2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Achieving Linear Speedup with Network-Independent Learning Rates in Decentralized Stochastic Optimization

Hao Yuan, Sulaiman A. Alghunaim, and Kun Yuan

Abstract— **Decentralized stochastic optimization has become a crucial tool for addressing large-scale machine learning and control problems. In decentralized algorithms, all computing nodes are connected through a network topology, and each node communicates only with its direct neighbors. Decentralized algorithms can significantly reduce communication overhead by eliminating the need for global communication. However, existing research on the linear speedup analysis of decentralized stochastic algorithms is limited to the condition of network-dependent learning rates, which rarely holds in practice since the network connectivity is typically unknown to each node. As a result, it remains an open question whether a linear speedup bound can be achieved using network-independent learning rates. This paper provides an affirmative answer. By utilizing a new analysis framework, we prove that D-SGD and Exact-Diffusion, two representative decentralized stochastic algorithms, can achieve linear speedup with network-independent learning rates. Simulations are provided to validate our theories.**

I. INTRODUCTION

In decentralized stochastic optimization, a network of n nodes collaborates to solve the following problem:

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

The local cost function $f_i:\mathbb{R}^d\to\mathbb{R}$ is kept privately by node i and ξ_i is a random variable that represents the local data following distribution $\mathcal{D}_i$. Each node i can evaluate stochastic gradient $\nabla F(x_i;\xi_i)$ locally, and must communicate in order to access information from other nodes. Decentralized algorithms operate on distributed data and communicate solely with direct neighbors, effectively eliminating the need for data sharing and centralized synchronization. As a result, they have been widely applied across various domains, including deep training with massive GPUs [1]–[5] and edge computing with extensive embedded devices [6]–[8].

Various effective algorithms have been proposed to solve problem (1), including D-SGD (also referred to as Diffusion) [1], [2], [9]–[12], explicit bias-correction methods such as EXTRA [13], Exact-Diffusion/NIDS [14]–[17], and Gradient-Tracking [18]–[23]. All of these algorithms have been proven to achieve linear speedup asymptotically. Linear speedup is a key feature of decentralized algorithms, wherein the convergence accuracy improves linearly with the number of nodes. For instance, when dealing with a strongly-convex problem, D-SGD can accomplish an accuracy $O(\sigma^2/n)$ with a constant learning rate where σ^2 is the magnitude of gradient noise, indicating that a more accurate solution can be attained by increasing the number of nodes (*e.g.*, machines) used.

The popularity of decentralized optimization is primarily due to its linear speedup property. However, recent research has shown that this property relies on the condition that it uses network-dependent learning rates [2], [12], [17], [20], [22]–[26] . These learning rates are strongly correlated with the connectivity of the network topology, particularly the second-largest eigenvalue of the mixing matrix (refer to Sec. II-B), which is unknown to each node. Although several methods [27], [28] have been proposed to estimate network connectivity, implementing them may result in additional communication overhead and ultimately reduce the efficiency of decentralized optimization.

Recent research has focused on establishing the convergence of decentralized optimization algorithms when learning rates are independent of network connectivity. For example, Exact-Diffusion/NIDS [15], [29] and the decentralized inexact proximal gradient method [30] have been demonstrated to converge with network-independent learning rates. However, these works only concentrate on deterministic optimization problems. To the best of our knowledge, there are currently no existing results that can achieve the linear speedup property with network-independent learning rates in decentralized stochastic optimization.

This paper presents a novel approach to achieving linear speedup in decentralized stochastic optimization using network-independent learning rates. Our contribution comprises three key elements. First, we demonstrate that simply adapting the analysis framework used in deterministic Exact-Diffusion/NIDS [29] to stochastic scenarios is inadequate for ensuring linear speedup, despite being capable of guaranteeing convergence. Second, we propose a new analysis that enables Exact-Diffusion/NIDS to achieve linear speedup with network-independent learning rates. Third, we apply the same analysis framework to show that D-SGD can achieve similar results. Our main results are listed in Table I.

Notations. We let $x_i\in\mathbb{R}^d$ be the estimate of $x\in\mathbb{R}^d$ at node i and introduce the augmented network quantities:

$$\mathbf{x}\triangleq \mathrm{col}\{x_1,\ldots,x_n\}\in\mathbb{R}^{nd},$$

Hao Yuan and Kun Yuan are with Center for Machine Learning Research, Peking University, Beijing 100871, P. R. China. Emails: {pkuyuanhao, kunyuan}@pku.edu.cn. Kun Yuan is also with AI for Science Institute, Beijing, P. R. China, and National Engineering Labratory for Big Data Analytics and Applications, Beijing, P. R. China.

Sulaiman A. Alghunaim is with Dept. Electrical Engr., Kuwait University, Safat 13060, Kuwait. sulaiman.alghunaim@ku.edu.kw

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

TABLE I: Comparisons of learning rates and the convergent accuracy between various decentralized algorithms for strongly-convex settings. We list the upper bound on the learning rate α in the "Learning rate" column. The quantity $\lambda \in (0,1)$ denotes the mixing rate of the network, see Eq. (4).

METHODS	LEARNING RATE	CONVERGENT ACC.
D-SGD [12]	$O(\frac{1-\lambda}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$
ED/NIDS [15]	$O(\frac{1}{L})$	N.A.
ED/NIDS [17], [24]	$O(\frac{1-\lambda}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$
GT [20], [31]	$O(\frac{(1-\lambda)^2}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$
GT [22], [23], [26]	$O(\frac{1-\lambda}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$
D-SGD (Thm.3)	$O(\frac{1}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$
ED/NIDS (Thm.2)	$O(\frac{1}{L})$	$O(\frac{\alpha\sigma^2}{n}) + O(\alpha^2)$

$$\mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^{n} f_i(x_i),$$
$$\nabla \mathbf{f}(\mathbf{x}) \triangleq \mathrm{col}\{\nabla f_1(x_1), \dots, \nabla f_n(x_n)\} \in \mathbb{R}^{nd},$$
$$\nabla \mathbf{F}(\mathbf{x};\xi) \triangleq \mathrm{col}\{\nabla F_1(x_1;\xi_1), \dots, \nabla F_n(x_n;\xi_n)\} \in \mathbb{R}^{nd}.$$

