

# Achieving Near-Optimal Oracle Complexity in Decentralized Stochastic Optimization With Channel Noise

Soham Mukherjee  and Mrityunjoy Chakraborty , *Senior Member, IEEE*

**Abstract**—We study a decentralized nonconvex stochastic optimization problem in which a group of agents/nodes seek to minimize a global objective function that can be expressed as a sum of local component functions, such that each node in the network has access to exactly one component function. The nodes collaborate with their neighbors by sharing their estimates over communication links that are assumed to be corrupted by additive noise. To address this problem, we propose a computationally efficient and robust algorithm that builds on a probabilistic technique for stochastic gradient computation, which we believe will be applicable to a wide range of problems in decentralized information processing, learning, and control. Specifically, we show that the proposed method achieves an oracle complexity (computational complexity) of  $O(1/\epsilon^2)$  for smooth and nonconvex functions with stochastic gradients, which is known to be sharp for its respective function class, and is an improvement over the computational cost obtained in previous works. In addition, we retain the  $O(1/\epsilon^3)$  rate for the communication cost, which is at par with the communication cost obtained in previous works. We also show how the proposed algorithm has robust performance in environments with unreliable computational resources. Finally, the theoretical findings are validated via numerical experiments.

**Index Terms**—Channel noise, decentralized optimization, nonconvex optimization, stochastic gradients.

## SUMMARY OF VARIABLES IN ALGORITHM 1

Quantity	Description
$K$	Number of iterations.
$\chi_i(t)$	Bernoulli random variable drawn by node $i$ at iteration $t$ .
$\theta$	Probability with which $\chi_i(t)$ takes the value 1.
$\mathbf{x}_i(t)$	Node $i$ 's estimate at iteration $t$ .
$\tilde{\mathbf{x}}_{ji}(t)$	Node $j$ 's noise-corrupted estimate received by node $i$ at iteration $t$ .
$\mathbf{h}_i(t)$	Quantity used by node $i$ as a stochastic approximation of its local gradient $\nabla f_i(\mathbf{x}_i(t))$ at iteration $t$ .

Received 28 November 2024; accepted 24 January 2025. Date of publication 7 February 2025; date of current version 20 June 2025. This work was supported by the ANRF (formerly SERB), Govt. of India. Recommended by Associate Editor Y. Wang. (*Corresponding author: Mrityunjoy Chakraborty.*)

The authors are with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721302, India (e-mail: sohammukherjee5898@gmail.com; mrityun@ece.iitkgp.ac.in).

Digital Object Identifier 10.1109/TCNS.2025.3539495

$g_i(\mathbf{x}_i(t))$	Stochastic gradient computed by node $i$ at iteration $t$ if and only if $\chi_i(t) = 1$ .
$\eta$	Step-size parameter associated with the gradient update term.
$c$	Step-size parameter associated with the consensus update term.

## I. INTRODUCTION

WE CONSIDER a distributed stochastic optimization problem in which a collection of  $m$  nodes connected over a network try to minimize  $f(x) = (1/m) \sum_{i=1}^m f_i(x)$ , such that each component function  $f_i(\cdot)$  is known exclusively to node  $i$ . In particular, we assume that only node  $i$  can access stochastic gradients associated with the function  $f_i(\cdot)$ . The goal is to develop an algorithm in which the nodes cooperate with each other by performing local computations and exchanging relevant information to estimate the global minimizer  $x^* = \arg \min_x f(x)$  or compute a first-order stationary point depending on the function class. Such a framework captures a wide variety of practical problems that frequently arise in distributed control [1], distributed machine learning [2], Big Data analytics [3], IoT applications [4], power networks [5] amongst others. Several distributed algorithms have been proposed in recent past to solve this problem, by building on centralized schemes, such as gradient descent-based algorithms [6], [7], [8]; alternating direction method of multipliers [9], [10]; Newton methods [11], [12]; and primal-dual methods [13], [14] to name a few. In this article, we focus our attention on gradient descent-based algorithms owing to their simplicity, low overhead, and widespread use. However, even among first-order gradient descent-based algorithms, the vast majority assume perfect communication between nodes without any channel noise [15], [16], [17], [18]; noise caused by quantization of transmitted information [19], [20], [21], [22]; or noise introduced for privacy preservation [23], [24], [25]—an ideal assumption hardly satisfied in real-world settings. Among the works that took into account noise effects, notable are [15] where the authors analyze the effects of channel noise (both under the zero-mean bounded variance noise assumption as well as under a Markovian noise sequence assumption) in the average consensus problem; [16], [17], and [18] in which the authors consider communication noise in the decentralized optimization setting where the communication noise is modeled as zero-mean bounded variance additive noise; and [19], [20], [21], and [22] in which the authors consider decentralized optimization problems with quantized communication between nodes where

TABLE I

COMPARISON WITH EXISTING METHODS THAT SOLVE DECENTRALIZED OPTIMIZATION PROBLEMS IN THE PRESENCE OF ADDITIVE ZERO-MEAN AND BOUNDED VARIANCE COMMUNICATION NOISE

Reference	Problem Type	Gradient Type	Gradient Computations	Communication Rounds	Gradient Computations (centralized counterpart)
[19]	Non-smooth, strongly-convex	Exact	$O(1/\epsilon^3)$	$O(1/\epsilon^3)$	$O(1/\epsilon)$ , [30]
[19]	Non-smooth, convex	Exact	$O(1/\epsilon^4)$	$O(1/\epsilon^4)$	$O(1/\epsilon^2)$ , [30]
[20]	Smooth, strongly-convex	Exact	$O(1/\epsilon^{2+\delta_1})$ <sup>2</sup>	$O(1/\epsilon^{2+\delta_1})$ <sup>2</sup>	$O(\log(1/\epsilon))$ , [30]
[21]	Smooth, strongly-convex	Stochastic	$O(1/\epsilon^{2+\delta_2})$ <sup>3</sup>	$O(1/\epsilon^{2+\delta_2})$ <sup>3</sup>	$O(1/\epsilon)$ , [30]
[21]	Smooth, nonconvex	Stochastic	$O(1/\epsilon^3)$	$O(1/\epsilon^3)$	$O(1/\epsilon^2)$ , [28]
[22]	Smooth, strongly-convex	Exact	$O((\log(1/\epsilon))^4/\epsilon^2)$	$O((\log(1/\epsilon))^4/\epsilon^2)$	$O(\log(1/\epsilon))$ , [30]
[16]	Smooth, strongly-convex	Exact	Converges to a fixed neighborhood around the optimum. Arbitrarily small $\epsilon$ -accuracy cannot be achieved.		$O(\log(1/\epsilon))$ , [30]
[17]	Smooth, strongly-convex	Exact	$O(1/\epsilon^2)$	$O(1/\epsilon^2)$	$O(\log(1/\epsilon))$ , [30]
[18] <sup>4</sup>	Smooth, nonconvex	Exact	Not explicit	Not explicit	$O(1/\epsilon)$ , [31]
[18] <sup>4</sup>	Smooth, nonconvex	Stochastic	Not explicit	Not explicit	$O(1/\epsilon^2)$ , [28]
<b>This work</b>	Smooth, nonconvex	Stochastic	$O(1/\epsilon^2)$	$O(1/\epsilon^3)$	$O(1/\epsilon^2)$ , [28]

again the quantization noise is modeled as a separate additive noise component having zero-mean and bounded variance. All of these works [16], [17], [18], [19], [20], [21], [22] consider the *Noisy Consensus + Stochastic Gradient Descent (SGD)* framework for developing their algorithms, with the works in [16] and [18] incorporating an additional gradient tracking step to enable individual nodes in the network to estimate the global gradient. The treatments in [16], [17], [19], [20], and [22], however, restrict them to a setting where exact gradients are available. This requirement is lifted in [18] and [21] in which the authors use a stochastic setting. Wang and Başar [18] derived conditions needed to ensure *almost sure* convergence under the zero-mean bounded variance communication noise assumption, with their results for *almost sure* convergence also holding under a more relaxed assumption that allows the noise variance to grow with increasing number of iterations under certain conditions. However, explicit characterization of the computational and communication costs is not provided in [18]. Reiszadeh et al. [21] proposed a *QuanTimed-DSGD* algorithm that considers zero-mean bounded variance communication noise and deploys a deadline based stochastic gradient computation framework to mitigate the effects of stragglers. It is shown that this algorithm requires  $O(1/\epsilon^{2+\delta})$  (where  $\delta > 0$  can be made arbitrarily small) stochastic gradient computations and communication rounds to compute an  $\epsilon$ -accurate solution for smooth and strongly convex functions ( $O(1/\epsilon^3)$  in the case of smooth, nonconvex functions<sup>1</sup>).

The computational cost of  $O(1/\epsilon^{2+\delta})$  ( $O(1/\epsilon^3)$ ) incurred by the *QuanTimed-DSGD* algorithm, however, makes it suboptimal vis-a-vis the centralized SGD algorithm that requires only  $O(1/\epsilon)$  ( $O(1/\epsilon^2)$ ) stochastic gradient computations to compute an  $\epsilon$ -accurate solution. In fact, in all of the works [16], [17], [19], [20], [21], [22], a common observation is that the *introduction of zero-mean bounded variance communication noise in the problem causes the computational complexity to increase as compared to their centralized counterparts* (i.e., the  $m = 1$  case for which there is no communication noise), as displayed in Table I. This increase in computational cost is not observed in the noiseless perfect communication setting, but appears to be a bottleneck when communication noise is introduced in the system. The study in this article is primarily motivated by

this gap in the computational complexity, especially since the volume of data to be processed has increased multifold over the last few years. Moreover, recent studies have shown that complex models, such as overparameterized neural networks, perform well both in theory and in practice [26], [27]. This combination of data deluge and increasing model complexity has pushed up the demand on computational resources enormously. This also poses a significant limitation in resource-constrained or unreliable environments where there might be occasional failures in carrying out such computationally intensive operations, thus necessitating the development of algorithms that have robust performance in such challenging settings.

In this work, we propose a computationally efficient and robust decentralized algorithm that operates with noisy communication links and incorporates a probabilistic technique for stochastic gradient computation in the widely used *Noisy Consensus + SGD* framework. Specifically, it allows individual nodes to randomly skip the stochastic gradient computation step in a given iteration with some probability.

