Toward $O(\log(1/\epsilon)/\epsilon)$ Computational Complexity for PL Functions in Decentralized Stochastic Optimization With Communication Noise

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

***Abstract*—A decentralized stochastic optimization problem is considered, in which a network of nodes collaborate over noisy communication links to minimize a global objective function. Recently, the authors in Mukherjee and Chakraborty (2025) incorporated a skipping technique in the *Noisy Consensus + Stochastic Gradient Descent (SGD)* framework to address this problem, and showed that their proposed skipping technique helps improve the $O(1/\epsilon^3)$ computational complexity obtained in previous works to $O(1/\epsilon^2)$ under a general smoothness assumption. In this letter, we consider the algorithm proposed in Mukherjee and Chakraborty (2025) to show how the computational complexity can be further improved to $O(\log(1/\epsilon)/\epsilon)$ when the Polyak-Lojasiewicz (PL) condition is satisfied in addition to the smoothness assumption. The obtained $O(\log(1/\epsilon)/\epsilon)$ rate in the current work is also an Improvement over the $O(1/\epsilon^2)$ rate obtained in previous works under the strong-convexity assumption (which is known to be stricter than the PL condition), and matches the $\Omega(1/\epsilon)$ lower bound for the number of stochastic gradient computations for the considered problem class up to an extra $\log(1/\epsilon)$ factor. Last but not least, the $O(\log(1/\epsilon)/\epsilon)$ computational cost is achieved while retaining the $O(\log^2(1/\epsilon)/\epsilon^2)$ rate for the number of iterations and communication rounds, which is at par with the results obtained in previous works which consider strong-convexity, up to logarithmic factors. A numerical experiment is conducted corroborate theoretical results.**

***Index Terms*—Stochastic optimization, decentralized optimization, communication noise, PL condition.**

I. Introduction

WE CONSIDER a decentralized optimization problem, where a group of m nodes, connected over a graph, seek to minimize a global objective function $f(x) = (1/m)\sum_{i=1}^{m} f_i(x)$, with f_i being known exclusively to node i in the network, by performing local stochastic gradient computations and sharing relevant information with their neighbours in the graph. The communication between nodes is assumed to be corrupted by zero-mean and bounded variance communication noise, which often arises in practical problems in various contexts like channel noise ([1], [2], [3], [4], [5]), noise added for privacy preservation ([6], [7], [8]) quantization noise due to the use of certain types of unbiased quantizers ([9], [10], [11]), to name a few.

Several recent works such as [3], [4], [5], [6], [7], [8], [9], [10], [11] have tried to address this problem by building upon the *Noisy Consensus + (Stochastic) Gradient Descent (SGD)* framework, owing to its simplicity, ease of implementation and widespread use. However, a common observation in all of these works is that the computational complexity incurred by the proposed algorithms is higher than the computational complexity that can be achieved by the centralized (Stochastic) Gradient Descent algorithm (i.e., the $m = 1$ case where there is only a single node in the network, and thus no communication noise). This issue was identified and addressed in [1] for the general class of smooth (and potentially non-convex) functions, where the authors showed how the computational complexity can be improved to match the computational complexity of the centralized (Stochastic) Gradient Descent algorithm by randomly choosing to skip the stochastic gradient computation step across iterations, with an appropriately tuned value for the probability of skipping. This then naturally raises the question of whether this skipping technique can be employed to achieve similar reductions in computational complexity when more structural assumptions are satisfied.

For this letter, we consider a class of functions which, in addition to the smoothness assumption, satisfy the Polyak-Lojasiewicz (PL) condition. PL functions have drawn significant attention over the last few years, more so with recent research such as [12], [13] suggesting that the loss landscapes in many neural network training problems satisfy the PL condition or related variants. In this letter, we show that under the PL condition, one can obtain a $O(log(1/\epsilon)/\epsilon)$ rate for the number of stochastic gradient computations. Since the PL condition covers strong-convexity (see [14]), the above rate also applies to strongly-convex functions, where previous works operating in the noisy communication setting (such as [4], [9], [10], [11]) obtained a $O(1/\epsilon^2)$ rate. This means that the $O(log(1/\epsilon)/\epsilon)$ rate obtained in this letter is

Received 25 July 2025; revised 19 September 2025; accepted 9 October 2025. Date of publication 30 October 2025; date of current version 5 November 2025. This work was supported by the Anusandhan National Research Foundation (ANRF), Govt. of India. Recommended by Senior Editor L. Menini. *(Corresponding author: Mrityunjoy Chakraborty.)*

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

Digital Object Identifier 10.1109/LCSYS.2025.3627045

a significant improvement from $O(1/\epsilon^2)$ for strongly-convex functions, obtained by incorporating the skipping technique in the *Noisy Consensus + SGD* framework. Furthermore, as the PL condition is more general than strong-convexity, the current work expands the scope of the problem setting by relaxing the strong-convexity assumption and replacing it with the less restrictive PL condition. We also note that the obtained $O(log(1/\epsilon)/\epsilon)$ rate for the number of stochastic gradient computations in the current work is near-optimal, in the sense that it matches the $\Omega(1/\epsilon)$ lower-bound obtained in [15] up to an extra log factor. Additionally, this reduction in computational complexity is achieved without making any sacrifice in the overall number of iterations and the number of communication rounds, both of which scale as $O(log^2(1/\epsilon)/\epsilon^2)$, and are at par with corresponding bounds obtained in the aforementioned previous works [4], [9], [10] and [11], up to log factors.