We also define the combination (mixing) matrices as

$$W = [w_{ij}] \in \mathbb{R}^{n\times n} \text{ and } \mathbf{W} \triangleq W \otimes I_d \in \mathbb{R}^{nd\times nd},$$

where w_{ij} represents the weight assigned to scale information transmitted from node j to node i. If nodes i and j are not directly connected, then $w_{ij} = w_{ji} = 0$. The symbol $\otimes$ denotes the Kronecker product. Additionally, given a positive definite matrix $A \in \mathbb{R}^{n\times n}$ and its eigen-decomposition $A = U\Lambda U^T$, where Λ is a positive diagonal matrix, we define $A^{1/2} \triangleq U\Lambda^{1/2}U^T$. We let $\mathbf{1} \in \mathbb{R}^n$ denote the vector with each element being 1. Moreover, we let $\|\cdot\|$ denote $\|\cdot\|_2$ throughout the paper.

II. DECENTRALIZED STOCHASTIC METHODS

This section describes the two studied decentralized algorithms, namely D-SGD and Exact-Diffusion, and introduces the assumptions required for their convergence analysis.

A. Decentralized stochastic algorithms

D-SGD. D-SGD was first developed in [9], [10] for control and adaptive signal processing, and then studied for machine learning applications [1], [2], [12]. Let $\mathbf{x}^0$ take any arbitrary value, D-SGD will iterate as follows:

$$\mathbf{x}^{k+1} = \mathbf{W}\Big(\mathbf{x}^k - \alpha\nabla\mathbf{F}(\mathbf{x}^k;\xi^k)\Big), \quad k = 0,1,2,\dots \tag{2}$$

where α is a constant learning rate. Note that the above algorithm is also known as (adapt-then-combine) Diffusion [10], [27], [32], [33] in the signal processing community.

Exact-Diffusion. Exact-Diffusion was first developed in [14], [15] for decentralized deterministic optimization and then studied under stochastic scenario [16], [24]. It aims to remove the intrinsic bias in D-SGD caused by data heterogeneity. Let $\mathbf{x}^0$ take any arbitrary value and $\boldsymbol{\psi}^0 = \mathbf{x}^0$, Exact-Diffusion will iterate for $k \ge 0$ as follows:

$$\boldsymbol{\psi}^{k+1} = \mathbf{x}^k - \alpha\nabla\mathbf{F}(\mathbf{x}^k;\xi^k), \tag{3a}$$
$$\boldsymbol{\phi}^{k+1} = \boldsymbol{\psi}^{k+1} + \mathbf{x}^k - \boldsymbol{\psi}^k, \tag{3b}$$
$$\mathbf{x}^{k+1} = \mathbf{W}\boldsymbol{\phi}^{k+1}. \tag{3c}$$

Exact-Diffusion is also known as NIDS in [15] or D^2 in [16]. We will refer to it as Exact-Diffusion throughout the paper.

B. Assumptions

We make the following standard assumptions to establish our convergence results for D-SGD and Exact-Diffusion.

Assumption 1 (COMBINATION MATRIX)**.** *The weight matrix W is assumed to be primitive, positive semidefinite, and doubly stochastic.* ■

Under Assumption 1, the weight matrix W has a single eigenvalue at one, denoted by $\lambda_1 = 1$. All other eigenvalues, denoted by $\{\lambda_i\}_{i=2}^n$, are strictly less than one in magnitude [33]. The mixing rate of the network is defined as

$$\lambda \triangleq \left\|W - \tfrac{1}{n}\mathbf{1}\mathbf{1}^T\right\| = \max_{i\in\{2,\dots,n\}} |\lambda_i| < 1. \tag{4}$$

The mixing rate λ reflects the network connectivity. The scenario $\lambda \to 0$ implies a densely-connected topology (e.g., for fully connected topology, we can choose $W = \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and hence $\lambda = 0$). In contrast, the scenario $\lambda \to 1$ implies a sparsely-connected topology.

Assumption 2 (COST FUNCTION)**.** *Each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly-convex for some $L \ge \mu > 0$. This implies that the aggregate function $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is also L-smooth and μ-strongly convex.* ■

Assumption 3 (GRADIENT NOISE)**.** *For all nodes indices $i = 1,\dots,n$ and iterations $k = 0,1,\dots$, we assume that*

$$\mathbb{E}\left[\nabla F_i(x_i^k;\xi_i^k) - \nabla f_i(x_i^k) \mid \mathcal{F}^k\right] = 0, \tag{5a}$$
$$\mathbb{E}\left[\|\nabla F_i(x_i^k;\xi_i^k) - \nabla f_i(x_i^k)\|^2 \mid \mathcal{F}^k\right] \le \sigma^2, \tag{5b}$$

for some $\sigma^2 \ge 0$ where $\mathcal{F}^k \triangleq \{\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^k\}$ is the filtration generated by Algorithm 2 or (3). We also assume that conditioned on $\mathcal{F}^k$, the random data $\{\xi_i^t\}$ are independent of each other for all $\{i\}_{i=1}^n$ and $\{t\}_{t\le k}$. ■

III. STABILITY ANALYSIS OF EXACT-DIFFUSION

Since the analysis of Exact-Diffusion is more challenging than D-SGD, we will provide a detailed exploration of its stability and linear speedup property. This analysis technique can be easily adapted to D-SGD, see Sec. V.

