in a ring, grid, and exponential graph. Centralized-RR (C-RR) and SGD serve as respective baselines. We plot the average iterate distance from the optimum $\frac{1}{n}\sum_{i=1}^n \left\|x_{i,t}^0 - x^*\right\|^2$. As discussed in Theorem 1 and Remark 1, the error floor for D-RR has network dependence. This is evident from the worse error floors of D-RR and GT-RR. However, if the graph is well-connected, as with exponential graph (Figure 1c), the difference is negligible.

We also explore the impact of changing the communication frequency. As communication becomes more infrequent (increasing value of C) the error floor worsens. However, for well-connected networks, D-RR can still achieve a significantly better error floor than D-SGD (with $C = 1$). For exponential graphs, we can choose C as large as 25 and still outperform D-SGD. This suggests that decentralized RR-based methods with multiple gradient steps between successive communications can be beneficial in communication-constrained settings and require further exploration. The special case with $C = m$ has been studied in [15].



(a) Ring graph (b) Grid graph (c) Exponential graph

Fig. 1: Comparison of decentralized random reshuffling based algorithms: D-RR and GT-RR with centralized-RR, D-SGD and SGD for solving the strongly-convex binary classification problem (5) on CIFAR10 dataset, over networks with $n = 16$ agents. The step-size is 0.001. C quantifies the frequency of communication relative to gradient computation. For $C \geq 1$, C consecutive stochastic gradient steps are followed by one round of communication with the neighboring agents. For $C < 1$, each stochastic gradient step is followed by $1/C$ rounds of communication.



(a) Ring graph (b) Exponential graph

Fig. 2: Comparison of decentralized random reshuffling based algorithms: D-RR and GT-RR with centralized-RR, D-SGD and SGD for solving the strongly-convex binary classification problem (5) on CIFAR10 dataset, over networks with $n = 16$ agents. C is fixed at 1. The step-size is sequentially set as $0.02, 0.004$ and 0.001.

At the other extreme, in some cases, communication might be cheaper than gradient computation. This case is represented by the $C < 1$ case, where each gradient computation is followed by $1/C$ rounds of communication. As expected, more communication results in improved consensus error, hence better convergence.

In the nonconvex case, we again solve a binary classification problem, but with a different regularizer, which leads to the following optimization problem.

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{6}$$

$$f_i(x) \triangleq \frac{1}{m}\sum_{j\in\mathcal{D}_i} \log\left(1+\exp(-x^\top u_j v_j)\right) + \frac{\eta}{2}\sum_{q=1}^{d}\frac{x_q^2}{1+x_q^2},$$

where x_q denotes the q-th element of $x \in \mathbb{R}^d$. We set $\eta = 0.2$.

In Figure 2, we fix the communication frequency to $C = 1$. We plot both the gradient norm at the average iterate $\left\|\nabla f(\bar{x}_t^0)\right\|^2$, as well as the average of gradient norms at individual iterates $\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f(x_{i,t}^0)\right\|^2$. For both ring and exponential graphs, the convergence of $\left\|\nabla f(\bar{\mathbf{x}}_t^0)\right\|^2$ seems independent of the network. This observation points to the need for an improved theoretical analysis, which removes the network dependence from the leading terms in Theorem 2. On the other hand, $\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f(\bar{\mathbf{x}}_{i,t}^0)\right\|^2$ has network dependence. GT-RR improves the error floor compared to D-RR but does not eliminate the network dependence completely.

4. CONCLUSION

We presented improved analyses of two existing random-reshuffling based decentralized algorithms, D-RR and GT-RR. We show that the convergence of the two algorithms improves with increasing network size. Experimental results, while corroborating our theory, also point out some gaps in the current theoretical understanding of the nonconvex case, which requires further investigation. Other pertinent directions for future work include proposing algorithms that achieve network-independent convergence and shorter transient times. Corresponding work on D-SGD like methods [7, 10] can possibly provide some insights in this direction.

5. REFERENCES

[1] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, vol. 31, no. 9, pp. 803–812, 1986.

[2] Dimitri Bertsekas, *Network optimization: continuous and discrete models*, vol. 8, Athena Scientific, 1998.

[3] Angelia Nedic and Asuman Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[4] Yurii Nesterov, *Lectures on convex optimization*, vol. 137, Springer, 2018.

[5] 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," *Advances in neural information processing systems*, vol. 30, 2017.

[6] Ran Xin, Usman A Khan, and Soummya Kar, "An improved convergence analysis for decentralized online stochastic non-convex optimization," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1842–1858, 2021.

[7] Shi Pu, Alex Olshevsky, and Ioannis Ch Paschalidis, "Asymptotic network independence in distributed stochastic optimization for machine learning: Examining distributed and centralized stochastic gradient descent," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 114–122, 2020.

[8] Shi Pu and Angelia Nedić, "Distributed stochastic gradient tracking methods," *Mathematical Programming*, vol. 187, pp. 409–457, 2021.

[9] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu, "D^2: Decentralized training over decentralized data," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4848–4856.

[10] Sulaiman A Alghunaim and Kun Yuan, "A unified and refined convergence analysis for non-convex decentralized learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3264–3279, 2022.

[11] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," *arXiv preprint arXiv:1109.5647*, 2011.

[12] M Gurbuzbalaban, Asu Ozdaglar, and Pablo A Parrilo, "Convergence rate of incremental gradient and incremental newton methods," *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 2542–2565, 2019.

[13] Lam M Nguyen, Quoc Tran-Dinh, Dzung T Phan, Phuong Ha Nguyen, and Marten Van Dijk, "A unified convergence analysis for shuffling-type gradient methods," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 9397–9440, 2021.

[14] Kun Yuan, Bicheng Ying, Jiageng Liu, and Ali H Sayed, "Variance-reduced stochastic learning by networked agents under random reshuffling," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 351–366, 2018.

[15] Xia Jiang, Xianlin Zeng, Jian Sun, Jie Chen, and Lihua Xie, "Distributed stochastic proximal algorithm with random reshuffling for nonsmooth finite-sum optimization," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] Kun Huang, Xiao Li, Andre Milzarek, Shi Pu, and Junwen Qiu, "Distributed random reshuffling over networks," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1143–1158, 2023.

[17] Kun Huang, Linli Zhou, and Shi Pu, "Distributed random reshuffling methods with improved convergence," *arXiv preprint arXiv:2306.12037*, 2023.

[18] Konstantin Mishchenko, Ahmed Khaled, and Peter Richtárik, "Random reshuffling: Simple analysis with vast improvements," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17309–17320, 2020.

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