Finally, our obtained $O(log(1/\epsilon)/\epsilon)$ and $O(log^2(1/\epsilon)/\epsilon^2)$ rates for the number of stochastic gradient computations and the number of iterations/communications rounds directly improve over the $O(1/\epsilon^2)$ and $O(1/\epsilon^3)$ rates obtained in [1], which considers only the smoothness assumption without the PL condition, thus showing how the structure conferred to the problem by the PL condition helps us to obtain faster rates.

Notation. We use $\mathbf{1}_p$ and $\mathbf{0}_p$ to represent the p-dimensional vector of all 1s and 0s respectively. $\boldsymbol{I}_p$ is used to represent the $p \times p$ identity matrix. $||\cdot||_F$ represents the Frobenius norm for matrices and $||\cdot||$ represents the 2-norm for vectors. We use *Bernoulli*(p) to represent a Bernoulli distribution which switches between 1 and 0 with probability p and $1-p$ respectively. For any two $p \times q$ matrices $\boldsymbol{A}, \boldsymbol{B}$, we use $\boldsymbol{A} \geq \boldsymbol{B}$ iff $A_{ij} \geq B_{ij}$ for all $i \in \{1, \ldots, p\}, j \in \{1, \ldots, q\}$. We use standard big-O (O) and big-Omega (Ω) notations while stating complexity results in terms of relevant parameters.

II. Problem Description and Algorithm Overview

We consider a network of m nodes connected over an undirected graph $G = (V, E)$, where $V = \{1, \ldots, m\}$ represents the set of nodes in the network and E represents the set of edges connecting the nodes. The nodes perform local computations and exchange relevant information to minimize the following global objective function

$$f(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\boldsymbol{x}), \tag{1}$$

such that $f_i : R^d \to R$ is accessible exclusively to node i in the network. Specifically, we consider a setting where given a point $\boldsymbol{x}_i(t) \in R^d$ at time t, node i can compute a stochastic gradient $\boldsymbol{g}_i(\boldsymbol{x}_i(t))$, which satisfies the following conditions.

$$E\big[\boldsymbol{g}_i(\boldsymbol{x}_i(t))|\boldsymbol{x}_i(t)\big] = \nabla f_i(\boldsymbol{x}_i(t)), \tag{2a}$$

$$E\Big[||\boldsymbol{g}_i(\boldsymbol{x}_i(t)) - \nabla f_i(\boldsymbol{x}_i(t))||^2|\boldsymbol{x}_i(t)\Big] \leq \sigma_g^2. \tag{2b}$$

Eqs. (2a)-(2b) are standard unbiased-ness and bounded variance assumptions which are common in decentralized stochastic optimization literature.

We make the following assumptions about the local component functions f_i and the global objective function f.

Assumption 1: The local objective functions f_i are L-smooth, i.e., for all $\boldsymbol{x}, \boldsymbol{y} \in R^d$ and for each $i \in V$, we have

$$f_i(\boldsymbol{y}) \leq f_i(\boldsymbol{x}) + \langle \nabla f_i(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L}{2}||\boldsymbol{y} - \boldsymbol{x}||^2. \tag{3}$$

This assumption implies the local gradients $\nabla f_i(\cdot)$ are L-Lipschitz continuous meaning $||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}||$ holds for all $\boldsymbol{x}, \boldsymbol{y} \in R^d$ and for each $i \in V$.

Assumption 2: The global objective function f is lower bounded by f^*, i.e., $f(\boldsymbol{x}) \geq f^*$ for all $\boldsymbol{x} \in R^d$. Furthermore, the set of points for which the lower bound f^* is attained, i.e., $\mathcal{S}^* = \{\boldsymbol{x} \in R^d : f(\boldsymbol{x}) = f^*\}$, is non-empty.

Assumption 3: The global objective function f satisfies the PL condition for some $\mu > 0$, i.e., for all $\boldsymbol{x} \in R^d$, we have

$$||\nabla f(\boldsymbol{x})||^2 \geq 2\mu\big(f(\boldsymbol{x}) - f^*\big). \tag{4}$$

Assumption 4: For all $\boldsymbol{x} \in \mathcal{S}^*$, the network average of the gradient disagreements is upper-bounded by $\sigma_{\mathcal{S}^*}^2$ for some constant $\sigma_{\mathcal{S}^*} \geq 0$, i.e., for all $\boldsymbol{x} \in \mathcal{S}^*$, we have

$$\frac{1}{m}\sum_{i=1}^{m} ||\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})||^2 \leq \sigma_{\mathcal{S}^*}^2. \tag{5}$$

We formalize our assumption about the graph G below.

Assumption 5: The graph G is an undirected and connected graph, with a doubly-stochastic weight matrix $\boldsymbol{W} \in R^{m \times m}$ associated with it, which satisfies $W_{ij} \geq 0$ for all $i, j \in V$, $W_{ii} > 0$ for all $i \in V$, $W_{ij} > 0$ if and only if $(i, j) \in E$, and $W_{ij} = W_{ji}$ for all $i, j \in V$.