Our main contributions are as follows. 1) We show that the computational complexity of the proposed method achieves at par dependence on the required accuracy  $\epsilon$  with its centralized counterpart (the centralized SGD Algorithm with a first-order oracle [28]). This is a marked improvement over previous works, where it has been seen that the introduction of noise in the information exchange process causes the computational complexity to increase as compared to the computational complexity that can be achieved using centralized versions of the algorithm (see Table I). Specifically, for smooth and nonconvex functions, the proposed method requires  $O(1/m\epsilon^3)$  communication rounds and  $O(1/m\epsilon^2)$  stochastic gradient computations. The  $1/\epsilon^2$  dependence of stochastic gradient computations in the proposed method improves the  $1/\epsilon^3$  dependence obtained in [21], resulting in significant computational savings while maintaining at par dependence on  $\epsilon$  for the number of iterations and the number of communication rounds as in [21]. This decrease in the number of stochastic gradient computations is achieved by carefully by choosing the probability with which the gradient computations step is skipped in a given iteration. 2) In addition, the  $O(1/\epsilon^2)$

<sup>2</sup> $\delta_1 > 0$  can be made arbitrarily small.<sup>3</sup> $\delta_2 > 0$  can be made arbitrarily small.<sup>4</sup>Analysis is focused on establishing *almost sure* convergence and does not provide finite-time results.<sup>1</sup>Refer to Section III-B for the definition of an  $\epsilon$ -accurate solution for smooth decentralized nonconvex optimization problems.

dependence on required accuracy  $\epsilon$  for the number of stochastic gradient computations is sharp, in the sense that it matches the lower bound for the number of stochastic gradient computations required to compute an  $\epsilon$ -accurate solution [29], although in the “expected” sense since the actual number of stochastic gradient computations in our proposed method is a random variable. We, however, show, by developing high-probability bounds, that the actual number of stochastic gradient computations is concentrated around its expected value. 3) We also show how, in the noisy communication setting, the proposed algorithm is robust to environments that have unreliable computational resources (e.g., frequent power outages and straggler nodes), where successful stochastic gradient computation in an iteration occurs with some probability less than 1. Specifically, we show that in the presence of zero-mean and bounded variance communication noise, the overall convergence of the algorithm remains unaffected as long as the probability of successful stochastic gradient computation is above a small threshold, whose value we explicitly characterize in subsequent sections. 4) Lastly, the presence of the  $1/m$  factor in the computational and communication costs indicates that the proposed method achieves a linear speed-up proportional to the number of nodes in the network  $m$ , which is an improvement over the result obtained in [21], where this effect is not fully realized.

*Notations:* We use  $\mathbf{I}_p$  to denote the  $p \times p$  identity matrix,  $\mathbf{1}_p$  to denote the  $p$ -dimensional column vector of all 1s,  $\mathbf{0}_p$  to denote the  $p$ -dimensional column vector of all 0s, and  $\|\cdot\|$  to denote the 2-norm for vectors. For matrices,  $\|\cdot\|$  denotes the spectral norm and  $\|\cdot\|_F$  denotes the Frobenius norm. The expression  $\Pr\{\mathcal{E}\}$  is used to denote the probability of occurrence of the event  $\mathcal{E}$ . The expression  $\text{Bernoulli}(p)$  is used to denote the Bernoulli distribution which, when sampled, outputs 1 with probability  $p$  and 0 with probability  $1-p$ , with  $0 \leq p \leq 1$ . Lastly, we use  $a \wedge b$  to denote  $\max\{a, b\}$ .

## II. PROBLEM FORMULATION

### A. Decentralized Stochastic Optimization Problem

We consider a setting where the network of  $m$  nodes wishes to solve the following unconstrained optimization problem:

$$\text{P1: minimize}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x}). \quad (1)$$

Here, the function  $f_i(\cdot)$  is known only to node  $i$ . Specifically, we assume that only node  $i$  can compute stochastic gradients of the function  $f_i(\cdot)$ . When node  $i$  queries a point  $\mathbf{x}_i(t) \in \mathbb{R}^d$ , at some time  $t \geq 0$ , the corresponding stochastic first-order oracle returns a random vector  $\mathbf{g}_i(\mathbf{x}_i(t)) \in \mathbb{R}^d$  satisfying

$$E[\mathbf{g}_i(\mathbf{x}_i(t)) | \mathbf{x}_i(t)] = \nabla f_i(\mathbf{x}_i(t)) \quad (2a)$$

$$E[\|\mathbf{g}_i(\mathbf{x}_i(t)) - \nabla f_i(\mathbf{x}_i(t))\|^2 | \mathbf{x}_i(t)] \leq \sigma_g^2 \quad (2b)$$

where  $\sigma_g$  is a positive constant. Equation (2a) implies that the stochastic approximation  $\mathbf{g}_i(\mathbf{x}_i)$  is an unbiased estimate of the true gradient  $\nabla f_i(\mathbf{x}_i)$ , and (2b) bounds the variance of the stochastic gradient. These conditions are very common in stochastic optimization literature (see, e.g., [18], [21], [28], [32], and [33]).

### B. Network Model

We consider a network of  $m$  nodes, which communicate over a static undirected graph  $G = (V, E)$  with  $V = \{1, \dots, m\}$  representing the individual nodes and  $E$  representing the connections between the nodes. Two nodes  $i$  and  $j$  are said to be connected if and only if  $(i, j) \in E$ . We further associate a  $m \times m$  weight matrix  $\mathbf{W}$  with the graph  $G$ , with  $W_{ij}$  denoting the weight the node  $i$  applies to the signal received from node  $j$ . For each  $i \in V$ , we use  $\mathcal{N}_i$  to denote the neighbors of node  $i$ , or more formally  $\mathcal{N}_i = \{j : j \neq i, W_{ij} > 0\}$ . For all  $i \in V$ , we assume that node  $i$  can exchange information with some  $j \in V$  if and only if  $j \in \mathcal{N}_i$ . Our assumptions about the graph  $G$  and the network matrix  $\mathbf{W}$  are stated below.

**Assumption 1:** The graph  $G$  is strongly connected. The weight matrix  $\mathbf{W}$  associated with the undirected graph  $G$  is Hermitian, doubly stochastic and satisfies  $W_{ii} > 0 \forall i \in V$ ,  $W_{ij} \geq 0 \forall i, j \in V$ , and  $W_{ij} > 0$  if and only if  $(i, j) \in E$ . These assumptions imply that the spectral norm of the matrix  $\mathbf{W} - (1/m)\mathbf{1}_m\mathbf{1}_m^T$ , which we denote by  $\omega$ , is strictly smaller than 1. This means that for any  $\mathbf{x} \in \mathbb{R}^m$ , we have the following contraction  $\|\mathbf{W}\mathbf{x} - \mathbf{1}_m\bar{x}\| \leq \omega\|\mathbf{x} - \mathbf{1}_m\bar{x}\|$ , where  $\bar{x} = (1/m)\mathbf{1}_m^T\mathbf{x}$ .

### C. Communication Noise Model

Consider two neighboring nodes  $i$  and  $j$ , where node  $i$  sends to node  $j$  a quantity  $\mathbf{x}_i(t) \in \mathbb{R}^d$ , at a time  $t \geq 0$ , which, in an imperfect communication setting, is corrupted by additive noise and is received as  $\tilde{\mathbf{x}}_{ij}(t)$ , given by

$$\tilde{\mathbf{x}}_{ij}(t) = \mathbf{x}_i(t) + \mathbf{n}_{ij}(t) \quad (3)$$

where  $\mathbf{n}_{ij}(t) \in \mathbb{R}^d$  represents the noise introduced in the link from node  $i$  to node  $j$  at time  $t$ . We assume that for all  $i \in V$ , the following conditions hold:

$$E[\mathbf{n}_{ij}(t) | \mathbf{x}_i(t)] = \mathbf{0} \quad (4a)$$

$$E[\|\mathbf{n}_{ij}(t)\|^2 | \mathbf{x}_i(t)] \leq \sigma_c^2 \quad (4b)$$

where  $\sigma_c$  is a positive constant. We also assume that for any  $t \geq 0$ , the set of random variables  $\{\mathbf{n}_{ij}(t)\}, (i, j) \in E$ , are pairwise independent conditioned on  $\mathbf{x}_1(t), \dots, \mathbf{x}_m(t)$ .

## III. PROPOSED METHOD AND CONVERGENCE RESULT

### A. Algorithm Description

Let  $\mathbf{x}_i(t)$  denote node  $i$ 's estimate at iteration  $t$ . Also, let  $\chi_i(t) \sim \text{Bernoulli}(\theta)$  ( $0 < \theta < 1$ ) be a random variable associated with node  $i$  at time  $t$ , which is independent of all other sources of randomness arising due to stochastic gradients and channel noise. Using the above, we summarize the proposed method in Algorithm 1.

In (5a) and (5b), the algorithm calculates  $\mathbf{h}_i(t)$  and uses it as an estimate for the local gradient in (5c), which is a standard *Noisy Consensus + SGD* update step (see [17], [19], [20], [21], and [22]). Refer to the Nomenclature for a list of variables used in Algorithm 1.

Focusing on (5a) and (5b), we see that  $\mathbf{g}_i(\mathbf{x}_i(t))$  is itself a stochastic approximation of the true gradient  $\nabla f_i(\mathbf{x}_i(t))$ . The

**Algorithm 1:** Proposed algorithm**Input:**  $K, \theta, \eta, c$ .**Initialization:** For each  $i \in V$ , initialize  $\mathbf{x}_i(0) = \mathbf{0}_d$ .**for**  $t = 0, 1, \dots, K - 1$  **do**  **for** each  $i \in V$  **do**    1. Draw  $\chi_i(t) \sim \text{Bernoulli}(\theta)$ .    2. **if**  $(\chi_i(t) == 1)$  **then**      Compute  $\mathbf{g}_i(\mathbf{x}_i(t))$  and

      set  $\mathbf{h}_i(t) = (1/\theta)\mathbf{g}_i(\mathbf{x}_i(t))$ .      (5a)

**else**

      Set  $\mathbf{h}_i(t) = \mathbf{0}_d$ .      (5b)

**end if**  3. Node  $i$  sends  $\mathbf{x}_i(t)$  to all  $j \in \mathcal{N}_i$  and receives     $\tilde{\mathbf{x}}_{ji}(t)$  from all  $j \in \mathcal{N}_i$ .