A. Primal-dual update of Exact-Diffusion

To facilitate the analysis of Exact-Diffusion, we rewrite recursion (3) into the following primal-dual form [29]:

$$\mathbf{v}^{k+1} = \mathbf{x}^k - \alpha\nabla\mathbf{F}(\mathbf{x}^k;\xi^k) - \mathbf{B}\mathbf{y}^k, \tag{6a}$$
$$\mathbf{x}^{k+1} = \mathbf{W}\mathbf{v}^{k+1}, \tag{6b}$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{B}\mathbf{v}^{k+1}, \tag{6c}$$

where $\mathbf{y}^k \in \mathbb{R}^{nd}$ is the dual variable with $\mathbf{y}^0 = 0$ and $\mathbf{B} \triangleq (\mathbf{I} - \mathbf{W})^{1/2} \in \mathbb{R}^{nd\times nd}$. Under our assumptions, there exists a primal dual pair $(\mathbf{x}^\star, \mathbf{y}^\star)$ satisfying [23]:

$$\mathbf{0} = \alpha\nabla\mathbf{f}(\mathbf{x}^\star) + \mathbf{B}\mathbf{y}^\star, \tag{7a}$$

$$\mathbf{0} = \mathbf{B}\mathbf{x}^\star, \tag{7b}$$

and we have that $\mathbf{x}^\star = \mathbf{1} \otimes x^\star$ and $x^\star$ is the optimal solution to (1). We refer to (7) as the optimality condition. Using (6) and (7) and letting $\tilde{\mathbf{v}}^k \triangleq \mathbf{v}^k - \mathbf{x}^\star$, $\tilde{\mathbf{x}}^k \triangleq \mathbf{x}^k - \mathbf{x}^\star$, and $\tilde{\mathbf{y}}^k \triangleq \mathbf{y}^k - \mathbf{y}^\star$, we get the following error recursion

$$\tilde{\mathbf{v}}^{k+1} = \tilde{\mathbf{x}}^k - \alpha\left(\nabla \mathbf{f}(\mathbf{x}^k) - \nabla \mathbf{f}(\mathbf{x}^\star) + \mathbf{s}^k\right) - \mathbf{B}\tilde{\mathbf{y}}^k, \tag{8a}$$

$$\tilde{\mathbf{x}}^{k+1} = \mathbf{W}\tilde{\mathbf{v}}^{k+1}, \tag{8b}$$

$$\tilde{\mathbf{y}}^{k+1} = \tilde{\mathbf{y}}^k + \mathbf{B}\tilde{\mathbf{v}}^{k+1}, \tag{8c}$$

where $\mathbf{s}^k \triangleq \nabla \mathbf{F}(\mathbf{x}^k;\xi^k) - \nabla \mathbf{f}(\mathbf{x}^k) \in \mathbb{R}^{nd}$.

B. Stability analysis

Given the error recursion (8), we are ready to derive the range of learning rates that enable Exact-Diffusion to converge. Adapting the analysis in [29] to recursion (8), we obtain the following result.

Theorem 1 (A CONVERGENCE RESULT). *Under Assumptions 1–3, if $\alpha \le \frac{1}{L}$, it then holds that*

$$\begin{aligned}&\mathbb{E}\,\|\tilde{\mathbf{x}}^{k+1}\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^{k+1}\|^2 \\ \le\,& \rho\left(\mathbb{E}\,\|\tilde{\mathbf{x}}^k\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^k\|^2\right) + n\alpha^2\sigma^2,\end{aligned} \tag{9}$$

where $\rho \triangleq \max\{(1-\mu\alpha)^2, \lambda\} \in (0,1)$.

Proof: See Appendix I. ■

Remark 1. *Theorem 1 shows that Exact-Diffusion converges for any network-independent learning rate $\alpha \le \frac{1}{L}$. Using $\|\tilde{\mathbf{x}}^k\|^2 = n\|\bar{x}^k - x^\star\|^2 + \sum_{i=1}^n \|x_i^k - \bar{x}^k\|^2$ where $\bar{x}^k = \frac{1}{n}\sum_{i=1}^n x_i^k$ and iterating* (9)*, it holds that*

$$\mathbb{E}\,\|\bar{x}^k - x^\star\|^2 \le C\rho^k + \tfrac{\alpha^2\sigma^2}{1-\rho} \tag{10}$$

where $C = \left(\mathbb{E}\,\|\tilde{\mathbf{x}}^0\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^0\|^2\right)/n$. From (10)*, we observe that the linear speedup term $O(\alpha\sigma^2/n)$ cannot be attained by letting $\alpha \le \frac{1}{L}$. In other words, a direct extension of the analysis in [29] to stochastic scenario cannot ensure linear speedup, despite being able to guarantee convergence.* ■

IV. LINEAR SPEEDUP ANALYSIS OF EXACT-DIFFUSION

A. Fundamental transformation

To obtain linear speedup in Exact-Diffusion, we transform (8) into an equivalent recursion. This transformation is fundamental for establishing the linear speedup convergence of Exact-Diffusion with network-independent learning rates. We first introduce several notations:

$$\bar{x}^k \triangleq \frac{1}{n}(\mathbf{1}_n^T \otimes I_d)\mathbf{x}^k = \frac{1}{n}\sum_{i=1}^n x_i^k, \tag{11a}$$

$$\bar{e}_x^k \triangleq \frac{1}{n}(\mathbf{1}_n^T \otimes I_d)\tilde{\mathbf{x}}^k = \bar{x}^k - x^\star, \tag{11b}$$

$$\bar{s}^k \triangleq \frac{1}{n}(\mathbf{1}_n^T \otimes I_d)\mathbf{s}^k = \frac{1}{n}\sum_{i=1}^n s_i^k, \tag{11c}$$

$$\overline{\nabla f}(\mathbf{x}^k) \triangleq \frac{1}{n}(\mathbf{1}_n^T \otimes I_d)\nabla \mathbf{f}(\mathbf{x}^k) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^k). \tag{11d}$$