This assumption implies that for any $\boldsymbol{x} \in R^m$, the contraction $||\boldsymbol{W}\boldsymbol{x} - \mathbf{1}_m\bar{\boldsymbol{x}}|| \leq \omega||\boldsymbol{x} - \mathbf{1}_m\bar{\boldsymbol{x}}||$ holds, where $\bar{\boldsymbol{x}} = (1/m)\mathbf{1}_m^T\boldsymbol{x}$ represents the network average and $\omega < 1$ is the spectral norm of the matrix $\boldsymbol{W} - (1/m)\mathbf{1}_m\mathbf{1}_m^T$.

We use $\mathcal{N}_i = \{j : W_{ij} > 0, j \neq i\}$ to denote the neighbours of node i in the graph G. When node i wishes to share a quantity $\boldsymbol{x}_i(t) \in R^d$ with a neighbouring node $j \in \mathcal{N}_i$ at time t, node j receives a noise-corrupted version of $\boldsymbol{x}_i(t)$ given by

$$\widetilde{\boldsymbol{x}}_{ij}(t) = \boldsymbol{x}_i(t) + \boldsymbol{n}_{ij}(t), \tag{6}$$

where $\boldsymbol{n}_{ij}(t) \in R^d$ is the noise introduced in the communication link connecting node i to node j. We make the following assumption about the communication noise.

Assumption 6: The noise vector $\boldsymbol{n}_{ij}(t) \in R^d$ satisfies

$$E\big[\boldsymbol{n}_{ij}(t)|\boldsymbol{x}_i(t)\big] = \mathbf{0}_d, \tag{7a}$$

$$E\Big[||\boldsymbol{n}_{ij}(t)||^2|\boldsymbol{x}_i(t)\Big] \leq \sigma_c^2, \tag{7b}$$

for some $\sigma_c > 0$.

Assumption 6 encompasses a variety of noise sources including but not limited to channel noise (see [1], [2], [3], [4], [5]), noise added for privacy preservation (see [6], [7], [8]) and quantization noise due to the use of certain types of unbiased quantizers (see [9], [10], [11]).

We now briefly review the algorithm proposed in [1], which is summarized in *Algorithm 1* for completeness, with the variables used in *Algorithm 1* summarized in Table I.

Step 4 in Algorithm 1 is the standard *Noisy Consensus + SGD* step, with $\boldsymbol{h}_i(t)$ used as an estimate for the local gradient and W_{ij} is the weight that node i assigns to the estimate

Algorithm 1 Noisy Consensus + Stochastic Gradient Descent With Skipping

Input: K, θ, η, c.
Initialization: For each $i \in V$, initialize $\boldsymbol{x}_i(0) = \mathbf{0}_d$.
for $t = 0, 1, \cdots, K-1$ **do**
  for each $i \in V$ **do**

1. Draw $\chi_i(t) \sim Bernoulli(\theta)$. (8)

2. Set $\boldsymbol{h}_i(t) = \begin{cases} \frac{1}{\theta}\boldsymbol{g}_i(\boldsymbol{x}_i(t)) & \text{if } \chi_i(t) == 1 \\ \mathbf{0}_d & \text{otherwise} \end{cases}$ (9)

3. Node i sends $\boldsymbol{x}_i(t)$ to all $j \in \mathcal{N}_i$ and receives $\widetilde{x}_{ji}(t)$ from all $j \in \mathcal{N}_i$.

4. Update $\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) - \eta \boldsymbol{h}_i(t) + c\sum_{j\in\mathcal{N}_i} W_{ij}\big(\widetilde{\boldsymbol{x}}_{ji}(t) - \boldsymbol{x}_i(t)\big)$. (10)

  end for
end for

TABLE I
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
θ	Probability with which $\chi_i(t)$ takes the value 1
$\boldsymbol{x}_i(t)$	Node i's estimate at iteration t
$\widetilde{\boldsymbol{x}}_{ji}(t)$	Node j's noise-corrupted estimate received by node i at iteration t
$\boldsymbol{h}_i(t)$	Quantity used by node i as a stochastic approximation of its local gradient $\nabla f_i(\boldsymbol{x}_i(t))$ at iteration t
$\boldsymbol{g}_i(\boldsymbol{x}_i(t))$	Stochastic gradient computed by node i at iteration t if and only if $\chi_i(t) = 1$
η, c	Step-size parameters

received rom node j. We then have the following result about $\boldsymbol{h}_i(t)$ from [1, Lemma 1].

$$E\big[\boldsymbol{h}_i(t)\big|\boldsymbol{x}_i(t)\big] = \nabla f_i(\boldsymbol{x}_i(t)), \tag{11a}$$

$$E\Big[\big|\big|\boldsymbol{h}_i(t) - \nabla f_i(\boldsymbol{x}_i(t))\big|\big|^2\big|\boldsymbol{x}_i(t)\Big] \le \frac{1}{\theta}\sigma_g^2 + \left(\frac{1}{\theta} - 1\right)||\nabla f_i(\boldsymbol{x}_i(t))||^2. \tag{11b}$$