  4. Update  $\mathbf{x}_i(t+1) = \underbrace{\mathbf{x}_i(t) - \eta\mathbf{h}_i(t)}_{\text{SGD}} + c \underbrace{\sum_{j \in \mathcal{N}_i} W_{ij}(\tilde{\mathbf{x}}_{ji}(t) - \mathbf{x}_i(t))}_{\text{NoisyConsensus}}$ .      (5c)

**end for****end for**

gradient estimator  $\mathbf{h}_i(t)$  adds another layer of randomness by randomly choosing whether or not to compute a stochastic gradient at time  $t$ . Note that (5a) and (5b) can be written compactly as  $\mathbf{h}_i(t) = \frac{\chi_i(t)}{\theta} \mathbf{g}_i(\mathbf{x}_i(t))$ . Then, we have the following result on  $\mathbf{h}_i(t)$ , which follows trivially from (2a), (2b), and statistical independence of  $\chi_i(t)$  from all other sources of randomness at node  $i$ , with  $E[\chi_i(t)] = \theta$  and  $E[\chi_i^2(t)] = \theta$ .

**Lemma 1:** The following holds for all  $t \geq 0$ :

$$E[\mathbf{h}_i(t) | \mathbf{x}_i(t)] = \nabla f_i(\mathbf{x}_i(t)) \quad (6a)$$

$$E\left[\|\mathbf{h}_i(t) - \nabla f_i(\mathbf{x}_i(t))\|^2 | \mathbf{x}_i(t)\right] \leq \frac{1}{\theta} \sigma_g^2 + \left(\frac{1}{\theta} - 1\right) \|\nabla f_i(\mathbf{x}_i(t))\|^2. \quad (6b)$$

Equation (6a) shows that  $\mathbf{h}_i(t)$  is an unbiased estimator of the exact gradient  $\nabla f_i(\mathbf{x}_i(t))$  conditioned on  $\mathbf{x}_i(t)$ . However, comparing (2b) and (6b), we observe that  $\mathbf{h}_i(t)$  has higher variance than  $\mathbf{g}_i(\mathbf{x}_i(t))$ . (The constant term  $\sigma_g^2$  is scaled up by a factor of  $1/\theta$  and there is an extra  $(1/\theta - 1)\|\nabla f_i(\mathbf{x}_i(t))\|^2$  term.) However notice that at time  $t$ , the stochastic gradient is computed with probability  $\theta$  (where, as we shall see in the next section,  $\theta$  will be chosen to be of the form  $\theta = \Theta(1/(m^\mu K^\tau))$ , with  $\mu > 0$  and  $\tau > 0$ ). This means that the stochastic gradient computation step is skipped in many iterations and  $\mathbf{h}_i(t)$  is set to a zero vector in such iterations. Thus, there are two opposing forces in play: higher variance associated with the gradient estimate  $\mathbf{h}_i(t)$  and lower expected computational cost per iteration (where the expectation is taken on  $\chi_i(t)$ ) by virtue of skipping

the stochastic gradient computation step in some iterations. This raises the question of whether such an operation reduces the overall computational complexity, by requiring fewer stochastic gradient computations totaled over all iterations, while at the same time ensuring that the increased variance of the gradient estimate  $\mathbf{h}_i(t)$  does not have any detrimental effect on the overall convergence of the algorithm. We answer this question positively in the next section.

**B. Convergence Result for the General Nonconvex Case**

In this section, we make the following assumptions about the local component functions and the global objective function.

**Assumption 2:** The local component functions  $f_i$ 's are  $L$ -smooth, i.e., they have  $L$ -Lipschitz continuous gradients. This means, for all  $i \in V$  and for all  $\mathbf{x}, \mathbf{y} \in R^d$ , we have

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \quad (7)$$

Note that this assumption implies that

$$f_i(\mathbf{x}) \leq f_i(\mathbf{y}) + \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2. \quad (8)$$

**Assumption 3:** The global objective function  $f(\cdot)$  is lower bounded by  $f^*$ , i.e., for all  $\mathbf{x} \in R^d$ , we have

$$f(\mathbf{x}) \geq f^*. \quad (9)$$

We introduce a quantity called the network-averages gradient disagreement, which is defined for all  $\mathbf{x} \in R^d$  as

$$\mathcal{NAGD}(\mathbf{x}) \triangleq \frac{1}{m} \sum_{i=1}^m \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2. \quad (10)$$

The function  $\mathcal{NAGD}(\mathbf{x})$  captures the heterogeneity of the data available to the different nodes in the network at  $\mathbf{x}$ . We make the following assumption on  $\mathcal{NAGD}(\cdot)$ .

**Assumption 4:** For all  $\mathbf{x} \in R^d$ , the network-averaged gradient disagreement satisfies

$$\mathcal{NAGD}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \sigma_G^2 \quad (11)$$

for some  $\sigma_G^2 < \infty$ . These assumptions are very common in decentralized stochastic nonconvex optimization (see, e.g., [21] and [32]).

**Definition 1:** Consider problem P1. Under Assumptions 1–4, an algorithm is said to compute an  $\epsilon$ -accurate solution at iteration  $K$  if the following conditions are satisfied:

$$\frac{1}{K} \sum_{t=0}^{K-1} E[\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \leq \epsilon \quad (12a)$$

$$\frac{1}{mK} \sum_{t=0}^{K-1} \sum_{i=1}^m E[\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2] \leq \epsilon \quad (12b)$$

where  $\bar{\mathbf{x}}(t) = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i(t)$ . This is a commonly used metric while characterizing the performance of decentralized nonconvex optimization algorithms (see, for example, the works [21] and [32]). We then have the following result characterizing the

performance of the proposed method in the general nonconvex setting.

**Theorem 1:** Let Assumptions 1–4 hold and  $\alpha, \beta$ , and  $\gamma$  be constants. Then, Algorithm 1 with the parameter choices

$$\eta = \frac{m^\alpha}{K^{2/3}}, c = \frac{m^\beta}{K^{1/2}}, \theta = \frac{m^\gamma}{K^{1/3}} \quad (13)$$

satisfies the following bounds for  $K \geq K_0$ :

$$\begin{aligned} \frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] &\leq O\left(\frac{m^{-\alpha}}{K^{1/3}}\right) [f(\bar{\mathbf{x}}(0)) - f^*] \\ &+ O\left(\frac{m^{(2\beta-1-\alpha)}}{K^{1/3}} + \frac{m^\beta}{K^{1/2}}\right) \sigma_c^2 \\ &+ O\left(\frac{m^{(\alpha-\gamma-1)}}{K^{1/3}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{1/2}}\right) \sigma_g^2 \\ &+ O\left(\frac{m^{(\alpha-\gamma-1)} + m^{2(\alpha-\beta)}}{K^{1/3}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{1/2}} + \frac{m^\alpha}{K^{2/3}}\right) \sigma_G^2 \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{1}{mK} \sum_{t=0}^{K-1} \sum_{i=1}^m E [\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2] \\ \leq O\left(\frac{m^{2(\alpha-\beta)}}{K^{1/3}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{1/2}}\right) \frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\ + O\left(\frac{m^{2(\alpha-\beta)}}{K^{1/3}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{1/2}}\right) \sigma_G^2 + O\left(\frac{m^{(2\alpha-\beta-\gamma)}}{K^{1/2}}\right) \sigma_g^2 \\ + O\left(\frac{m^\beta}{K^{1/2}}\right) \sigma_c^2 \end{aligned} \quad (15)$$

where  $K_0$  is given by

$$K_0 = \max\{1, K_1, K_2\} \quad (16)$$

with

$$K_1 = \left( \frac{36m^{(2\alpha-2\beta)}L^2}{(1-\omega)^2} + \frac{192m^{(2\alpha-2\beta-\gamma)}L^2}{(1-\omega)} \right)^3 \quad (17)$$

$$K_2 = (8m^{\alpha-\gamma-1}L + 8m^\alpha L)^3. \quad (18)$$

Moreover, for any node  $i \in \{1, \dots, m\}$ , the expected number of stochastic gradient computations up to time  $K$  is given by

$$\sum_{s=0}^{K-1} E [\chi_i(s)] = m^\gamma K^{2/3} \quad (19)$$

with the actual number of stochastic gradient computations satisfying the following high-probability bound:

$$\Pr\left\{ \sum_{s=0}^{K-1} \chi_i(s) \leq 2m^\gamma K^{2/3} \right\} \geq 1 - \exp\left(-2m^{2\gamma} K^{1/3}\right). \quad (20)$$

**Proof:** Given in Section IV.

It is to be noted that the parameters  $\alpha, \beta$ , and  $\gamma$  in Theorem 1 are left as free parameters and that the slowest decaying term is

$O(1/K^{1/3})$ . We choose  $\alpha, \beta$ , and  $\gamma$  to obtain the best possible dependence on  $m$  in the  $O(1/K^{1/3})$  term in the following result.

**Corollary 1:** Choosing  $\alpha = 1/3, \beta = 1/2$ , and  $\gamma = -1/3$  in Theorem 1 gives us the following bound for  $K \geq K_0$ :

$$\frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \leq O\left(\frac{1}{m^{1/3}K^{1/3}} + \frac{m^{1/2}}{K^{1/2}}\right) \quad (21)$$

$$\frac{1}{mK} \sum_{t=0}^{K-1} \sum_{i=1}^m E [\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2] \leq O\left(\frac{1}{m^{1/3}K^{1/3}} + \frac{m^{1/2}}{K^{1/2}}\right). \quad (22)$$

If we let  $K_\epsilon$  denote the number of iterations required for Algorithm 1 to obtain an  $\epsilon$ -accurate solution according to Definition 1, then under the conditions in Corollary 1, we have

$$K_\epsilon \leq O\left(\frac{1}{m\epsilon^3} \wedge \frac{m}{\epsilon^2}\right). \quad (23)$$

Consequently, the expected number of stochastic gradient computations at any given node  $i \in V$  is given by

$$\sum_{s=0}^{K-1} E [\chi_i(s)] = \frac{K_\epsilon^{2/3}}{m^{1/3}} \leq O\left(\frac{1}{m\epsilon^2} \wedge \frac{m^{1/3}}{\epsilon^{4/3}}\right). \quad (24)$$

Thus, if  $\epsilon$  is chosen to be sufficiently small while  $m$  is moderate (as is typically the case), the communication and computational costs scale as  $O(1/(m\epsilon^3))$  and  $O(1/(m\epsilon^2))$ , respectively, thus improving over the  $O(1/(\epsilon^3))$  computational and communication costs obtained in [21].

