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

Gradient Tracking with Multiple Local SGD for Decentralized Non-Convex Learning

Songyang Ge[1] and Tsung-Hui Chang[1]

***Abstract*— The stochastic Gradient Tracking (GT) method for distributed optimization, is known to be robust against the inter-client variance caused by data heterogeneity. However, the stochastic GT method can be communication-intensive, requiring a large number of communication rounds of message exchange for convergence. To address this challenge, this paper proposes a new communication-efficient stochastic GT algorithm called the Local Stochastic GT (LSGT) algorithm, which adopts the local stochastic gradient descent (local SGD) technique in the GT method. With LSGT, each agent can perform multiple SGD updates locally within each communication round. Although it is not known previously whether the stochastic GT method can benefit from the local SGD, we establish the conditions under which our proposed LSGT algorithm enjoys the linear speedup brought by local SGD. Compared with the existing work, our analysis requires less restrictive conditions on the mixing matrix and algorithm stepsize. Moreover, it reveals that the local SGD does not only reserve the resilience of the stochastic GT method against the data heterogeneity but also speeds up reducing the tracking error reduction in the optimization process. The experimental results demonstrate that the proposed LSGT exhibits improved convergence speed and robust performance in various heterogeneous environments.**

I. INTRODUCTION

In recent years, the rapid advancement of information technology has led to the constant generation of data from various sources, including a wide range of sensors in daily life and the vast number of users on the internet. To instantaneously process real-time large-scale data, the concept of distributed optimization has emerged, utilizing the collective computational and communicational capabilities of a network. The application of distributed optimization is relevant to numerous areas of science and engineering, such as machine learning, signal processing, and control systems [1]–[7]. In such applications, the optimization problem in an N-agent network is often expressed as the sum of local objective functions below

$$\min_{\boldsymbol{y}\in\mathbb{R}^p} F(\boldsymbol{y}) \triangleq \frac{1}{N}\sum_{n=1}^{N} f_n(\boldsymbol{y}), \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^p$ is the parameter vector to optimize and each $f_n : \mathbb{R}^p \to \mathbb{R}$ is a smooth and possibly non-convex local cost function of agent n, $\forall n \in [N] \triangleq \{1, \ldots, N\}$. For a statistical learning problem, one may assume $f_n(\boldsymbol{y}) = \mathbb{E}_{\xi\sim\mathcal{D}_n}[\ell_n(\boldsymbol{y}, \xi)]$ where ℓ_n is a loss function of $\boldsymbol{y}$, and the data sample ξ is randomly drawn from a local dataset $\mathcal{D}_n$.

The challenge of solving the distributed problem (1) lies in coordinating the cooperation among distributed agents with potentially disparate data, while minimizing computational and communication costs and ensuring accurate and efficient resolution of the optimization problem. A range of algorithms and techniques have been developed to achieve this goal, including distributed gradient descent (DGD) [8], decentralized stochastic gradient descent (DSGD) [9], [10], D^2 [11], and the gradient tracking (GT) method [12]–[16]. The DGD algorithm leverages consensus gradient descent (GD) to address the problem (1). However, when local datasets of agents have different statistical properties, DGD cannot attain an optimal solution. This limitation arises due to the gradient bias generated among agents by heterogeneous data, which slows down the convergence rate [11], [17], or even results in local overfitting, known as client drift. Moreover, DSGD relies on the condition of $\frac{1}{N}\sum_{n=1}^{N}|\nabla f_n(\boldsymbol{y}) - \nabla F(\boldsymbol{y})| \leq \varsigma$, which exhibits slower convergence as ς increases.

To address the above challenge of processing heterogeneous data, D^2 corrects the bias term by storing the gradients of previous iterations. However, D^2 stringently requires a sufficiently connected network. Compared to D^2, the GT method has been proven to be more resilient against data heterogeneity. It achieves this by incorporating local auxiliary variables that track global gradients, enabling agents to perform a virtual centralized GD scheme. With a more relaxed constraint on the mixing matrix than D^2 and a single stepsize parameter, it can converge to the neighborhood of a stationary solution sublinearly.