Moreover, we introduce the diagonal matrix

$$\mathbf{\Lambda} \triangleq \begin{bmatrix} \lambda_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \otimes I_d \in \mathbb{R}^{(n-1)d \times (n-1)d} \tag{12}$$

and $\mathbf{\Lambda}_b \triangleq (\mathbf{I} - \mathbf{\Lambda})^{1/2}$. It can be verified that matrices $\mathbf{W}$ and $\mathbf{B}$ have the following eigen-decomposition

$$\begin{aligned}\mathbf{W} &= \mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1} \\ &= \underbrace{\begin{bmatrix}\mathbf{1}\otimes I_d & \hat{\mathbf{U}}\end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} I_d & 0 \\ 0 & \mathbf{\Lambda}\end{bmatrix}}_{\mathbf{\Sigma}} \underbrace{\begin{bmatrix}\frac{1}{n}\mathbf{1}^T \otimes I_d \\ \hat{\mathbf{U}}^T\end{bmatrix}}_{\mathbf{U}^{-1}},\end{aligned} \tag{13a}$$

$$\begin{aligned}\mathbf{B} &= \mathbf{U}\mathbf{\Sigma}_b\mathbf{U}^{-1} \\ &= \begin{bmatrix}\mathbf{1}\otimes I_d & \hat{\mathbf{U}}\end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{\Lambda}_b\end{bmatrix} \begin{bmatrix}\frac{1}{n}\mathbf{1}^T \otimes I_d \\ \hat{\mathbf{U}}^T\end{bmatrix},\end{aligned} \tag{13b}$$

where matrix $\hat{\mathbf{U}} \in \mathbb{R}^{nd \times (n-1)d}$ satisfies the following properties:

$$\hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{I},\ (\mathbf{1}^T \otimes I_d)\hat{\mathbf{U}} = 0, \hat{\mathbf{U}}\hat{\mathbf{U}}^T = \mathbf{I} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^T \otimes I_d. \tag{14}$$

With the above notation, we can transform Exact-Diffusion (8) into an equivalent but fundamental recursion.

Lemma 1 (TRANSFORMED ERROR RECURSION). *Under Assumption 1, there exists a matrix $\hat{\mathbf{V}}^{-1} \in \mathbb{C}^{2(n-1)d \times 2(n-1)d}$ and a block diagonal matrix $\mathbf{\Gamma} \in \mathbb{C}^{2(n-1)d \times 2(n-1)d}$ so that*

$$\bar{e}_x^{k+1} = \bar{e}_x^k - \alpha\overline{\nabla f}(\mathbf{x}^k) - \alpha\bar{s}^k, \tag{15a}$$

$$\hat{\mathbf{x}}^{k+1} = \mathbf{\Gamma}\hat{\mathbf{x}}^k - \alpha\hat{\mathbf{V}}^{-1}\begin{bmatrix}\mathbf{\Lambda}\hat{\mathbf{U}}^T\big(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star) + \mathbf{s}^k\big) \\ \mathbf{\Lambda}\mathbf{\Lambda}_b\hat{\mathbf{U}}^T\big(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star) + \mathbf{s}^k\big)\end{bmatrix}, \tag{15b}$$

where

$$\hat{\mathbf{x}}^k \triangleq \hat{\mathbf{V}}^{-1}\begin{bmatrix}\hat{\mathbf{U}}^T\tilde{\mathbf{x}}^k \\ \mathbf{\Lambda}\hat{\mathbf{U}}^T\tilde{\mathbf{y}}^k\end{bmatrix}. \tag{16}$$

Moreover, we have $\|\hat{\mathbf{V}}\|^2 = 2$, $\|\hat{\mathbf{V}}^{-1}\|^2 \le \frac{1}{2\underline{\lambda}}$, $\|\mathbf{\Gamma}\| = \sqrt{\lambda}$, $\|\mathbf{\Lambda}\| = \lambda$, where $\lambda = \max_{i\in\{2,\ldots,n\}}\lambda_i$ and $\underline{\lambda}$ is the minimum non-zero eigenvalue of W.

The proof is omitted due to space constraints.

Before deriving our final result about linear speedup of Exact-Diffusion, we first present two lemmas.

Lemma 2 (COUPLED ERROR INEQUALITY). *Under Assumptions 1–3, if $\alpha \le \frac{1}{4L}$ and we start from consensual initialization $\mathbf{x}^0 = \mathbf{1} \otimes x^0$, then we have*

$$\mathbb{E}\,\|\bar{e}_x^{k+1}\|^2 \le (1-\mu\alpha)\,\mathbb{E}\|\bar{e}_x^k\|^2 + \tfrac{3L\alpha}{n}\,\mathbb{E}\,\|\hat{\mathbf{x}}^k\|^2 + \tfrac{\alpha^2\sigma^2}{n}, \tag{17a}$$

$$\mathbb{E}\,\|\hat{\mathbf{x}}^{k+1}\|^2 \le \sqrt{\lambda}\,\mathbb{E}\,\|\hat{\mathbf{x}}^k\|^2 + \tfrac{\alpha^2\lambda^2 n\sigma^2}{\underline{\lambda}} + \tfrac{\alpha^2\lambda^2L^2}{\underline{\lambda}(1-\sqrt{\lambda})}\,\mathbb{E}\,\|\tilde{\mathbf{x}}^k\|^2. \tag{17b}$$

The proof is omitted due to space constraints.

Lemma 3 (UPPER BOUND OF $\|\hat{\mathbf{x}}^0\|^2$). *If $\mathbf{x}^0$ and $\mathbf{y}^0$ are initialized as $\mathbf{x}^0 = \mathbf{1} \otimes x^0$ and $\mathbf{y}^0 = 0$, then we have*

$$\|\hat{\mathbf{x}}^0\|^2 \le \frac{\alpha^2}{2\underline{\lambda}(1-\lambda)}\|\nabla\mathbf{f}(\mathbf{x}^\star)\|^2. \tag{18}$$

The proof is omitted due to space constraints.

We now show that Exact-Diffusion can achieve linear speedup with network-independent learning rates.