Focusing specifically on the computational complexity, it is interesting to note that a properly tuned value of  $\theta$  in Algorithm 1 has allowed us to obtain a  $1/\epsilon^2$  dependence on the required accuracy  $\epsilon$  for the number of stochastic gradient computations. This matches the  $1/\epsilon^2$  dependence obtained in [28] for the number of stochastic gradient computations in centralized first-order SGD with smooth and nonconvex functions. In addition, the  $1/\epsilon^2$  dependence is also known to be sharp, in the sense that it matches the  $1/\epsilon^2$  dependence obtained in the lower bound established in [29] for the number of stochastic gradient computations in smooth and nonconvex optimization. Thus, the randomized gradient computation framework in Algorithm 1 has allowed us to achieve near-optimal dependence on  $\epsilon$  in the computational complexity. We say near-optimal because we state our results about the computational complexity in terms of the expected number of stochastic gradient computations, which is to be contrasted with the fact that the results in [28] and [29] consider a deterministic number of stochastic gradient computations. However, we make up for this by providing a high-probability bound in (20), which shows that the actual number of stochastic gradient computations is close to its expected value in the order sense with high probability.

Moreover, the presence of the  $1/m$  factor in both the communication and computational costs indicates that the proposed method is able to achieve a linear speed-up proportional to the number of nodes in the network. This is expected and has been seen in previous works, which assume noiseless perfect communication settings (see, for example, [32]). The result obtained in [21] uses step-sizes, which are independent of the

network size  $m$  and is therefore not able to fully capture this effect, as opposed to the current work and the work in [32], which are able to capture this effect by using step-sizes which are dependent on  $m$ .

#### IV. CONVERGENCE ANALYSIS

For our analysis, we use a strategy similar to the ones found in [19] and [21]. At a high level, we try to bound the disagreements between the estimates computed by the various nodes and the network average of these estimates. We then analyze how well the network average approximates a first-order stationary point. These intermediate results are then combined to obtain our final result in Theorem 1.

Consider the update step in (5c). Separating out the channel noise term, we rewrite (5c) as

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + c \sum_{j \in \mathcal{N}_i} W_{ij} (\mathbf{x}_j(t) - \mathbf{x}_i(t)) - \eta \mathbf{h}_i(t) \\ &\quad + c \underbrace{\sum_{j \in \mathcal{N}_i} W_{ij} \mathbf{n}_{ji}(t)}_{\mathbf{n}_i(t)}. \end{aligned} \quad (25)$$

We now stack the variables associated with various nodes to define the following  $d \times m$  matrices:

$$\mathbf{X}(t) = [\mathbf{x}_1(t), \dots, \mathbf{x}_m(t)] \in R^{d \times m} \quad (26a)$$

$$\mathbf{H}(t) = [\mathbf{h}_1(t), \dots, \mathbf{h}_m(t)] \in R^{d \times m} \quad (26b)$$

$$\mathbf{N}(t) = [\mathbf{n}_1(t), \dots, \mathbf{n}_m(t)] \in R^{d \times m}. \quad (26c)$$

Then, from the communication noise model described in Section II-C, the following holds for all  $i \in V$ :

$$E[\mathbf{n}_i(t) | \mathbf{X}(t)] = 0 \quad (27a)$$

$$E[\|\mathbf{n}_i(t)\|^2 | \mathbf{X}(t)] \leq \sigma_c^2 \quad (27b)$$

with the set of random variables  $\{\mathbf{n}_i(t)\}$ ,  $i \in V$  being pairwise-independent conditioned on  $\mathbf{X}(t)$ .

Further, let us define

$$F(\mathbf{X}(t)) = \sum_{i=1}^m f_i(\mathbf{x}_i(t)). \quad (28)$$

Then, we denote

$$\nabla F(\mathbf{X}(t)) = [\nabla f_1(\mathbf{x}_1(t)), \dots, \nabla f_m(\mathbf{x}_m(t))] \in R^{d \times m}. \quad (29)$$

In addition, we define a lazy version of  $W$  as follows:

$$\mathbf{W}_c = (1-c)\mathbf{I}_m + c\mathbf{W}. \quad (30)$$

It is easy to see that the spectral norm of the matrix  $\mathbf{W}_c - (1/m)\mathbf{1}_m\mathbf{1}_m^T$ , which we denote by  $\omega_c$ , satisfies  $\omega_c \leq (1-c) + c\omega$ .

With the above quantities defined, the update step (5c) of Algorithm 1 can be written compactly as

$$\mathbf{X}(t+1) = \mathbf{X}(t)\mathbf{W}_c^T - \eta\mathbf{H}(t) + c\mathbf{N}(t). \quad (31)$$

We also define the following network averages:

$$\overline{\nabla F}(\mathbf{X}(t)) = \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}_i(t)), \quad \bar{\mathbf{x}}(t) = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i(t),$$

$$\bar{\mathbf{h}}(t) = \frac{1}{m} \sum_{i=1}^m \mathbf{h}_i(t), \quad \bar{\mathbf{n}}(t) = \frac{1}{m} \sum_{i=1}^m \mathbf{n}_i(t).$$

#### A. Auxiliary Results

**Lemma 2 (Evolution of the network average):** The following holds at all times  $t \geq 0$ :

$$\bar{\mathbf{x}}(t+1) = \bar{\mathbf{x}}(t) - \eta \bar{\mathbf{h}}(t) + c \bar{\mathbf{n}}(t). \quad (32)$$

**Proof:** The proof is given in Appendix A.

**Lemma 3 (Disagreement bound):** Under the assumption that  $c \leq 1$  in (5c), the following holds for all  $t \geq 0$ :

$$\begin{aligned} E[\|\mathbf{X}(t+1) - \bar{\mathbf{x}}(t+1)\mathbf{1}_m^T\|_F^2] &\leq (1 - (1-\omega)c) E[\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\ &\quad + \left(\frac{4\eta^2 L^2}{(1-\omega)c} + \frac{2\eta^2 L^2}{\theta}\right) E[\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\ &\quad + \left(\frac{4\eta^2}{(1-\omega)c} + \frac{2\eta^2}{\theta}\right) E[\|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2] \\ &\quad + \frac{\eta^2 m \sigma_g^2}{\theta} + mc^2 \sigma_c^2. \end{aligned} \quad (33)$$

**Proof:** The proof is given in Appendix A.

**Lemma 4 (Effect of the network average):** The following holds at all times  $t \geq 0$ :

$$\begin{aligned} E[f(\bar{\mathbf{x}}(t+1))] &\leq E[f(\bar{\mathbf{x}}(t))] - \frac{\eta}{2} E[\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\ &\quad + \left(\frac{\eta L^2 + 2\eta^2 L^3}{2m} + \frac{\eta^2 L^3}{\theta m^2}\right) E[\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\ &\quad + \left(\frac{\eta^2 L}{\theta m^2} + \frac{\eta^2 L}{m}\right) E[\|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2] \\ &\quad + \frac{\eta^2 L \sigma_g^2}{2m\theta} + \frac{c^2 L \sigma_c^2}{2m}. \end{aligned} \quad (34)$$

**Proof:** The proof is given in Appendix A.

#### B. Proof of Theorem 1

Using Assumption 3, we have

$$\begin{aligned} &\|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2 \\ &\leq 2\|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T) - \nabla f(\bar{\mathbf{x}}(t))\mathbf{1}_m^T\|_F^2 + 2\|\nabla f(\bar{\mathbf{x}}(t))\mathbf{1}_m^T\|_F^2 \\ &\leq 2m\sigma_G^2 + 2\|\nabla f(\bar{\mathbf{x}}(t))\mathbf{1}_m^T\|_F^2. \end{aligned} \quad (35)$$

Using this, we can rewrite the result in Lemma 3 as

$$\begin{aligned} E[\|\mathbf{X}(t+1) - \bar{\mathbf{x}}(t+1)\mathbf{1}_m^T\|_F^2] &\leq (1 - (1-\omega)c) E[\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\ &\quad + \left(\frac{4\eta^2 L^2}{(1-\omega)c} + \frac{2\eta^2 L^2}{\theta}\right) E[\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\ &\quad + \left(\frac{4\eta^2}{(1-\omega)c} + \frac{2\eta^2}{\theta}\right) 2m E[\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \end{aligned}$$

$$\begin{aligned}
& + \frac{4m\eta^2(\sigma_G^2 + \sigma_g^2)}{\theta} + \frac{8m\eta^2\sigma_G^2}{(1-\omega)c} + mc^2\sigma_c^2 \\
& \stackrel{(a)}{\leq} \left(1 - \frac{(1-\omega)c}{2}\right) E [\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\
& + \left(\frac{4\eta^2}{(1-\omega)c} + \frac{2\eta^2}{\theta}\right) 2mE [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\
& + \frac{4m\eta^2(\sigma_G^2 + \sigma_g^2)}{\theta} + \frac{8m\eta^2\sigma_G^2}{(1-\omega)c} + mc^2\sigma_c^2 \\
& \leq \left(1 - \frac{(1-\omega)c}{2}\right)^{t+1} [\|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2] \\
& + \left(\frac{4\eta^2}{(1-\omega)c} + \frac{2\eta^2}{\theta}\right) 2m \\
& \times \sum_{s=0}^t \left[\left(1 - \frac{(1-\omega)c}{2}\right)^{t-s} E [\|\nabla f(\bar{\mathbf{x}}(s))\|^2]\right] \\
& + \left(\frac{4m\eta^2(\sigma_G^2 + \sigma_g^2)}{\theta} + \frac{8m\eta^2\sigma_G^2}{(1-\omega)c} + mc^2\sigma_c^2\right) \\
& \times \sum_{s=0}^t \left(1 - \frac{(1-\omega)c}{2}\right)^s \tag{36}
\end{aligned}$$

where the condition

$$(1-\omega)c \geq \frac{8\eta^2L^2}{(1-\omega)c} + \frac{4\eta^2L^2}{\theta} \tag{37}$$

used in (a) holds for the values of  $\eta, c$ , and  $\theta$  given in the statement of Theorem 1 for all  $K \geq K_0$ . Using the fact that  $\sum_{s=0}^t \left(1 - \frac{(1-\omega)c}{2}\right)^s < \sum_{s=0}^{\infty} \left(1 - \frac{(1-\omega)c}{2}\right)^s = \frac{2}{(1-\omega)c}$ , and unrolling the recursion in (36), we get

$$\begin{aligned}
& \sum_{t=0}^{K-1} E [\|\mathbf{X}(t+1) - \bar{\mathbf{x}}(t+1)\mathbf{1}_m^T\|_F^2] \\
& \leq \sum_{t=0}^{K-1} \left[\left(1 - \frac{(1-\omega)c}{2}\right)^{t+1}\right] [\|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2] \\
& + \left(\frac{4\eta^2}{(1-\omega)c} + \frac{2\eta^2}{\theta}\right) 2m \\
& \times \sum_{t=0}^{K-1} \left[E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \left\{ \sum_{s=0}^{K-t-1} \left(1 - \frac{(1-\omega)c}{2}\right)^s \right\}\right] \\
& + \frac{8m\eta^2(\sigma_G^2 + \sigma_g^2)}{(1-\omega)c\theta} + \frac{16m\eta^2\sigma_G^2}{(1-\omega)^2c^2} + \frac{2mc\sigma_c^2}{(1-\omega)}. \tag{38}
\end{aligned}$$