To further enhance computational efficiency, stochastic GT with stochastic GD (SGD) updates has been examined in recent studies, such as [14], [18] for strongly convex problems and GNSD [15] and GT-DSGD [19] for non-convex problems. Additionally, the stochastic gradient tracking (GT) method eliminates the reliance on the aforementioned constraint of DSGD algorithms, which implies that the stochastic GT method maintains its efficacy in the presence of heterogeneous data. However, it is worth noting that all of the aforementioned algorithms require significant communication resources to achieve convergence.

Accordingly, it is of paramount importance to reduce communication cost. To address this challenge, we adopt the

*T.-H. Chang is the corresponding author.

[1] S. Ge and T.-H. Chang are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, and Shenzhen Research Institute of Big Data, Shenzhen, China 518172. songyangge@link.cuhk.edu.cn; tsunghui.chang@ieee.org

The work is supported by Shenzhen Science and Technology Program under Grant No. RCJC20210609104448114 and JCYJ20190813171003723, the NSFC, China, under Grant 62071409, and by Guangdong Provincial Key Laboratory of Big Data Computing.

TABLE I: Comparison of different GT-based algorithms

Algorithm	function	gradient	stepsize	comp.	comm.	λ	$\boldsymbol{W}$	Initialization ($n \in [N]$)
GT [12]	ns-cvx.	full	cst.	$\mathcal{O}(\frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\epsilon})$	$(-1,1)$	d.s.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
DSGT [14]	cvx.	stochastic	dimi.	$\mathcal{O}(\frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\epsilon})$	$(-1,1)$	d.s.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
GNSD [15]	n-cvx.	stochastic	cst.	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{\epsilon^2})$	$(-1,1)$	d.s. and symm.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
GT-DSGD [19]	n-cvx.	stochastic,	cst.	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{\epsilon^2})$	$(-1,1)$	d.s.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
D^2 [11]	n-cvx.	stochastic	cst.	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{\epsilon^2})$	$(-\frac{1}{3},1)$	r.s. and symm.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
LU-GT [20]	n-cvx.	E-step full	cst.	$\mathcal{O}(\frac{1}{\epsilon})$	$\mathcal{O}(\frac{1}{\epsilon})$	$(-1,1)$	d.s. and symm.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$
Periodical GT [21]	n-cvx.	E-step stochastic	cst.	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{E\epsilon^2})$	$(-1,1)$	d.s. and symm.	$\boldsymbol{v}_n^0 = \frac{1}{N}\sum_{i=1}^N \boldsymbol{g}_i^0$
LSGT (Proposed)	n-cvx.	E-step stochastic	cst.	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{E\epsilon^2})$	$(-1,1)$	d.s.	$\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$

"cvx.", "n-cvx." and "ns-cvx." manifests convex, nonconvex, and nonstrongly convex, "comp." denotes computation complexity, "comm." represents communication complexity, "cst." and "dimi." are the abbreviation of constant and diminishing, "d.s." indicates doubly stochastic, "r.s." shows right stochastic, "symm." suggests symmetric, ϵ is the solution accuracy, E is the local update number, $\boldsymbol{W}$ is the mixing matrix, λ is $\boldsymbol{W}$'s eigenvalue except the largest eigenvalue, $\boldsymbol{v}_n^0, n \in [N]$ denotes the initialized tracking variable, $\boldsymbol{g}_n^0 = \frac{1}{N}\sum_{i=1}^N \frac{1}{|\mathcal{I}_i|}\sum_{\xi\in\mathcal{I}_i}\nabla \ell_i(\boldsymbol{y}_i^0;\xi)$ is the stochastic gradient at agent n, and $\mathcal{I}_n \subseteq \mathcal{D}_n$ represents the agent n's randomly chosen mini-batch data set.

local stochastic gradient descent (SGD) technique [22], [23] to diminish the number of communication rounds required by convergence. Instead of a single SGD update in existing stochastic GT methods, our approach involves each agent performing multiple local SGD updates within every communication round. The local SGD technique has been effectively applied in the federated learning framework, as demonstrated in FedAvg [24], FedLin [25], and SCAFFOLD [26], with a central server used to aggregate and broadcast information. Furthermore, both theoretical and empirical evidence [22], [23] suggest that the local SGD technique can effectively accelerate communication under appropriate conditions.