Theorem 2 (EXACT-DIFFUSION CONVERGENCE). *Under Assumptions 1–3, if $\alpha \le \frac{1}{4L}$ and $\mathbf{x}^0 = \mathbf{1} \otimes x^0$, it holds that*

$$\mathbb{E}\,\|\bar{e}_x^k\|^2 \le (1-\mu\alpha)^k\,\mathbb{E}\,\|\bar{e}_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu} + O(\alpha^2). \tag{19}$$

Proof: From Theorem 1, we obtain

$$\begin{aligned}\mathbb{E}\,\|\tilde{\mathbf{x}}^k\|^2 \le& \mathbb{E}\,\|\tilde{\mathbf{x}}^k\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^k\|^2\\ \le&\rho\big(\,\mathbb{E}\,\|\tilde{\mathbf{x}}^{k-1}\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^{k-1}\|^2\big) + n\alpha^2\sigma^2\\ \le&\rho^k\big(\|\tilde{\mathbf{x}}^0\|^2 + \|\tilde{\mathbf{y}}^0\|^2\big) + \tfrac{n\alpha^2\sigma^2}{1-\rho}.\end{aligned}$$

Using Lemma 2 and substituting the above inequality into (17b), we get

$$\begin{aligned}\mathbb{E}\,\|\hat{\mathbf{x}}^{k+1}\|^2 \le& \sqrt{\lambda}\,\mathbb{E}\,\|\hat{\mathbf{x}}^k\|^2 + \tfrac{\alpha^2\lambda^2 n\sigma^2}{\underline{\lambda}} + \tfrac{\alpha^4 n\sigma^2 L^2\lambda^2}{\underline{\lambda}(1-\sqrt{\lambda})(1-\rho)}\\ &+ \tfrac{\alpha^2 L^2\lambda^2}{\underline{\lambda}(1-\sqrt{\lambda})}\rho^k\big(\|\tilde{\mathbf{x}}^0\|^2 + \|\tilde{\mathbf{y}}^0\|^2\big).\end{aligned} \tag{20}$$

Let

$$D \triangleq \tfrac{\alpha^2 L^2\lambda^2}{\underline{\lambda}(1-\sqrt{\lambda})}\big(\|\tilde{\mathbf{x}}^0\|^2 + \|\tilde{\mathbf{y}}^0\|^2\big), \tag{21}$$

$$F \triangleq \tfrac{\alpha^2\lambda^2 n\sigma^2}{\underline{\lambda}} + \tfrac{\alpha^4 n\sigma^2 L^2\lambda^2}{\underline{\lambda}(1-\sqrt{\lambda})(1-\rho)}. \tag{22}$$

By iterating (20), we get

$$\begin{aligned}\mathbb{E}\,\|\hat{\mathbf{x}}^k\|^2 \le&\sqrt{\lambda}\,\mathbb{E}\,\|\hat{\mathbf{x}}^{k-1}\|^2 + \rho^{k-1}D + F\\ \overset{(a)}{\le}&(\sqrt{\lambda})^k\|\hat{\mathbf{x}}^0\|^2 + \tfrac{1}{1-\sqrt{\lambda}}(D+F)\\ \le&\|\hat{\mathbf{x}}^0\|^2 + \tfrac{1}{1-\sqrt{\lambda}}(D+F)\\ \overset{(b)}{\le}&\tfrac{\alpha^2}{2\underline{\lambda}(1-\lambda)}\|\nabla\mathbf{f}(\mathbf{x}^\star)\|^2 + \tfrac{1}{1-\sqrt{\lambda}}(D+F),\end{aligned} \tag{23}$$

where inequality (a) holds due to $\rho < 1$, and inequality (b) holds due to Lemma 3. Using Lemma 2 and substituting (23) into (17a), we have

$$\begin{aligned}\mathbb{E}\,\|\bar{e}_x^k\|^2 \le& (1-\mu\alpha)\,\mathbb{E}\,\|\bar{e}_x^{k-1}\|^2 + \tfrac{3L\alpha}{n}\,\mathbb{E}\,\|\hat{\mathbf{x}}^{k-1}\|^2 + \tfrac{\alpha^2\sigma^2}{n}\\ \le& (1-\mu\alpha)\,\mathbb{E}\,\|\bar{e}_x^{k-1}\|^2 + \tfrac{\alpha^2\sigma^2}{n}\\ &+ \tfrac{3L\alpha}{n}\big(\tfrac{\alpha^2}{2\underline{\lambda}(1-\lambda)}\|\nabla\mathbf{f}(\mathbf{x}^\star)\|^2 + \tfrac{1}{1-\sqrt{\lambda}}(D+F)\big)\\ \le& (1-\mu\alpha)^{k-1}\|e_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu}\\ &+\tfrac{3L}{n\mu}\big(\tfrac{\alpha^2}{2\underline{\lambda}(1-\lambda)}\|\nabla\mathbf{f}(\mathbf{x}^\star)\|^2 + \tfrac{1}{1-\sqrt{\lambda}}(D+F)\big).\end{aligned} \tag{24}$$

Substituting the definitions of D and F into the above inequality, we get

$$\begin{aligned}\mathbb{E}\,\|\bar{e}_x^k\|^2 \le& (1-\mu\alpha)^k\|\bar{e}_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu} + \tfrac{3\alpha^2 L}{2n\mu\underline{\lambda}(1-\lambda)}\|\nabla\mathbf{f}(\mathbf{x}^*)\|^2\\ &+ \tfrac{3\alpha^2L^3\lambda^2}{n\mu\underline{\lambda}(1-\sqrt{\lambda})^2}(\|\tilde{\mathbf{x}}^0\|^2 + \|\tilde{\mathbf{y}}^0\|^2)\\ &+ \tfrac{3\alpha^2 L\lambda^2\sigma^2}{\mu\underline{\lambda}(1-\sqrt{\lambda})} + \tfrac{3\alpha^4L^3\lambda^2\sigma^2}{\mu\underline{\lambda}(1-\sqrt{\lambda})^2(1-\rho)}\\ =& (1-\mu\alpha)^k\|\bar{e}_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu} + O(\alpha^2),\end{aligned} \tag{25}$$

which is the result listed in (19). ■

Remark 2 (LINEAR SPEEDUP UNDER NETWORK-INDEPENDENT LEARNING RATE). *It is established in Theorem 2 that the linear speedup term $O(\alpha\sigma^2/n)$ can be achieved when $\alpha \le \frac{1}{4L}$, a condition that is network independent. When the learning rate α is sufficiently small, the term $O(\alpha\sigma^2/n)$ dominates* (19)*, which improves linearly with the number of nodes n.* ■