While the unbiasedness property is retained in (11b), we see from comparing (11b) and (2b) that $\boldsymbol{h}_i(t)$ has significantly higher variance than $\boldsymbol{g}_i(\boldsymbol{x}_i(t))$, with the σ_g^2 term scaled up by a factor of $1/\theta$ followed by addition with the term $(1/\theta - 1)||\nabla f_i(\boldsymbol{x}_i(t))||^2$. However, the fact that computation of the local stochastic gradient $\boldsymbol{g}_i(\boldsymbol{x}_i(t))$ is randomly skipped with probability $1-\theta$ means that the average computational cost per iteration is also reduced by a factor of $1/\theta$. The presence of these two opposing forces (i.e., higher variance associated with $\boldsymbol{h}_i(t)$ versus lower per-iteration computational complexity) then raises the following questions: (1) is the overall computational complexity (i.e., the per-iteration computational complexity times the total number of iterations) reduced, and (2) is there any detrimental effect on the overall convergence of the algorithm due to the increased variance of the gradient estimate $\boldsymbol{h}_i(t)$. The result in [1] showed that under a general smoothness assumption (specifically, Assumption 1 in the current work), the overall computational complexity can indeed be reduced by such an operation without any detrimental effect on the overall convergence of the algorithm, as opposed to the vanilla *Noisy Consensus + SGD* method, where the stochastic gradient is computed across all iterations (i.e., θ is set to 1). We show in the next section that this effect carries over to the PL setting as well, along with additional improvements in both the computational and communication costs as compared to the results obtained in [1].

III. CONVERGENCE ANALYSIS

First, we define the following quantities.

$$\boldsymbol{X}(t) = \big[\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_m(t)\big] \in R^{d\times m},$$
$$\nabla F\big(\boldsymbol{X}(t)\big) = \big[\nabla f_1\big(\boldsymbol{x}_1(t)\big), \ldots, \nabla f_m\big(\boldsymbol{x}_m(t)\big)\big] \in R^{d\times m}.$$

We also use $\bar{\boldsymbol{x}}(t) = (1/m)\sum_{i=1}^m \boldsymbol{x}_i(t)$ to represent the network average of the nodes' estimates at time t.

We are now ready to state our result showing the reduction in complexity for PL functions, which is summarized below.

Theorem 1: Let Assumptions 1–6 hold. Then Algorithm 1 with the parameter choices

$$\eta = \frac{16log(K)}{\mu K}, c = \frac{m^{1/4}log(K)}{K^{3/4}}, \theta = \frac{m^\gamma}{K^\tau}, \tag{12}$$

where $\gamma \in R$ and $\tau \in [0, 1/2]$ are chosen such that $\theta \le 1$, satisfies the following bound for sufficiently large K

$$\begin{aligned}
&\max\left\{\frac{1}{m}\sum_{i=1}^m E\big[f(\boldsymbol{x}_i(K)) - f^*\big], \frac{1}{m}\sum_{i=1}^m E\big[||\boldsymbol{x}_i(K) - \bar{\boldsymbol{x}}(K)||^2\big]\right\} \\
&\le O\left(\frac{1}{K^2}\right)\big[f(\bar{x}(0)) - f^*\big] + O\left(\frac{log(K)}{m^{1+\gamma}K^{1-\tau}}\right)\sigma_g^2 \\
&+ O\left(\frac{log(K)}{m^{1/2}K^{1/2}} + \frac{m^{1/2}log(K)}{K^{3/4}} + \frac{log(K)}{m^{3/4+\gamma}K^{7/4-\tau}}\right)\sigma_c^2 \\
&+ O\left(\frac{log(K)}{m^{1+\gamma}K^{1-\tau}} + \frac{1}{m^{1/2}K^{1/2}} + \frac{log(K)}{m^{1/2+\gamma}K^{5/4-\tau}}\right)\sigma_{\mathcal{S}^*}^2.
\end{aligned} \tag{13}$$

Moreover, for any node $i \in V$, the expected number of stochastic gradient computations up to time K is given by

$$\sum_{s=0}^{K-1} E\big[\chi_i(s)\big] = \theta K = m^\gamma K^{1-\tau}. \tag{14}$$

Proof: We bound the first term in the LHS of (13) as follows

$$\begin{aligned}
&\frac{1}{m}\sum_{i=1}^m E\big[f(\boldsymbol{x}_i(K)) - f^*\big] \\
&= \frac{1}{m}\sum_{i=1}^m E\big[f(\boldsymbol{x}_i(K)) - f(\bar{\boldsymbol{x}}(K))\big] + E\big[f(\bar{\boldsymbol{x}}(K)) - f^*\big] \\
&\le \frac{1}{m}\sum_{i=1}^m E\big[\langle\nabla f(\bar{\boldsymbol{x}}(K)), \boldsymbol{x}_i(K) - \bar{\boldsymbol{x}}(K)\rangle\big]
\end{aligned}$$

$$+\frac{L}{2m}\sum_{i=1}^{m}E\big[||\boldsymbol{x}_i(K)-\bar{\boldsymbol{x}}(K)||^2\big]+E\big[f(\bar{\boldsymbol{x}}(K))-f^*\big]$$
$$=\frac{L}{2m}E\big[||\boldsymbol{X}(K)-\bar{\boldsymbol{x}}(K)\mathbf{1}_m^T||_F^2\big]+E\big[f(\bar{\boldsymbol{x}}(K))-f^*\big]. \quad (15)$$

To bound the $E[||\boldsymbol{X}(K)-\bar{\boldsymbol{x}}(K)\mathbf{1}_m^T||_F^2]$ and $E[f(\bar{\boldsymbol{x}}(K))-f^*]$ terms, we use the following intermediate results from [1] ([1, Lemmas 3 and 4] respectively), which hold for all times $t\geq 0$.