However, the local SGD technique in fully decentralized networks has been less studied. As the local update number increases, client-drift can exacerbate, potentially hindering improvements in communication efficiency. Therefore, integrating local SGD with the GT method while ensuring heterogeneity independence poses a non-trivial challenge. For instance, recent work proposed a deterministic locally updated GT method, as in [20], which required an extra stepsize and a symmetric mixing matrix. Furthermore, it remains unclear under what circumstances local updates reduce communication costs in convergence analysis. A concurrent study [21] considered multi-step stochastic GT but required consensus initialization of the tracking variable, which resulted in an unnecessary increase in communication costs across a connected network.

In this paper, we propose a new decentralized algorithm, termed as the local stochastic GT (LSGT) algorithm, by integrating the local SGD technique into the stochastic GT method. With LSGT, each agent can perform multiple local SGD updates to imitate centralized SGD scheme. Theoretically, we build more relaxed conditions under which our proposed LSGT algorithm enjoys the linear speedup with a $\mathcal{O}(\frac{1}{E\epsilon^2})$ communication complexity, compared with existing work summarized in Table I. Moreover, it reveals the insights that LSGT reserves the bias-correction advantage of GT method while accelerating the tracking process for estimating the global gradient. We empirically describe the impact of the local updates, stepsize, and the network topologies on the convergence. Moreover, simulation results demonstrate that our proposed LSGT algorithm is robust against heterogeneous data while reducing the communication cost.

Notation: $\boldsymbol{I}_n$ is the n by n identity matrix, and $\mathbf{1}$ is the all-one vector. $A_{i,j}$ is the (i,j)-th element of matrix $\boldsymbol{A}$. $\otimes$ denotes the Kronecker product; $\boldsymbol{a}^\top$ and $\boldsymbol{A}^\top$ perspectively represent the transpose operation of vector $\boldsymbol{a}$ and matrix $\boldsymbol{A}$; $\langle \boldsymbol{a}, \boldsymbol{b}\rangle$ represents the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\|\boldsymbol{a}\|$ is the Euclidean norm, and $\|\boldsymbol{A}\|$ represents the largest singular value of $\boldsymbol{A}$; $\|\boldsymbol{A}\|_F$ denotes the matrix Frobenius norm.

II. Problem Setting

We are interested in finding a first-order stationary point of problem (1) over a multi-agent network.

Firstly, we model the connections among distributed agents in the multi-agent network as an undirected graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{E}$ represents the set of edges and $\mathcal{V} = [N]$ denotes the set of agents. Subsequently, the mixing matrix for the specified graph can be defined as $\boldsymbol{W} \in \mathbb{R}^{N\times N}$, where $W_{n,m} > 0$ denotes the scaling factor for the information received by agent n from agent m if and only if the edge $(n,m) \in \mathcal{E}$, and $W_{n,m} = 0$ otherwise. Moreover, we have the following standard assumptions.

Assumption 1 *The underlying graph is connected.*

Assumption 2 *The mixing matrix $\boldsymbol{W}$ is doubly stochastic satisfying*

$$\boldsymbol{W}\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^\top \boldsymbol{W} = \mathbf{1}, \quad |\lambda_w| < 1, \tag{2}$$

where λ_w is the second largest eigenvalue of $\boldsymbol{W}$ which implies $\|\boldsymbol{W} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\| < 1$.

There are many choices of such mixing matrix satisfying Assumption 2, see [15, Remark 2]. One example is the max-degree rule [27]. Notice that we require less stringent conditions on the mixing matrix, compared with the required condition $\lambda \in \left(-\frac{1}{3}, 1\right)$ for the D^2 algorithm in [11], and the required symmetric $\boldsymbol{W}$ in [11], [15], [20], [21].

Furthermore, we assume the objective function in problem (1) satisfying the standard conditions below.

Assumption 3 *The loss function $F(\boldsymbol{y}) \triangleq \frac{1}{N}\sum_{n=1}^{N} f_n(\boldsymbol{y})$ is bounded below, denoted by $F(\boldsymbol{y}) \geq \underline{F},\ \forall \boldsymbol{y}$.*

Assumption 4 *Each f_n is smooth and its gradient satisfies*

$$\|\nabla f_n(\boldsymbol{y}) - \nabla f_n(\boldsymbol{y}')\| \leq L\|\boldsymbol{y} - \boldsymbol{y}'\|,\ \forall \boldsymbol{y}, \boldsymbol{y}', \tag{3}$$

where L is the Lipschitz constant.