Adding  $\|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2$  on both sides, we get

$$\begin{aligned}
& \frac{1}{K} \sum_{t=0}^K E [\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\
& \leq \frac{2}{(1-\omega)cK} \|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{8\eta^2}{(1-\omega)^2c^2} + \frac{4\eta^2}{(1-\omega)c\theta}\right) \frac{2m}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\
& + \frac{8m\eta^2(\sigma_G^2 + \sigma_g^2)}{(1-\omega)c\theta} + \frac{16m\eta^2\sigma_G^2}{(1-\omega)^2c^2} + \frac{2mc\sigma_c^2}{(1-\omega)}. \tag{39}
\end{aligned}$$

From the initialization condition, we have  $\|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2 = 0$ . Plugging in the values of  $\eta, c$ , and  $\theta$  as given in the statement of Theorem 1 in (39), (15) follows.

Again, we use (35) and the fact that  $\eta L < \theta < 1$  holds for large  $K$  to rewrite the result in Lemma 4 as follows:

$$\begin{aligned}
f^* \leq E [f(\bar{\mathbf{x}}(t+1))] & \leq E [f(\bar{\mathbf{x}}(t))] - \frac{\eta}{2} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\
& + \frac{3\eta L^2}{m} E [\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\
& + 2\left(\frac{\eta^2 L}{\theta m} + \eta^2 L\right) E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] + \frac{c^2 L \sigma_c^2}{2m} \\
& + 2\eta^2 L \sigma_G^2 + \frac{\eta^2 L (\sigma_g^2 + \sigma_G^2)}{m\theta}. \tag{40}
\end{aligned}$$

From the choice of the parameters  $\eta, c$ , and  $\theta$ , it is clear the following condition holds for all  $K \geq K_0$ :

$$\frac{8\eta L}{\theta m} + 8\eta L \leq 1. \tag{41}$$

Then, this allows us to rewrite (40) as

$$\begin{aligned}
& \frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \leq \frac{4[f(\bar{\mathbf{x}}(0)) - f^*]}{\eta K} \\
& + \frac{2c^2 L \sigma_c^2}{\eta m} + 8\eta L \sigma_G^2 + \frac{4\eta L (\sigma_g^2 + 2\sigma_G^2)}{m\theta} \\
& + \frac{12L^2}{m} \frac{1}{K} \sum_{t=0}^{K-1} E [\|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2] \\
& \leq \frac{4[f(\bar{\mathbf{x}}(0)) - f^*]}{\eta K} + \frac{2c^2 L \sigma_c^2}{\eta m} + 8\eta L \sigma_G^2 + \frac{4\eta L (\sigma_g^2 + 2\sigma_G^2)}{m\theta} \\
& + \left(\frac{96\eta^2 L^2}{(1-\omega)^2 c^2} + \frac{48L^2 \eta^2}{(1-\omega)c\theta}\right) \frac{2}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\
& + \frac{96\eta^2 L^2 (\sigma_g^2 + \sigma_G^2)}{(1-\omega)c\theta} + \frac{192\eta^2 L^2 \sigma_G^2}{(1-\omega)^2 c^2} + \frac{24L^2 c \sigma_c^2}{(1-\omega)} \\
& + \frac{12L^2}{(1-\omega)cK} \frac{1}{m} \|\mathbf{X}(0) - \bar{\mathbf{x}}(0)\mathbf{1}_m^T\|_F^2. \tag{42}
\end{aligned}$$

Again, the following inequality holds for the choice of the parameters  $\eta, c$ , and  $\theta$  for all  $K \geq K_0$ :

$$\frac{96\eta^2 L^2}{(1-\omega)^2 c^2} + \frac{48L^2 \eta^2}{(1-\omega)c\theta} \leq \frac{1}{4}. \tag{43}$$

Using the above, we can simplify (42) as follows:

$$\frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \leq \frac{8[f(\bar{\mathbf{x}}(0)) - f^*]}{\eta K}$$

$$\begin{aligned}
& + \frac{4c^2 L \sigma_c^2}{\eta m} + 16\eta L \sigma_G^2 + \frac{8\eta L (\sigma_g^2 + \sigma_G^2)}{m\theta} \\
& + \frac{192\eta^2 L^2 (\sigma_g^2 + \sigma_G^2)}{(1-\omega)c\theta} + \frac{384\eta^2 L^2 \sigma_G^2}{(1-\omega)^2 c^2} + \frac{48L^2 c \sigma_c^2}{(1-\omega)} \\
& + \frac{24L^2}{(1-\omega)cK} \frac{1}{m} \|\mathbf{X}(0) - \bar{\mathbf{x}}(0) \mathbf{1}_m^T\|_F^2. \quad (44)
\end{aligned}$$

Recalling that  $\|\mathbf{X}(0) - \bar{\mathbf{x}}(0) \mathbf{1}_m^T\|_F^2 = 0$  holds from the initialization condition and plugging in the values of  $\eta$ ,  $c$ , and  $\theta$  from the statement of Theorem 1, one obtains (14).

Finally, we need to prove that the actual number of stochastic gradient computations is concentrated around its expected value. To show this, we note that for any given  $i \in \{1, \dots, m\}$ , the random variables  $\{\chi_i(s)\}$ ,  $s = 0, \dots, K-1$ , are independent Bernoulli random variables, bounded between 0 and 1. Then, applying Hoeffding's inequality for some  $\phi > 0$ , we get

$$\Pr \left\{ \sum_{s=0}^{K-1} \chi_i(s) \geq \sum_{s=0}^{K-1} E[\chi_i(s)] + \phi K \right\} \leq \exp(-2K\phi^2). \quad (45)$$

Substituting the value of  $\sum_{s=0}^{K-1} E[\chi_i(s)]$  from (19) and choosing  $\phi = \frac{m^\gamma}{K^{1/3}}$ , we obtain (20). This concludes the proof ■.

**Remark 1:** In this work, we have used fixed step-sizes, which are dependent on  $K$ . This is because the required accuracy  $\epsilon$  is usually known in advance, which in turn allows us to calculate the number of iterations  $K$  to be run, as we have done in (23) and (24) following the result in Corollary 1. This is indeed a commonly adopted practice in both theoretical analysis and practical implementations (see, for example, the works [21], [28], [33], and [32]), and also allows for relatively easier analysis. However, there might be situations where  $\epsilon$  is not known in advance, and in such cases, one can follow the proof technique presented in the current work, along with some minor modifications, to show that the use of a time-dependent step-size scheme given by  $\eta_k = m^{1/3}/(1+k)^{2/3}$ ,  $c_k = m^{1/2}/(1+k)^{1/2}$ , and  $\theta_k = 1/(m^{1/3}(1+k)^{1/3})$  at iteration  $k$ , in Algorithm 1, yields a  $O(\log(k)/(m^{1/3}k^{1/3}))$  convergence rate, which ultimately translates to  $O((\log(1/\epsilon))^3/(m\epsilon^2))$  computational cost and  $O((\log(1/\epsilon))^3/(m\epsilon^3))$  communication cost. Thus, there is an extra  $(\log(1/\epsilon))^3$  factor when using the time-dependent step-size scheme described above as compared to the result obtained in (23) and (24) using fixed step-sizes, which is not too onerous.

## V. FURTHER INSIGHTS AND ROBUSTNESS

Note that setting  $\theta$  to 1 in Algorithm 1 reduces it to the algorithms proposed in [17], [19], [20], [21], and [22]. One of the key contributions of this work is to show that one can obtain similar number of iterations as in previous works with a significantly smaller value of  $\theta$  (which, in turn, leads to reduction in computational complexity). To see this in more detail, we consider the parameters  $\eta$ ,  $c$ , and  $\theta$  as given by (13). Leaving the exponent of  $K$  in  $\theta$  as a free parameter  $\tau$ , we have

$$\eta = \frac{m^\alpha}{K^{2/3}}, c = \frac{m^\beta}{K^{1/2}}, \theta = \frac{m^\gamma}{K^\tau}. \quad (46)$$