$$E\left[||\boldsymbol{X}(t+1)-\bar{\boldsymbol{x}}(t+1)\mathbf{1}_m^T||_F^2\right]$$
$$\leq\big(1-(1-\omega)c\big)E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]$$
$$+\left(\frac{4\eta^2L^2}{(1-\omega)c}+\frac{2\eta^2L^2}{\theta}\right)E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]$$
$$+\left(\frac{4\eta^2}{(1-\omega)c}+\frac{2\eta^2}{\theta}\right)E\Big[||\nabla F\big(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T\big)||_F^2\Big]$$
$$+\frac{\eta^2m\sigma_g^2}{\theta}+mc^2\sigma_c^2, \quad (16)$$

$$E\left[f\big(\bar{\boldsymbol{x}}(t+1)\big)\right]\leq E\big[f\big(\bar{\boldsymbol{x}}(t)\big)\big]-\frac{\eta}{2}E\big[||\nabla f\big(\bar{\boldsymbol{x}}(t)\big)||^2\big]$$
$$+\left(\frac{\eta L^2+2\eta^2L^3}{2m}+\frac{\eta^2L^3}{\theta m^2}\right)E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]$$
$$+\left(\frac{\eta^2L}{\theta m^2}+\frac{\eta^2L}{m}\right)E\big[||\nabla F\big(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T\big)||_F^2\big]+\frac{\eta^2L\sigma_g^2}{2m\theta}+\frac{c^2L\sigma_c^2}{2m}. \quad (17)$$

To further simplify the terms in the RHS of (16) and (17), we use the fact that since the global objective function f satisfies the PL condition, it must also satisfy the "Quadratic Growth" condition, i.e., for all $\boldsymbol{x}\in R^d$, we must have

$$f(\boldsymbol{x})-f^*\geq\frac{\mu}{2}||\boldsymbol{x}-\boldsymbol{x}_p||^2, \quad (18)$$

where $\boldsymbol{x}_p$ is the Euclidean projection of $\boldsymbol{x}$ on $\mathcal{S}^*$. We refer the interested reader to [14, Th. 2] for a proof of this result. This then allows us to simplify the $||\nabla F(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T)||^2$ term as follows

$$\big|\big|\nabla F\big(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T\big)\big|\big|_F^2$$
$$\leq 2\big|\big|\nabla F\big(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T\big)-\nabla F\big(\bar{\boldsymbol{x}}(t)_p\mathbf{1}_m^T\big)\big|\big|_F^2$$
$$+2\big|\big|\nabla F\big(\bar{\boldsymbol{x}}(t)_p\mathbf{1}_m^T\big)\big|\big|_F^2$$
$$\leq 2mL^2||\bar{\boldsymbol{x}}(t)-\bar{\boldsymbol{x}}(t)_p||^2+2\sum_{i=1}^{m}||\nabla f_i\big(\bar{\boldsymbol{x}}(t)_p\big)||^2$$
$$\leq\frac{4mL^2}{\mu}\big[f\big(\bar{\boldsymbol{x}}(t)\big)-f^*\big]+2m\sigma_{\mathcal{S}^*}^2, \quad (19)$$

where we have used (18) and (5) in the last inequality and the fact that the local gradients ∇f_i are L-Lipschitz continuous (i.e., Assumption 1) in the second to last inequality. Using (19) and the fact that $\frac{4\eta^2L^2}{(1-\omega)c}+\frac{2\eta^2L^2}{\theta}\leq\frac{(1-\omega)c}{2}$ holds from our step-size choice for sufficiently large K, we can simplify the RHS in (16) as follows.

$$E\left[||\boldsymbol{X}(t+1)-\bar{\boldsymbol{x}}(t+1)\mathbf{1}_m||_F^2\right]$$
$$\leq\left(1-\frac{(1-\omega)c}{2}\right)E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]$$
$$+\frac{L^2}{\mu}\left(\frac{16\eta^2}{(1-\omega)c}+\frac{8\eta^2}{\theta}\right)mE\big[f\big(\bar{\boldsymbol{x}}(t)\big)-f^*\big]$$
$$+2\left(\frac{4\eta^2}{(1-\omega)c}+\frac{2\eta^2}{\theta}\right)m\sigma_{\mathcal{S}^*}^2+\frac{\eta^2m\sigma_g^2}{\theta}+mc^2\sigma_c^2. \quad (20)$$

Similarly, in (17), if we use the inequality in (19) to bound the terms containing $||\nabla F(\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T)||^2$ and use the PL condition to bound the term containing $||\nabla f(\bar{\boldsymbol{x}}(t))||^2$, we get

$$E\left[f\big(\bar{\boldsymbol{x}}(t+1)\big)-f^*\right]$$
$$\leq\left(1-\eta\mu+\frac{4\eta^2L^3}{\theta m\mu}+\frac{4\eta^2L^3}{\mu}\right)E\big[f\big(\bar{\boldsymbol{x}}(t)\big)-f^*\big]$$
$$+\left(\frac{\eta L^2+\eta^2L^3}{m}+\frac{\eta^2L^3}{\theta m^2}\right)E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]$$
$$+2\left(\frac{\eta^2L}{\theta m}+\eta^2L\right)\sigma_{\mathcal{S}^*}^2+\frac{\eta^2L\sigma_g^2}{2m\theta}+\frac{c^2L\sigma_c^2}{2m}. \quad (21)$$