Remark 3 (TIGHTER UPPER BOUND). *While the analysis in Theorem 2 establishes the linear speedup property of Exact-Diffusion, it is not sharp due to using loose bounds to simplify the derivations. For example, the bound* (23) *does not imply convergence to zero error for constant learning rate and deterministic scenario in which $\sigma^2 = 0$, which contradicts the results in [14], [15], [29]. With a refined (but lengthy) analysis, we can establish the following tighter bound for $\alpha \le \frac{1}{4L}$:*

$$\begin{aligned}\mathbb{E}\,\|\bar{e}_x^k\|^2 \le& (1-\mu\alpha)^k\|\bar{e}_x^0\|^2\\ &+ \rho_0^k\Big(\|\tilde{\mathbf{x}}^0\|^2 + \|\tilde{\mathbf{y}}^0\|^2 + \|\nabla\mathbf{f}(\mathbf{x}^\star)\|^2\Big)O(\alpha^3)\\ &+ \tfrac{\alpha\sigma^2}{n\mu} + \tfrac{3\alpha^2L\lambda^2\sigma^2}{\mu\underline{\lambda}(1-\sqrt{\lambda})}\big(\tfrac{\alpha^2L^2}{(1-\sqrt{\lambda})(1-\rho)} + 1\big).\end{aligned} \tag{26}$$

where $\rho_0 = \max\{\sqrt{\lambda}, 1-\mu\alpha\}$, and $\rho = \max\{(1-\mu\alpha)^2, \lambda\} < 1$. With the above bound, when $\sigma^2 = 0$, Exact-Diffusion will converge to 0 exponentially fast as $k \to \infty$. This is consistent with [14], [15], [29]. We omit the analysis details due to space limits. ■

V. LINEAR SPEEDUP ANALYSIS OF D-SGD

The convergence framework for Exact-Diffusion discussed in Sec. IV can also be adapted to D-SGD to show the linear speedup with network-independent learning rates. Before presenting the result, we let $\mathbf{x}^\infty \in \mathbb{R}^{nd}$ denote the fixed point of the deterministic D-SGD algorithm:

$$\mathbf{x}^\infty = \mathbf{W}[\mathbf{x}^\infty - \alpha\nabla\mathbf{f}(\mathbf{x}^\infty)]. \tag{27}$$

Theorem 3 (D-SGD CONVERGENCE). *Under Assumptions 1–3, if $\alpha \le \frac{1}{4L}$ and $\mathbf{x}^0 = \mathbf{1} \otimes x^0$, then it holds that*

$$\mathbb{E}\,\|\bar{e}_x^k\|^2 \le (1-\mu\alpha)^k\|\bar{e}_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu} + O(\alpha^2). \tag{28}$$

Proof: The proof of Theorem 3 follows similar, but simpler, than the proof of Theorem 2. We omit it due to space limits. ■

Remark 4 (LINEAR SPEEDUP UNDER NETWORK-INDEPENDENT LEARNING RATE). *It is established in Theorem 3 that the linear speedup term $O(\alpha\sigma^2/n)$ can be achieved when $\alpha \le \frac{1}{4L}$, which is network independent.* ■

Remark 5 (TIGHTER UPPER BOUND). *Similar to Remark 3, we can also establish a much tighter bound for D-SGD. Under Assumptions 1–3, if $\alpha \le \frac{1}{4L}$, then it holds that*

$$\begin{aligned}\mathbb{E}\,\|\bar{e}_x^k\|^2 \le& (1-\mu\alpha)^k\,\mathbb{E}\,\|\bar{e}_x^0\|^2 + \tfrac{\alpha\sigma^2}{n\mu} + \tfrac{3\alpha^2L\lambda^2}{n\mu(1-\lambda)^2}\|\nabla\mathbf{f}(\mathbf{x}^\infty)\|^2\\ &+ \tfrac{3\alpha^2L\sigma^2}{2\mu(1-\lambda)}\big(\tfrac{2\alpha\lambda^2L^2}{\mu(1-\lambda)(2-\mu\alpha)} + 1\big) + \rho_1^k\|\mathbf{x}^0 - \mathbf{x}^\infty\|^2 O(\alpha^3),\end{aligned}$$

where $\rho_1 = \max\{1-\mu\alpha, \lambda\} < 1$. Comparing with Exact-Diffusion (26)*, D-SGD suffers from an additional intrinsic bias $O(\frac{\alpha^2}{(1-\lambda)^2})$ when using a constant learning rate even*

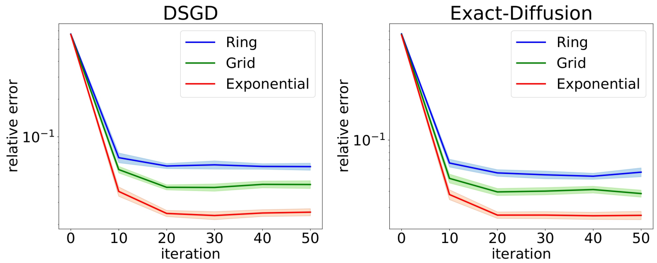

Fig. 1: Performances of D-SGD and Exact-Diffusion over ring, 2D-Grid and exponential graphs with 160 nodes. The spectral gap $1-\lambda$ is 0.00025, 0.00403 and 0.11111 for ring, 2D-Grid and exponential graphs respectively.

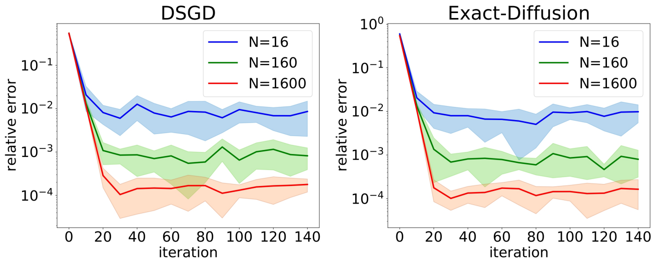

Fig. 2: Performances of D-SGD and Exact-Diffusion over ring graphs with 16, 160 and 1600 nodes.

when $\sigma^2 = 0$. This result is expected and consistent with previous works, see e.g., [24]. We omit the analysis of the above inequality due to space limits. ■