Here, we clarify that the dependencies of the parameters  $\eta$  and  $c$  in (13) (and by extension in (46)) on  $K$  are chosen so as to obtain the best possible dependence on  $K$  in the RHS of (44), which is then reflected in the result obtained in Theorem 1.<sup>5</sup> With the values of  $\eta$  and  $c$  decided, we try to determine the range of values that  $\tau$  can take, for which, we take into account the relationships between the step-sizes  $\eta$ ,  $c$ , and  $\theta$  that need to be satisfied. These are given by (37), (41), and (43). Plugging in the values of  $\eta$ ,  $c$ , and  $\theta$  from (46) in (37), (41), and (43), it is straightforward to see that  $\tau$  needs to be in the range  $0 \leq \tau < 2/3$  for the inequalities (37), (41), and (43) to hold for sufficiently large and finite  $K$ . We then plug in the values from (46) in (44) and recall that  $\|\mathbf{X}(0) - \bar{\mathbf{x}}(0) \mathbf{1}_m^T\|_F^2 = 0$  holds from the initialization condition to obtain the following bound for  $0 \leq \tau < 2/3$ :

$$\begin{aligned}
& \frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \leq O\left(\frac{m^{-\alpha}}{K^{1/3}}\right) [f(\bar{\mathbf{x}}(0)) - f^*] \\
& + O\left(\frac{m^{(2\beta-1-\alpha)}}{K^{1/3}} + \frac{m^\beta}{K^{1/2}}\right) \sigma_c^2 \\
& + O\left(\frac{m^{(\alpha-1-\gamma)}}{K^{2/3-\tau}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{5/6-\tau}}\right) \sigma_g^2 \\
& + O\left(\frac{m^{(\alpha-1-\gamma)}}{K^{2/3-\tau}} + \frac{m^{2(\alpha-\beta)}}{K^{1/3}} + \frac{m^{(2\alpha-\beta-\gamma)}}{K^{5/6-\tau}} + \frac{m^\alpha}{K^{2/3}}\right) \sigma_G^2. \quad (47)
\end{aligned}$$

Restricting our focus solely on the dependence on  $K$ , the above expression simplifies to

$$\begin{aligned}
& \frac{1}{K} \sum_{t=0}^{K-1} E [\|\nabla f(\bar{\mathbf{x}}(t))\|^2] \\
& \leq \begin{cases} O\left(\frac{1}{K^{1/3}}\right), & \text{if } 0 \leq \tau \leq 1/3 \\ O\left(\frac{1}{K^{2/3-\tau}}\right), & \text{if } 1/3 < \tau < 2/3. \end{cases} \quad (48)
\end{aligned}$$

Remarkably, this suggests that as long as  $\tau \leq 1/3$ , the dependence of the convergence rate on the number of iterations  $K$  remains unaffected and scales as  $O(1/K^{1/3})$ , thus matching the iteration and communication complexities of previous works while achieving order of magnitude reduction in the computational complexity if  $\tau > 0$  (specifically, when expressed in terms of the required accuracy  $\epsilon$ , the computational and communication costs, for  $0 < \tau \leq 1/3$ , scale as  $O(1/\epsilon^{3(1-\tau)})$  and  $O(1/\epsilon^3)$ , respectively, with the  $O(1/\epsilon^{3(1-\tau)})$  computational cost improving over the  $O(1/\epsilon^3)$  computational cost obtained in [21] and the  $O(1/\epsilon^3)$  communication cost matching the communication cost obtained in [21]). Intuitively, we believe that this is due to the fact that the convergence rate of the algorithm is slowed down more by the noise introduced in the information exchange process than by the noise introduced due to stochastic gradient computation. This is also visible from the fact that if we set  $\tau = 0$  in (47), the convergence rate decay

<sup>5</sup>This can be verified by initially setting the exponents of  $K$  in  $\eta$  and  $c$  as free parameters, say,  $\tau_1$  and  $\tau_2$  and then performing algebraic calculations to optimize for the dependence on  $K$  in the RHS of (44)

term associated with the variance of the stochastic gradient  $\sigma_g^2$  scales as  $1/K^{2/3}$ , which is a much smaller decay term as compared to the  $1/K^{1/3}$  decay term associated with the communication noise variance  $\sigma_c^2$ , the network averaged gradient disagreement  $\sigma_c^2$ , and the initialization error term  $f(\bar{\mathbf{x}}(0)) - f^*$ . As a result, one can get away with even worse gradient estimates (gradient estimates having higher variance than  $\sigma_g^2$ ) without any effect on the dependence of the convergence rate on the number of iterations  $K$  up to a certain extent. This ‘‘certain extent’’ is quantified in (48) through the choice of  $\tau$  in the parameter  $\theta$ , which governs how frequently the stochastic gradient computation step takes place and is given by the range  $0 \leq \tau \leq 1/3$ . However, once  $\tau > 1/3$  (i.e. the gradient computation probability starts decreasing further), the convergence rate starts deteriorating and no longer retains the  $O(1/K^{1/3})$  dependence as evident from (48). In (13), in Theorem 1, we have chosen the highest possible value of  $\tau$  to allow for the lowest value of the probability of stochastic gradient computation, while at the same time retaining the best possible overall convergence rate of  $O(1/K^{1/3})$ . As it turns out from (48), this condition amounts to choosing  $\tau = 1/3$ .

The above discussion also suggests that the proposed method is robust to environments where the availability of computational resources is unreliable. This can occur in situations where there are frequent power outages or straggler nodes in the network, which makes it difficult to compute the stochastic gradient reliably across all iterations. This problem is particularly relevant in today’s day and age where models are becoming increasingly complex and data-hungry, as discussed in Section I. Such computationally unreliable environments with random failures in stochastic gradient computation can be modeled using the framework developed in Algorithm 1. For example, one could model power outages to cause successful stochastic gradient computation to occur in a given iteration with probability  $\theta < 1$ . Until now, it was up to the algorithm developer to choose the value of  $\theta$ , where, as here, the value of  $\theta$  will be set by the environment. Remarkably, our result shows that if zero-mean bounded variance communication noise is present, unreliable computation does not have any effect on the overall convergence of the algorithm as long as the value of  $\tau$  lies in the range  $0 \leq \tau \leq 1/3$ . To give an example with realistic numbers, suppose a particular application requires the accuracy to be  $\epsilon = 0.01$ . Then, the number of iterations (i.e.,  $K$ ) to be run would scale as  $1/\epsilon^3 = 1\,000\,000$ . Our result in (48) then suggests that the number of iterations would remain unaffected as long as the probability of successful stochastic gradient computation is above a certain small threshold which scales as  $1/K^{1/3}$ , which in the current example evaluates to 0.01. This means that power outages could randomly disrupt the stochastic gradient computation step roughly 99 out of 100 times on average and it still would not affect the overall convergence rate, which from (48) would still enjoy a  $1/K^{1/3}$  dependence.

## VI. NUMERICAL EXPERIMENTS

We consider a network consisting of  $m = 20$  nodes, which we generate randomly, such that for each pair of nodes  $i$  and  $j$ , the probability of an edge connecting them is 0.75. The network matrix  $\mathbf{W} \in R^{20 \times 20}$  is set to  $\mathbf{W} = \mathbf{I}_{20} - (3/4\lambda_{\max}(\mathbf{L}))\mathbf{L}$ ,

where  $\mathbf{L} \in R^{20 \times 20}$  is the Laplacian of the graph and  $\lambda_{\max}(\mathbf{L})$  is its maximum eigenvalue.

We consider the *Phishing* dataset, which is publicly available at *openML.org*. It consists of a total of 11 056 data points and 30 features. We select the first 10 000 data points and all 30 features to construct our loss function, for which we assign 500 nonoverlapping data points to each of the 20 nodes. For our study, we consider the following local nonconvex loss functions at node  $i$ , the first of which is a robust regression model [36], and the second is a nonconvex logistic regression model [37]:

$$f_i(\mathbf{x}) = \frac{1}{500} \sum_{j=1}^{500} \frac{(\langle \mathbf{a}_{ij}, \mathbf{x} \rangle - b_{ij})^2}{1 + (\langle \mathbf{a}_{ij}, \mathbf{x} \rangle - b_{ij})^2} \quad (49)$$

$$f_i(\mathbf{x}) = \frac{1}{500} \sum_{j=1}^{500} \log \left( 1 + e^{-\langle \mathbf{a}_{ij}, \mathbf{x} \rangle b_{ij}} \right) + \lambda \sum_{k=1}^d \frac{x_k^2}{1 + x_k^2}. \quad (50)$$

Here,  $d = 30$ ,  $\mathbf{a}_{ij} \in R^{30}$  are the feature vectors at node  $i$ , and  $b_{ij} \in R$  are the corresponding labels. In (50),  $x_k$  denotes the entry at the  $k$ th position of the column vector  $\mathbf{x} \in R^{30}$  and  $\lambda = 0.0001$  denotes the regularization parameter. Among the first 10 000 points in the dataset, there are 4437 data points with label  $-1$  (i.e., their  $b_{ij}$  value is  $-1$ ), one data point with label 0, and 5562 data points with label 1. Nodes 1–8 are assigned data points with label  $-1$ , node 9 is assigned data point having a mix of labels  $-1$  and 1 and the single data point with label 0, and nodes 10–20 are assigned data points with label 1. This ensures that there is sufficient heterogeneity in the data available at different nodes [34], which is often a critical issue encountered in decentralized optimization problems.

In the current work, we have incorporated the skipping technique in the widely used *Noisy Consensus + (Stochastic) Gradient Descent* framework to create Algorithm 1. We shall abbreviate the *Noisy Consensus + (Stochastic) Gradient Descent* method as NCSGD for brevity and use NCSGD+skip to abbreviate the *Noisy Consensus + (Stochastic) Gradient Descent* method with the skipping technique incorporated in it. In fact, NCSGD can be obtained as a special case of NCSGD+skip by setting the gradient computation probability as  $\theta = 1$  in NCSGD+skip. In our experiments, for both NCSGD and NCSGD+skip, we consider stochastic gradients (specifically, we use minibatch size = 1 for stochastic gradient computation by each node), and use zero-mean Gaussian noise with variance 0.1 to model communication noise [specifically, the entries of  $\mathbf{N}(t)$  in (31) are drawn from a Gaussian distribution with mean 0 and variance 0.1].

The results corresponding to (49) and (50) are shown in Figs. 1 and 2, respectively, with each plot averaged over 100 runs, and the step-size parameters corresponding to each plot appropriately tuned over a grid. In particular, for both (49) and (50), we consider NCSGD first and tune the parameters  $\eta$  and  $c$  over the grids  $\{0.0033, 0.01, 0.033, 0.1, 0.33\}$  for (49) and  $\{0.01, 0.033, 0.1, 0.33, 1\}$  for (50), meaning that we test out  $5 \times 5 = 25$  combinations of parameters for each of considered loss functions. Following this, for each of the considered loss functions (49) and (50), we consider NCSGD+skip, for which we use the previously tuned values of  $\eta$  and  $c$  from the vanilla NCSGD method, and tune  $\theta$  over the

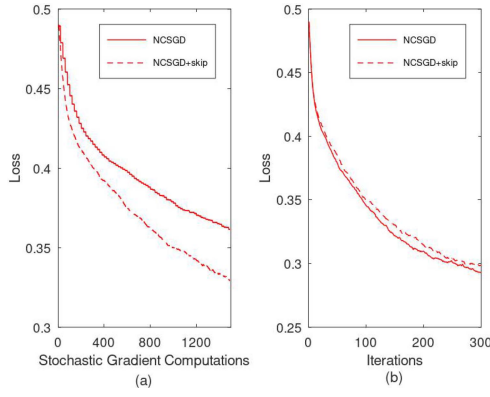


Fig. 1. Robust regression problem (49).