Further, noticing that the conditions $\frac{4\eta^2L^3}{\theta m\mu}+\frac{4\eta^2L^3}{\mu}\leq\frac{\mu\eta}{2}$ and $\eta L\leq\theta\leq 1$ hold for our choice of step-sizes for sufficiently large K, we obtain

$$E\left[f\big(\bar{\boldsymbol{x}}(t+1)\big)-f^*\right]\leq\left(1-\frac{\eta\mu}{2}\right)E\big[f\big(\bar{\boldsymbol{x}}(t)\big)-f^*\big]$$
$$+\frac{3\eta L^2}{m}E\big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\big]+2\left(\frac{\eta^2L}{\theta m}+\eta^2L\right)\sigma_{\mathcal{S}^*}^2$$
$$+\frac{\eta^2L\sigma_g^2}{2m\theta}+\frac{c^2L\sigma_c^2}{2m}, \quad (22)$$

Let us define the vector $\boldsymbol{u}(t)\in R^2$ as follows

$$\boldsymbol{u}^T(t)=\left[\frac{1}{m}E\Big[||\boldsymbol{X}(t)-\bar{\boldsymbol{x}}(t)\mathbf{1}_m^T||_F^2\Big],\ \frac{\mu}{12L^2}E\big[f\big(\bar{\boldsymbol{x}}(t)\big)-f^*\big]\right].$$

We then have the following matrix inequality

$$\boldsymbol{u}(t+1)\leq\boldsymbol{A}\boldsymbol{u}(t)+\boldsymbol{b}, \quad (23)$$

where $\boldsymbol{A}\in R^{2\times 2}$ is given by

$$\boldsymbol{A}=\begin{bmatrix}1-\frac{(1-\omega)c}{2} & \frac{12L^4}{\mu^2}\left(\frac{16\eta^2}{(1-\omega)c}+\frac{8\eta^2}{\theta}\right)\\ \frac{\mu\eta}{4} & 1-\frac{\eta\mu}{2}\end{bmatrix} \quad (24)$$

and $\boldsymbol{b}\in R^2$ given by

$$\boldsymbol{b}=\begin{bmatrix}2\left(\frac{4\eta^2}{(1-\omega)c}+\frac{2\eta^2}{\theta}\right)\sigma_{\mathcal{S}^*}^2+\frac{\eta^2\sigma_g^2}{\theta}+c^2\sigma_c^2\\ \left(\frac{\eta^2\mu}{6\theta mL}+\frac{\eta^2\mu}{6L}\right)\sigma_{\mathcal{S}^*}^2+\frac{\eta^2\sigma_g^2\mu}{24\theta mL}+\frac{c^2\sigma_c^2\mu}{24mL}\end{bmatrix}. \quad (25)$$

Unrolling the above matrix inequality, we have

$$||\boldsymbol{u}(K)||\leq||\boldsymbol{A}^K\boldsymbol{u}(0)+\big(\boldsymbol{I}_2-\boldsymbol{A}^K\big)(\boldsymbol{I}_2-\boldsymbol{A})^{-1}\boldsymbol{b}||$$
$$\leq\rho(\boldsymbol{A})^K||\boldsymbol{u}(0)||+\big(1+\rho(\boldsymbol{A})^K\big)||(\boldsymbol{I}_2-\boldsymbol{A})^{-1}\boldsymbol{b}|| \quad (26)$$

where $\rho(\boldsymbol{A})$ is the spectral norm of the matrix $\boldsymbol{A}$. Using some routine algebraic calculations, it can be shown that we have following bound for $\rho(\boldsymbol{A})$ for sufficiently large K.

$$\rho(\boldsymbol{A})=\sqrt{\lambda_{max}\big(\boldsymbol{A}^T\boldsymbol{A}\big)}\leq\sqrt{1-\frac{\eta\mu}{4}}. \quad (27)$$

Then we have

$$\rho(\boldsymbol{A})^K \leq \left(1-\frac{\eta\mu}{4}\right)^{K/2} \leq exp\left(-\frac{\eta\mu K}{8}\right). \tag{28}$$

Noticing that $||\boldsymbol{X}(0)-\bar{\boldsymbol{x}}(0)\boldsymbol{1}_m^T||=0$ holds from the initialization condition, we have

$$||\boldsymbol{u}(0)|| = \left(\mu/12L^2\right)\left(f\left(\bar{x}(0)\right)-f^*\right). \tag{29}$$

To bound the second term in (26), we compute the determinant of $\boldsymbol{I}_2-\boldsymbol{A}$ as follows

$$det(\boldsymbol{I}_2-\boldsymbol{A}) = \frac{(1-\omega)\mu\eta c}{4} - \frac{L^4}{\mu}\left(\frac{48\eta^3}{(1-\omega)c}+\frac{24\eta^3}{\theta}\right)$$
$$\implies \frac{(1-\omega)\mu\eta c}{8} \leq det(\boldsymbol{I}_2-\boldsymbol{A}) \leq \frac{(1-\omega)\mu\eta c}{4},$$

where the condition $\frac{L^4}{\mu}(\frac{48\eta^3}{(1-\omega)c}+\frac{24\eta^3}{\theta}) \leq \frac{(1-\omega)\mu\eta c}{8}$ used in the last inequality holds given our step-size choice for sufficiently large K. We then have

$$(\boldsymbol{I}_2-\boldsymbol{A})^{-1} = adjoint(\boldsymbol{I}_2-\boldsymbol{A})/det(\boldsymbol{I}_2-\boldsymbol{A})$$
$$\leq \begin{bmatrix} \frac{4}{(1-\omega)c} & -\frac{48L^4}{(1-\omega)\mu^3}\left(\frac{16\eta}{(1-\omega)c^2}+\frac{8\eta}{\theta c}\right) \\ -\frac{1}{(1-\omega)c} & \frac{4}{\mu\eta} \end{bmatrix} \tag{30}$$