VI. Numerical Experiments

In this section, we present numerical simulations to validate our findings. We consider an ℓ_2-regularized logistic regression problem, where the objective function for each node i is given by $f_i(x) = \mathbb{E}[\ln(1+\exp(-y_i h_i^T x))] + \rho\|x\|^2$. Here, (h_i, y_i) represents the training dataset stored in node i, where $h_i \in \mathbb{R}^d$ denotes the feature vector and $y_i \in \{-1, +1\}$ denotes the label. To begin, we generate a local solution $x_i^\star$, as $x_i^\star = (x^\star + v_i)/\|x^\star\|$, where $v_i \sim \mathcal{N}(0, \sigma_v^2 I_d)$. Then, using $x_i^\star$, we generate local data with different distributions. To accomplish this. we generate each feature vector h_i at node i as $h_i \sim \mathcal{N}(0, I_d)$, and produce the corresponding label y_i as follows: create a random variable $z_i \sim \mathcal{U}(0,1)$, and set $y_i = 1$ if $z_i \le 1 + \exp(-y_i h_i^T x)$, and $y_i = -1$ otherwise. In this setup, the solution $x_i^\star$ controls the distribution of the labels, while σ_v^2 governs data heterogeneity.

Network-independent learning rates. We start by validating our findings on network-independent learning rates. For this, we solve the logistic regression problem across various network topologies, all using the same learning rate. Specifically, we set the number of nodes as $n = 160$, the feature dimension to $d = 5$, and $\sigma_v = 0.1$. We conduct experiments on ring, 2D-Grid, and exponential graphs. The constant learning rate of each algorithm is set to $1/(4L)$, which is solely determined by $f(x)$ and is unaffected by network topology. We run each simulation ten times, plotting the average performance with a solid line and the standard deviation with a shaded area in Fig. 1. We measure the relative error on the y-axis as $\|\tilde{\mathbf{x}}^k\|^2/\|\mathbf{x}^\star\|^2$. The results show that both Exact-Diffusion and D-SGD converge using a constant and network-independent learning rate, regardless of network topology, which validates our findings.

Linear speedup. We next simulate over the ring graphs of sizes $n = 16$, $n = 160$, and $n = 1600$. For each n, we set the learning rate to $1/(4L)$, a value independent of the network topology. The remaining parameters are identical to those used in the previous experiment. The performance of D-SGD and Exact Diffusion are depicted in Fig. 2. The relative error is shown on the y-axis. When the number of nodes n is increased by tenfold ($10 \times n$), the relative errors of both D-SGD and Exact-Diffusion are reduced by 90% roughly, validating our results about linear speedup.

VII. Conclusion

This paper presents a new analysis for achieving linear speedup in strongly-convex decentralized stochastic optimization using network-independent learning rates. This analysis applies to both Exact-Diffusion and D-SGD. Numerical results are provided to validate our theoretical findings.

Appendix I
Proof of Theorem 1

From (8), it holds that

$$\begin{aligned}\|\tilde{\mathbf{v}}^{k+1}\|^2 &= \|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k - \mathbf{B}\tilde{\mathbf{y}}^k\|^2 \\ &= \|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k\|^2 + \|\mathbf{B}\tilde{\mathbf{y}}^k\|^2 \\ &\quad - 2\langle \mathbf{B}\tilde{\mathbf{y}}^k,\ \tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k\rangle,\end{aligned}$$

and

$$\begin{aligned}\|\tilde{\mathbf{y}}^{k+1}\|^2 &= \|\tilde{\mathbf{y}}^k + \mathbf{B}\tilde{\mathbf{v}}^{k+1}\|^2 \\ &= \|\tilde{\mathbf{y}}^k\|^2 + \|\mathbf{B}\tilde{\mathbf{v}}^{k+1}\|^2 + 2\langle\tilde{\mathbf{y}}^k, \mathbf{B}\tilde{\mathbf{v}}^{k+1}\rangle \\ &= \|\tilde{\mathbf{y}}^k\|^2 + \|\mathbf{B}\tilde{\mathbf{v}}^{k+1}\|^2 + 2\langle\mathbf{B}\tilde{\mathbf{y}}^k, \\ &\qquad \tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k - \mathbf{B}\tilde{\mathbf{y}}^k\rangle.\end{aligned}$$

Summing up $\|\tilde{\mathbf{v}}^{k+1}\|^2$ and $\|\tilde{\mathbf{y}}^{k+1}\|^2$, we get

$$\begin{aligned}&\|\tilde{\mathbf{v}}^{k+1}\|^2_{\mathbf{I}-\mathbf{B}^2} + \|\tilde{\mathbf{y}}^{k+1}\|^2 \\ =&\|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k\|^2 + \|\tilde{\mathbf{y}}^k\|^2_{\mathbf{I}-\mathbf{B}^2}.\end{aligned}$$

Under our assumptions, it holds that

$$\begin{aligned}\|\tilde{\mathbf{y}}^k\|^2_{\mathbf{I}-\mathbf{B}^2} &\le \lambda\|\tilde{\mathbf{y}}^k\|^2, \\ \|\tilde{\mathbf{x}}^{k+1}\|^2 = \|\tilde{\mathbf{v}}^{k+1}\|^2_{\mathbf{W}^2} &\le \|\tilde{\mathbf{v}}^{k+1}\|^2_{\mathbf{W}} = \|\tilde{\mathbf{v}}^{k+1}\|^2_{\mathbf{I}-\mathbf{B}^2}.\end{aligned}$$

Using the above two bounds, taking expectations, and using Assumption 3, we get

$$\begin{aligned}&\mathbb{E}\,\|\tilde{\mathbf{x}}^{k+1}\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^{k+1}\|^2 \\ \le&\,\mathbb{E}\,\|\tilde{\mathbf{v}}^{k+1}\|^2_{\mathbf{I}-\mathbf{B}^2} + \mathbb{E}\,\|\tilde{\mathbf{y}}^{k+1}\|^2 \\ =&\,\mathbb{E}\,\|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star) - \alpha\mathbf{s}^k\|^2 + \mathbb{E}\,\|\tilde{\mathbf{y}}^k\|^2_{\mathbf{I}-\mathbf{B}^2} \\ \le&\,\mathbb{E}\,\|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star)\|^2 \\ &+ \alpha^2\,\mathbb{E}\,\|\mathbf{s}^k\|^2 + \lambda\,\mathbb{E}\,\|\tilde{\mathbf{y}}^k\|^2 \\ \le&\,\mathbb{E}\,\|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star)\|^2\end{aligned}$$

$$+ \alpha^2 n\sigma^2 + \lambda\,\mathbb{E}\,\|\tilde{\mathbf{y}}^k\|^2. \tag{29}$$