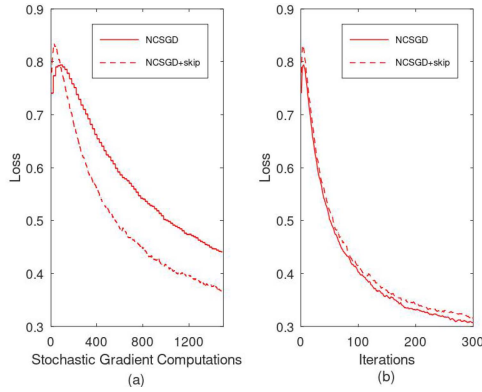


Fig. 2. Nonconvex logistic regression problem (50).

grid  $\{1/2, 1/3, 1/4, 1/5\}$  conditioned on these fixed choices for  $\eta$  and  $c$  (note here that we do not consider all  $5 \times 5 \times 4$  combinations for the triplet  $(\eta, c, \theta)$ ).

In both Figs. 1(a) and 2(a), we study the decrease in the loss, i.e., the average of the global objective function evaluated at the different nodes' estimates  $(1/m) \sum_{i=1}^m f(x_i(t))$  as a function of the total number of stochastic gradient computations up to time  $t$  (i.e.,  $\sum_{s=0}^t \sum_{i=1}^m \chi_i(s)$ ). Here, it is clear that the skipping technique is able to reduce computational cost since NCSGD+skip requires fewer stochastic gradient computations to reach a certain value of the loss, as compared to NCSGD. Similarly, in Figs. 1(b) and 2(b), we study the decrease in the quantity  $(1/m) \sum_{i=1}^m f(x_i(t))$  as a function of the iteration number  $t$ . Here again, we see that there is very little degradation in the overall convergence of the algorithm when the skipping technique is introduced. This confirms reduction in computational cost while retaining overall algorithm convergence as claimed earlier in this article.

## VII. CONCLUSION AND FUTURE WORK

In this article, we incorporated a skipping technique in the *Noisy Consensus + SGD* framework, and showed that it helps us achieve a significant reduction in computational cost without hurting overall algorithm convergence. In future, we wish to extend this skipping technique to other famous algorithms, e.g., *Robust Gradient Tracking*-based algorithms [16], [18] for decentralized optimization in the presence of zero-mean and bounded

variance communication noise, and conduct theoretical analysis of the same to derive convergence rate, computational cost, and communication costs.

## APPENDIX

### A. Proofs of Auxiliary Results

Let  $\mathcal{F}_t$  denote the history of the algorithm up to time  $t$ .

*Proof of Lemma 2:* The proof follows trivially by postmultiplying both the left-hand side and the right-hand side of (31) by  $\frac{1}{m} \mathbf{1}_m$  and noting that  $\mathbf{W}_c$  is doubly stochastic, meaning  $\mathbf{W}_c^T \mathbf{1}_m = \mathbf{1}_m$ . ■

*Proof of Lemma 3:* From (31) and (32), we have

$$\begin{aligned} & \mathbf{X}(t+1) - \bar{\mathbf{x}}(t+1) \mathbf{1}_m^T \\ &= (\mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T) \mathbf{W}_c^T - \eta (\mathbf{H}(t) - \bar{\mathbf{h}}(t) \mathbf{1}_m^T) \\ & \quad + c (\mathbf{N}(t) - \bar{\mathbf{n}}(t) \mathbf{1}_m^T) \\ &= (\mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T) \mathbf{W}_c^T - \eta (\nabla F(\mathbf{X}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \mathbf{1}_m^T) \\ & \quad - \eta (\mathbf{H}(t) - \bar{\mathbf{h}}(t) \mathbf{1}_m^T) + \eta (\nabla F(\mathbf{X}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \mathbf{1}_m^T) \\ & \quad + c (\mathbf{N}(t) - \bar{\mathbf{n}}(t) \mathbf{1}_m^T). \end{aligned} \quad (51)$$

Denoting

$$\begin{aligned} \mathcal{A}_1(t) &= (\mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T) \mathbf{W}_c^T \\ & \quad - \eta (\nabla F(\mathbf{X}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \mathbf{1}_m^T) \end{aligned} \quad (52)$$

$$\begin{aligned} \mathcal{A}_2(t) &= -\eta [\mathbf{H}(t) - \bar{\mathbf{h}}(t) \mathbf{1}_m^T] \\ & \quad + \eta [\nabla F(\mathbf{X}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \mathbf{1}_m^T] \\ & \quad + c (\mathbf{N}(t) - \bar{\mathbf{n}}(t) \mathbf{1}_m^T) \end{aligned} \quad (53)$$

we get

$$\begin{aligned} & E \left[ \|\mathbf{X}(t+1) - \bar{\mathbf{x}}(t+1) \mathbf{1}_m^T\|_F^2 | \mathcal{F}_t \right] \\ &= \|\mathcal{A}_1(t)\|_F^2 + E \left[ \|\mathcal{A}_2(t)\|_F^2 | \mathcal{F}_t \right] + 2E \left[ \langle \mathcal{A}_1(t), \mathcal{A}_2(t) \rangle | \mathcal{F}_t \right] \\ &= \|\mathcal{A}_1(t)\|_F^2 + E \left[ \|\mathcal{A}_2(t)\|_F^2 | \mathcal{F}_t \right]. \end{aligned} \quad (54)$$

We bound  $\|\mathcal{A}_1(t)\|_F^2$  as follows:

$$\begin{aligned} \|\mathcal{A}_1(t)\|_F^2 &\stackrel{(a)}{\leq} \left(1 + \phi_t\right) \left\| (\mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T) \mathbf{W}_c^T \right\|_F^2 \\ & \quad + \left(1 + \frac{1}{\phi_t}\right) \eta^2 \left\| \nabla F(\mathbf{X}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \mathbf{1}_m^T \right\|_F^2 \\ &\stackrel{(b)}{\leq} (1 - (1 - \omega)c) \left\| \mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T \right\|_F^2 \\ & \quad + \frac{2\eta^2}{(1 - \omega)c} \left\| \nabla F(\mathbf{X}(t)) \right\|_F^2 \\ &\stackrel{(b)}{\leq} (1 - (1 - \omega)c) \left\| \mathbf{X}(t) - \bar{\mathbf{x}}(t) \mathbf{1}_m^T \right\|_F^2 \\ & \quad + \frac{4\eta^2}{(1 - \omega)c} \left\| \nabla F(\mathbf{X}(t)) - \nabla F(\bar{\mathbf{x}}(t) \mathbf{1}_m^T) \right\|_F^2 \end{aligned}$$

$$\begin{aligned}
 & + \frac{4\eta^2}{(1-\omega)c} \|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2 \\
 & \leq (1 - (1-\omega)c) \|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2 \\
 & + \frac{4\eta^2 L^2}{(1-\omega)c} \|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2 \\
 & + \frac{4\eta^2}{(1-\omega)c} \|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2 \quad (55)
 \end{aligned}$$

where (a) follows from Young's inequality for some  $\phi_t \geq 0$  and (b) follows by substituting  $\phi_t = (1-\omega)c$ . We bound  $E[\|\mathcal{A}_2(t)\|^2|\mathcal{F}_t]$  as follows:

$$\begin{aligned}
 & E[\|\mathcal{A}_2(t)\|_F^2|\mathcal{F}_t] \\
 & \leq c^2 E[\|\mathbf{N}(t) - \bar{\mathbf{n}}(t)\mathbf{1}_m^T\|_F^2|\mathcal{F}_t] \\
 & + \eta^2 E[\|(\mathbf{H}(t) - \nabla F(\mathbf{X}(t))) \\
 & - (\bar{\mathbf{h}}(t) - \bar{\nabla} F(\mathbf{X}(t)))\mathbf{1}_m^T\|_F^2|\mathcal{F}_t] \\
 & \leq c^2 E[\|\mathbf{N}(t)\|_F^2|\mathcal{F}_t] + \eta^2 E[\|\mathbf{H}(t) - \nabla F(\mathbf{X}(t))\|_F^2|\mathcal{F}_t] \\
 & \stackrel{(a)}{\leq} mc^2\sigma_c^2 + \frac{\eta^2}{\theta} (m\sigma_g^2 + \|\nabla F(\mathbf{X}(t))\|_F^2) \\
 & \leq mc^2\sigma_c^2 + \frac{m\eta^2\sigma_g^2}{\theta} + \frac{2\eta^2 L^2}{\theta} \|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2 \\
 & + \frac{2\eta^2}{\theta} \|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2 \quad (56)
 \end{aligned}$$

where in (a), we have used the result from Lemma 1. Combining (54)–(56) and taking expectations with respect to  $\mathcal{F}_t$ , we obtain the desired result. ■

*Proof of Lemma 4:* Using  $L$ -smoothness of the global objective function

$$\begin{aligned}
 & f(\bar{\mathbf{x}}(t+1)) \\
 & \leq f(\bar{\mathbf{x}}(t)) + \langle \nabla f(\bar{\mathbf{x}}(t)), \bar{\mathbf{x}}(t+1) - \bar{\mathbf{x}}(t) \rangle \\
 & + \frac{L}{2} \|\bar{\mathbf{x}}(t+1) - \bar{\mathbf{x}}(t)\|^2 \\
 & = f(\bar{\mathbf{x}}(t)) - \langle \nabla f(\bar{\mathbf{x}}(t)), \eta\bar{\mathbf{h}}(t) - c\bar{\mathbf{n}}(t) \rangle \\
 & + \frac{L}{2} \|\eta\bar{\mathbf{h}}(t) - c\bar{\mathbf{n}}(t)\|^2. \quad (57)
 \end{aligned}$$