Combining and plugging in the values of η, c and θ from (12) in (25), (26), (28), (29) and (30), and noting that the RHS of (15) $\leq (24L^2/\mu)\times$ LHS of (26), we obtain (13). This concludes the proof. ■

We note that the slowest decaying term in (13) for $\tau \in [0, 1/2]$ has a $log(K)/K^{1/2}$ dependence (see the term associated with σ_c^2). However, if we restrict our attention to the dependence of K in the σ_g^2 term in (13), we notice that the slowest decaying term associated with σ_g^2 scales as $log(K)/K^{1-\tau}$. For no random skipping, i.e., when both τ and γ are set to zero (so that $\theta=1$), this turns out to be a $log(K)/K$ dependence on K. However, even in such case, the overall convergence rate continues to scale as $log(K)/K^{1/2}$ (as determined by the term associated with σ_c^2 in (13)). This means that setting $\tau=0$ and computing the stochastic gradient across all iterations would be an overkill and can be avoided. In other words, we can afford some degradation in the "quality" of the gradient estimates available to us, i.e., we should be able to retain the overall convergence rate of $log(K)/K^{1/2}$ even if the gradient estimates have higher variance than σ_g^2. As we already saw in (11b), the variance of the gradient estimate $\boldsymbol{h}_i(t)$ is controlled by the parameter θ, whose value in turn is controlled through the variable τ. In the following result we choose the highest possible value of τ to optimize for the per-iteration computational cost, while still ensuring that we retain the overall $log(K)/K^{1/2}$ convergence rate.

Corollary 1: Choosing $\tau = 1/2$ and $\gamma = -1/2$ in Theorem 1 gives the following bound.

$$\frac{1}{m}\sum_{i=1}^m E\left[f(\boldsymbol{x}_i(K))-f^*\right] \leq O\left(\frac{log(K)}{m^{1/2}K^{1/2}}+\frac{m^{1/2}log(K)}{K^{3/4}}\right)$$

Consequently, for sufficiently small ϵ, at any given node $i \in V$, the number of iterations and the number of stochastic gradient computations required to compute an ϵ-accurate solution (i.e., to ensure that the LHS in Corollary 1 is no greater than ϵ) are respectively given by

$$K_\epsilon \leq O\left(\frac{log^2(1/\epsilon)}{m\epsilon^2}\right),$$
$$\sum_{s=0}^{K_\epsilon-1} E\left[\chi_i(s)\right] = m^{-1/2}K_\epsilon^{1/2} \leq O\left(\frac{log(1/\epsilon)}{m\epsilon}\right),$$

thus proving the theoretical claims made in the *Introduction* section.

Remark 1: It is to be noted here that Assumption 4 in the current work requires the network average of the gradient disagreements to be bounded only at points in the set $\mathcal{S}^*$, whereas the results obtained in [1] required the gradient disagreements to be bounded for all $x \in R^d$. Thus, the inclusion of the PL condition as an additional assumption in the current work has allowed us to relax the assumption about bounded gradient disagreements as compared to the work in [1] which considered only the smoothness assumption without the PL condition.

Remark 2: Also note that the results obtained in the current work are applicable in environments with unreliable computational resources (for e.g., environments witnessing frequent power outages or having straggler nodes). In such environments, there are disruptions in the gradient computation step, resulting in stochastic gradients being computed successfully with a certain small probability. The result in Theorem 1 and by extension in Corollary 1 suggest that the $log(K)/K^{1/2}$ dependence in the overall convergence rate will remain unaffected as long as the probability of successful stochastic gradient computation is above a small threshold which has a $1/K^{1/2}$ dependence on K. This interpretation from a robustness standpoint (i.e., robustness of the algorithm against unreliable computational resources) has been explored in more detail in [1], where the skipping technique was originally proposed and we refer the interested reader to the same for a more comprehensive treatment.

IV. Numerical Experiments

We consider a set of $m=20$ nodes connected over a graph, such that for each pair of nodes i, j, the probability of an edge connecting them is 0.75. The network matrix $\boldsymbol{W} \in R^{20\times 20}$ is set to $\boldsymbol{W} = \boldsymbol{I}_{20} - (3/4\lambda_{max}(\boldsymbol{L}))\boldsymbol{L}$, where $\boldsymbol{L} \in R^{20\times 20}$ is the Laplacian of the graph and $\lambda_{max}(\boldsymbol{L})$ is its maximum eigenvalue.