Since $0 < \alpha \le \frac{1}{L} \le \frac{2}{L+\mu}$, we have $\frac{2}{\alpha} - \mu \ge L$. Let $\mu' = \mu$, $L' = \frac{2}{\alpha} - \mu$, then $\mathbf{f}$ is also μ'-strongly convex and L'-smooth. As a result, it holds that [34]

$$\begin{aligned}&\langle \nabla\mathbf{f}(\mathbf{x}) - \nabla\mathbf{f}(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star\rangle \\ \ge&\tfrac{\mu L'}{\mu+L'}\|\mathbf{x} - \mathbf{x}^\star\|^2 + \tfrac{1}{\mu+L'}\|\nabla\mathbf{f}(\mathbf{x}) - \nabla\mathbf{f}(\mathbf{x}^\star)\|^2\end{aligned} \tag{30}$$

Expanding the first term in (29), we have

$$\begin{aligned}&\|\tilde{\mathbf{x}}^k - \alpha\nabla\mathbf{f}(\mathbf{x}^k) + \alpha\nabla\mathbf{f}(\mathbf{x}^\star)\|^2 \\ =&\|(\mathbf{x}^k - \mathbf{x}^\star) - \alpha(\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star))\|^2 \\ =&\|\mathbf{x}^k - \mathbf{x}^\star\|^2 - 2\alpha\langle \mathbf{x}^k - \mathbf{x}^\star, \nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star)\rangle \\ &\quad + \alpha^2\|\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star)\|^2 \\ \overset{(a)}{\le}&(1 - 2\alpha\tfrac{\mu' L'}{\mu'+L'})\|\mathbf{x}^k - \mathbf{x}^\star\|^2 \\ &(\alpha^2 - 2\alpha\tfrac{1}{\mu'+L'})\|\nabla\mathbf{f}(\mathbf{x}^k) - \nabla\mathbf{f}(\mathbf{x}^\star)\|^2 \\ =&(1 - \alpha\mu)^2\|\mathbf{x}^k - \mathbf{x}^\star\|^2\end{aligned} \tag{31}$$

where inequality (a) holds because of (30). The result (9) can be attained by substituting (31) to (29).

References

[1] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems (NIPS)*, (Long Beach, CA, USA), pp. 5330–5340, 2017.

[2] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *International Conference on Machine Learning*, (Long Beach, CA, USA), pp. 344–353, 2019.

[3] T. Lin, S. P. Karimireddy, S. Stich, and M. Jaggi, "Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data," in *International Conference on Machine Learning*, (Virtual), pp. 6654–6665, PMLR, 2021.

[4] K. Yuan, Y. Chen, X. Huang, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Decentlam: Decentralized momentum SGD for large-batch deep training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (Montreal, QC, Canada), pp. 3029–3039, 2021.

[5] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin, "Exponential graph is provably efficient for decentralized deep training," in *Advances in Neural Information Processing Systems*, vol. 34, (Virtual), pp. 13975–13987, 2021.

[6] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2021.

[7] C. Cicconetti, M. Conti, and A. Passarella, "A decentralized framework for serverless edge computing in the internet of things," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2166–2180, 2020.

[8] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges," *arXiv preprint arXiv:2211.08413*, 2022.

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

[10] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, CA, USA), pp. 251–255, Oct. 2008.

[11] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3, pp. 323–453, Elsevier, 2014.

[12] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *International Conference on Machine Learning*, (Virtual), pp. 5381–5393, 2020.

[13] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[14] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning-Part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, pp. 708–723, Feb. 2019.

[15] 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*, vol. 67, pp. 4494–4506, Sept. 2019.

[16] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *International Conference on Machine Learning*, (Stockholm, Sweden), pp. 4848–4856, 2018.

[17] K. Yuan, S. A. Alghunaim, and X. Huang, "Removing data heterogeneity influence enhances network topology dependence of decentralized SGD," To appear in *Journal of Machine Learning Research (JMLR)*, 2023. Available on arXiv:2105.08023.

[18] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in Proc. 54th *IEEE Conference on Decision and Control (CDC)*, (Osaka, Japan), pp. 2055–2060, 2015.

[19] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

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

[21] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.

[22] A. Koloskova, T. Lin, and S. U. Stich, "An improved analysis of gradient tracking for decentralized machine learning," in *Advances in Neural Information Processing Systems*, vol. 34, (Virtual), pp. 11422–11435, 2021.

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

[24] K. Yuan, S. A. Alghunaim, B. Ying, and A. H. Sayed, "On the influence of bias-correction on distributed stochastic optimization," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4352–4367, 2020.

[25] K. Huang and S. Pu, "Improving the transient times for distributed stochastic gradient methods," *IEEE Transactions on Automatic Control*, 2022.

[26] S. A. Alghunaim and K. Yuan, "An enhanced gradient-tracking bound for distributed online stochastic convex optimization," *arXiv preprint arXiv:2301.02855*, 2023.

[27] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, pp. 460–497, Apr. 2014.

[28] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, (Maui, HI, US), pp. 5453–5458, IEEE, 2012.

[29] S. A. Alghunaim, E. K. Ryu, K. Yuan, and A. H. Sayed, "Decentralized proximal gradient algorithms with linear convergence rates," *IEEE Transactions on Automatic Control*, vol. 66, pp. 2787–2794, June 2021.

[30] L. Guo, X. Shi, J. Cao, and Z. Wang, "Decentralized inexact proximal gradient method with network-independent stepsizes for convex composite optimization," *IEEE Transactions on Signal Processing*, 2023.

[31] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Mathematical Programming*, vol. 187, no. 1, pp. 409–457, 2021.

[32] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[33] A. H. Sayed, "Adaptation, learning, and optimization over neworks.," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[34] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer, 2013.