Taking expectations conditioned on  $\mathcal{F}_t$ , we get

$$\begin{aligned}
 & E[f(\bar{\mathbf{x}}(t+1))|\mathcal{F}_t] \\
 & \leq f(\bar{\mathbf{x}}(t)) - \eta \langle \nabla f(\bar{\mathbf{x}}(t)), \bar{\nabla} F(\mathbf{X}(t)) \rangle \\
 & + \frac{\eta^2 L}{2} E[\|\bar{\mathbf{h}}(t)\|^2|\mathcal{F}_t] + \frac{c^2 L \sigma_c^2}{2m}. \quad (58)
 \end{aligned}$$

We observe that

$$\bar{\mathbf{h}}(t) = \frac{1}{m} \sum_{i=1}^m \underbrace{[\mathbf{h}_i(t) - \nabla f_i(\mathbf{x}_i(t))]}_{\delta_i(t)} + \frac{1}{m} \sum_{i=1}^m \nabla f_i(\mathbf{x}_i(t)).$$

We note that the set of random variables  $\{\delta_i(t)\}$ ,  $i \in V$ , are zero-mean independent random variables conditioned on  $\mathcal{F}_t$ . Consequently, we have

$$\begin{aligned}
 & E[\|\bar{\mathbf{h}}(t)\|^2|\mathcal{F}_t] \\
 & \leq \frac{1}{m^2} \sum_{i=1}^m E[\|\delta_i(t)\|^2|\mathcal{F}_t] + \frac{1}{m^2} \left\| \sum_{i=1}^m \nabla f_i(\mathbf{x}_i(t)) \right\|^2 \\
 & \stackrel{(a)}{\leq} \frac{1}{\theta m^2} \sum_{i=1}^m [\sigma_g^2 + \|\nabla f_i(\mathbf{x}_i(t))\|^2] \\
 & + \frac{1}{m} \sum_{i=1}^m \|\nabla f_i(\mathbf{x}_i(t))\|^2 \\
 & \leq \frac{\sigma_g^2}{\theta m} + \left( \frac{1}{\theta m^2} + \frac{1}{m} \right) \|\nabla F(\mathbf{X}(t))\|_F^2 \\
 & \leq \frac{\sigma_g^2}{\theta m} + 2L^2 \left( \frac{1}{\theta m^2} + \frac{1}{m} \right) \|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2 \\
 & + 2 \left( \frac{1}{\theta m^2} + \frac{1}{m} \right) \|\nabla F(\bar{\mathbf{x}}(t)\mathbf{1}_m^T)\|_F^2 \quad (59)
 \end{aligned}$$

where (a) follows from the result in (6b). Next, we observe that

$$\begin{aligned}
 & -\eta \langle \nabla f(\bar{\mathbf{x}}(t)), \bar{\nabla} F(\mathbf{X}(t)) \rangle \\
 & = -\eta \langle \nabla f(\bar{\mathbf{x}}(t)), \nabla f(\bar{\mathbf{x}}(t)) - \bar{\nabla} F(\mathbf{X}(t)) \rangle \\
 & - \eta \|\nabla f(\bar{\mathbf{x}}(t))\|^2 \\
 & \leq \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}(t)) - \bar{\nabla} F(\mathbf{X}(t))\|^2 - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}(t))\|^2 \\
 & \leq \frac{\eta}{2m} \sum_{i=1}^m \|\nabla f_i(\bar{\mathbf{x}}(t)) - \nabla f_i(\mathbf{x}_i(t))\|^2 - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}(t))\|^2 \\
 & \leq \frac{\eta L^2}{2m} \|\mathbf{X}(t) - \bar{\mathbf{x}}(t)\mathbf{1}_m^T\|_F^2 - \frac{\eta}{2} \|\nabla f(\bar{\mathbf{x}}(t))\|^2. \quad (60)
 \end{aligned}$$

Combining (58)–(60) and taking expectations w.r.t.  $\mathcal{F}_t$ , we obtain the desired result. ■

## REFERENCES

- [1] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [2] P. Forero, A. Cano, and G. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, no. 55, pp. 1663–1707, 2010.
- [3] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for Big Data: Scalable, randomized, and parallel algorithms for Big Data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32–43, Sep. 2014, doi: 10.1109/MSP.2014.2329397.
- [4] M. Stolpe, "The Internet of Things: Opportunities and challenges for distributed data analysis," *ACM SIGKDD Explorations Newslett.*, vol. 18, no. 1, pp. 15–34, Jun. 2016.
- [5] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, May 2013.
- [6] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
- [7] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

- [8] B. Swenson, R. Murray, H. V. Poor, and S. Kar, "Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 14751–14812, 2022.
- [9] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. 51st IEEE Conf. Decis. Control*, Maui, HI, USA, 2012, pp. 5445–5450.
- [10] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
- [11] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network Newton distributed optimization methods," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, Jan. 2017.
- [12] R. Tutunov, H. Bou-Ammar, and A. Jadbabaie, "Distributed newton method for large-scale consensus optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 3983–3994, Oct. 2019.
- [13] Y. Li, P. G. Voulgaris, D. M. Stipanović, and N. M. Freris, "Communication efficient curvature aided primal-dual algorithms for decentralized optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 11, pp. 6573–6588, Nov. 2023.
- [14] S. Liang, L. Y. Wang, and G. Yin, "Distributed smooth convex optimization with coupled constraints," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 347–353, Jan. 2020.
- [15] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.
- [16] S. Pu, "A robust gradient tracking method for distributed optimization over directed networks," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 2335–2341.
- [17] H. Reiszadeh, B. Touri, and S. Mohajer, "Distributed optimization over time-varying graphs with imperfect sharing of information," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4420–4427, Jul. 2023.
- [18] Y. Wang and T. Başar, "Gradient-tracking-based distributed optimization with guaranteed optimality under noisy information sharing," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 4796–4811, Aug. 2023.
- [19] T. T. Doan, S. T. Maguluri, and J. Romberg, "Convergence rates of distributed gradient methods under random quantization: A stochastic approximation approach," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4469–4484, Oct. 2021.
- [20] A. Reiszadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934–4947, Oct. 2019.
- [21] A. Reiszadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 33, pp. 8388–8399.
- [22] M. M. Vasconcelos, T. T. Doan, and U. Mitra, "Improved convergence rate for a distributed two-time-scale gradient method under random quantization," in *Proc. 60th IEEE Conf. Decis. Control*, Austin, TX, USA, 2021, pp. 3117–3122.
- [23] J. He, L. Cai, and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, 2020.
- [24] Y. Wang and T. Başar, "Quantization enabled privacy protection in decentralized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4038–4052, Jul. 2023.
- [25] Y. Wang and A. Nedić, "Tailoring gradient methods for differentially private distributed optimization," *IEEE Trans. Autom. Control*, vol. 69, no. 2, pp. 872–887, Feb. 2024.
- [26] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in over-parameterized neural networks, going beyond two layers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 33, pp. 6155–6166.
- [27] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Springer Knowl. Inf. Syst.*, vol. 63, pp. 2585–2619, 2021.
- [28] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.
- [29] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth, "Lower bounds for non-convex stochastic optimization," *Math. Program.*, vol. 199, pp. 165–21, 2023.
- [30] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3–4, pp. 231–357, 2015.
- [31] Y. E. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Norwell, MA, USA: Kluwer, 2004.
- [32] X. Lian, C. Zhang, H. Zhang, C. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5330–5340.
- [33] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *J. Mach. Learn. Res.*, vol. 13, pp. 165–202, 2012.
- [34] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D2: Decentralized training over decentralized data," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, vol. 80, pp. 4848–4856.
- [35] R. Mohammad and L. McCluskey, "Phishing websites [Dataset]," UCI Machine Learning Repository, 2012, doi: [10.24432/C51W2X](https://doi.org/10.24432/C51W2X).
- [36] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, no. 3, pp. 147–185, 1974.
- [37] A. Antoniadis, I. Gijbels, and M. Nikolova, "Penalized likelihood regression for non-quadratic penalties," *Ann. Inst. Stat. Math.*, vol. 63, no. 3, pp. 585–615, 2011.



**Soham Mukherjee** received the B.Tech. and M.Tech. degrees in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2021.

He is currently a Quantitative Analyst in the financial services industry. His current research interests include optimization and learning in large-scale settings.



**Mrityunjoy Chakraborty** (Senior Member, IEEE) received the B.Eng. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 1983, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1985, and the Doctor of Philosophy degree in electrical engineering from the Indian Institute of Technology Delhi, New Delhi, India, in 1994.

In 1994, he joined the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India, as a Lecturer, where he is currently a Senior Professor. He was the Chair of the department during 2021–2023 and the prestigious Prithviraj Banerjee and Swati Banerjee Chair Professor during 2022–2024. His research interests include digital and adaptive signal processing, linear algebra, optimization, compressive sensing, and graph signal processing.

Dr. Chakraborty has been an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS during 2004–2007 and 2010–2012, and of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS (TCAS II) during 2008–2009 and 2022–2023. During 2017–2020, he was a Senior Editorial Board (SEB) Member of the *IEEE Signal Processing Magazine*, and during 2016–2017, he was an SEB Member of IEEE JOURNAL ON EMERGING TECHNIQUES IN CIRCUITS AND SYSTEMS. During 2016–2018, he was the Chair of the DSP Technical Committee (TC) of the IEEE Circuits and Systems Society. He has also been the Guest Editor of the *EURASIP Journal on Advances in Signal Processing* and Special Issues of the TCAS II, the DSP Track Co-Chair of ISCAS 2015–2023, the TPC Co-Chair of IEEE SIPS-2018, the Special Session Co-Chair of DSP-18, and the Gabor Track Chair of DSP-15. He is a Co-Founder of the Asia Pacific Signal and Information Processing Association (APSIPA), was a Member of the APSIPA BOG during 2013–2016, and was also the Chair of the APSIPA TC on Signal and Information Processing Theory and Methods. In 2012 and 2020, he was the General Chair of the National Conference on Communications. He is a Fellow of the National Academy of Sciences, India, and the Indian National Academy of Engineering (INAE). He was the recipient of the prestigious Chair Professorship of the INAE. During 2012–2013, he was selected as a Distinguished Lecturer of the APSIPA.