The local component function at node i is chosen in the $d=1$ dimensional space as follows:

$$f_i(\boldsymbol{x}) = p_i(\boldsymbol{x}-3)^2 + q_i sin^2(\boldsymbol{x}-3) + r_i sin(\boldsymbol{x}-3)$$
$$+s_i log(1+exp(-(\boldsymbol{x}-3))) + t_i.$$

Here p_i, q_i, r_i, s_i and t_i are randomly chosen such that $\sum_{i=1}^{20} p_i = 1$, $\sum_{i=1}^{20} q_i = 3$, $\sum_{i=1}^{20} r_i = 0$, $\sum_{i=1}^{20} s_i = 0$ and $\sum_{i=1}^{20} t_i = 0$, with some of p_i, q_i, r_i, s_i and t_i taking negative values. The global objective function $f(\boldsymbol{x}) = (1/20)((\boldsymbol{x}-3)^2 + 3sin^2(\boldsymbol{x}-3))$ satisfies the PL condition (see [14]). The communication noise is modeled as zero-mean Gaussian noise with variance $\sigma_c^2 = 0.1$.

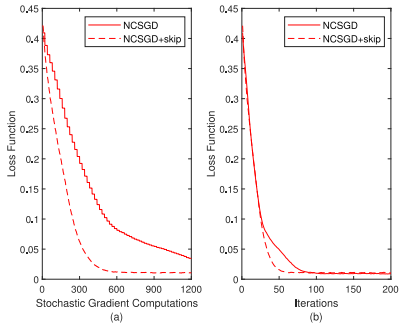

Fig. 1. Numerical Experiment Results.

We note that if we substitute $\theta = 1$ in the algorithm, we get back the *Noisy Consensus + (Stochastic) Gradient Descent* algorithm (see for example [4], [9], [10]), which we abbreviate as `NCSGD` for brevity. We abbreviate the algorithm presented in the current work as `NCSGD+skip`, owing to the random skipping of the gradient computation step. For `NCSGD`, we tune the parameters η and c over the grid $\{0.1, 0.33, 1, 3.3\}$, and for `NCSGD+skip`, we use the tuned values of η and c from `NCSGD` and then tune θ over the grid $\{1, 1/2, 1/3, 1/4\}$ conditioned on these fixed choices of η and c. The simulation results are presented in Fig. 1, where each plot is averaged over 100 runs. In Fig. 1(a), we study the decrease in the loss function, i.e., the quantity $(1/m)\sum_{i=1}^{m} f(\boldsymbol{x}_i(t))$ as a function of the total number of stochastic gradient computations up to time t (i.e., $\sum_{s=0}^{t}\sum_{i=1}^{m}\chi_i(s)$) and observe that `NCSGD+skip` requires significantly fewer stochastic gradient computations to reach a particular value of the loss function as compared to `NCSGD`. In Fig. 1(b), we study the decrease in the aforementioned quantity $(1/m)\sum_{i=1}^{m} f(\boldsymbol{x}_i(t))$ as a function of the number of iterations and see that there is no degradation in the overall convergence of the algorithm, thus validating our theoretical result.

V. Conclusion and Future Work

In this letter, we conducted a performance analysis of the algorithm proposed in [1] with the additional PL condition and showed we require only $O(log(1/\epsilon)/\epsilon)$ stochastic gradient computations, which is a significant improvement over the $O(1/\epsilon^2)$ rate obtained in previous works [4], [9], [10] and [11] that consider the stricter strong-convexity assumption as opposed to the less restrictive PL condition considered in this letter. This improvement is achieved while retaining the $O(log^2(1/\epsilon)/\epsilon^2)$ rate of [4], [9], [10] and [11] for the number of iterations and communication rounds, showing that significant computational reduction is achieved without any sacrifice in the overall convergence time. In this process, we also improve upon the $O(1/\epsilon^2)$ and $O(1/\epsilon^3)$ rates obtained in [1] for the number of stochastic gradient computations and number of iterations respectively, which considers only the smoothness assumption without the PL condition.

In our subsequent research, we wish to investigate the performance of the skipping technique when *Gradient Tracking* ([16], [17]) is introduced in the algorithm, as *Gradient Tracking* is typically known to accelerate convergence further by enabling individual nodes to estimate the global gradient. Another interesting direction that is worth exploring is that of directed and/or dynamic graphs, where the benefits of the skipping technique are likely to accrue since it is a technique which does not depend on graph structure. The detailed exploration in this regard is left for future research.

References

[1] S. Mukherjee and M. Chakraborty, "Achieving near-optimal oracle complexity in decentralized stochastic optimization with channel noise," *IEEE Trans. Control. Netw. Syst.*, vol. 12, no. 2, pp. 1215–1226, Jun. 2025.

[2] 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.

[3] S. Pu, "A robust gradient tracking method for distributed optimization over directed networks," in *Proc. 59th IEEE Conf. Decision Control.*, Jeju, South Korea, 2020, pp. 2335–2341.

[4] H. Reisizadeh, 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] A. Reisizadeh, 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.

[10] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Proc. Adv. Neural. Inf. Process. Syst.*, 2019, pp. 8388–8399.

[11] 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. Decision Control.*, Austin, TX, USA, 2021, pp. 3117–3122.

[12] C. Liu, L. Zhu, and M. Belkin, "Loss landscapes and optimization in over-parameterized non-linear systems and neural networks," *Appl. Comput. Harmonic Anal.*, vol. 59, pp. 85–116, Jul. 2022.

[13] A. Aich, A. B. Aich, and B. Wade, "From sublinear to linear: Fast convergence in deep networks via locally Polyak-Lojasiewicz regions," 2025, *arXiv:2507.21429v1*.

[14] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2016, pp. 795–811.

[15] 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, Jan. 2012.

[16] 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, vol. 80, 2018, pp. 4848–4856.

[17] 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.