To improve computation efficiency, we consider stochastic local updates instead of full gradients. Specifically, we define the following stochastic gradient for each agent n as

$$\boldsymbol{g}_n \triangleq \frac{1}{|\mathcal{I}_n|}\sum_{\xi \in \mathcal{I}_n} \nabla \ell_n(\boldsymbol{y}_n; \xi), \tag{4}$$

where $\mathcal{I}_n \subseteq \mathcal{D}_n$ represents a randomly chosen mini-batch data set at agent n. Without loss of generality, we assume that $|\mathcal{I}_n| = \ldots = |\mathcal{I}_N| \triangleq |\mathcal{I}|$ and give the following standard assumption for SGD methods.

Assumption 5 *For each agent $n \in [N]$, we have*
- *Unbiased gradient:* $\mathbb{E}[\boldsymbol{g}_n] = \nabla f_n(\boldsymbol{y}_n)$;
- *Uniform bounded variance:* $\mathbb{E}[\|\boldsymbol{g}_n - \nabla f_n(\boldsymbol{y}_n)\|^2] \leq \frac{\sigma^2}{|\mathcal{I}|}$.

III. Proposed LSGT Algorithm

In this section, for solving the non-convex problem (1), we present a new decentralized communication-efficient algorithm, called *local stochastic gradient tracking (LSGT)*, by integrating the stochastic GT method [15], [16] with the local SGD technique.

Specifically, in addition to the local variable $\boldsymbol{y}_n$, each agent n constructs another auxiliary variable $\boldsymbol{v}_n \in \mathbb{R}^p$ to estimate a stochastic approximation of the global gradient $\nabla F(\boldsymbol{y}) = \frac{1}{N}\sum_{n=1}^{N} \nabla f_n(\boldsymbol{y})$. The LSGT is initialized by $\boldsymbol{y}_n^0$ and stochastic gradient $\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$ for each agent n.

In the r-th round of the algorithm, each agent $n \in [N]$ performs a sequence of E consecutive steps of SGD and tracks the updates within every communication round. Specifically, at each step $q \in [E]$, agent n randomly selects a mini-batch dataset $\mathcal{I}_n^{r,q}$ of size $|\mathcal{I}|$ and executes SGD in the direction of $\boldsymbol{v}_n^{r,q-1}$ with a positive step size γ as given in equation (6a). Following this, the local auxiliary variable $\boldsymbol{v}_n^{r,q-1}$ is updated locally by adding the new gradient $\boldsymbol{g}_n^{r,q}$ and removing the previous gradient $\boldsymbol{g}_n^{r,q-1}$, as illustrated in equation (6b). Subsequently, after completing E local updates, each agent n sends the pair $(\boldsymbol{y}_n^{r+1}, \boldsymbol{v}_n^{r+1})$ to its neighboring agents and scales the received information via $\boldsymbol{W}$. The algorithmic details of LSGT are presented in Algorithm 1. Below, we discuss the relation of the proposed LSGT algorithm with some of the existing methods.

Algorithm 1 Proposed LSGT algorithm for solving (1)

Initialize: Let $\boldsymbol{y}_1^0 = \ldots = \boldsymbol{y}_N^0$ and $\boldsymbol{v}_n^0 = \boldsymbol{g}_n^0$, $\forall n \in [N]$.
for communication round $r = 0$ **to** T **do**
  for agent $n = 1$ **to** N in parallel **do**
    Receive information from neighbours and set

$$\begin{bmatrix}\boldsymbol{y}_n^{r,0} \\ \boldsymbol{v}_n^{r,0}\end{bmatrix} = \sum_{m=1}^{N} W_{n,m}\begin{bmatrix}\boldsymbol{y}_m^{r} \\ \boldsymbol{v}_m^{r}\end{bmatrix},\ \boldsymbol{g}_n^{r,0} = \boldsymbol{g}_n^{r}. \tag{5}$$

    for local update $q = 1, \ldots, E$, **do**

$$\boldsymbol{y}_n^{r,q} = \boldsymbol{y}_n^{r,q-1} - \gamma \boldsymbol{v}_n^{r,q-1}, \tag{6a}$$
$$\boldsymbol{v}_n^{r,q} = \boldsymbol{v}_n^{r,q-1} + \boldsymbol{g}_n^{r,q} - \boldsymbol{g}_n^{r,q-1}, \tag{6b}$$

      where the stochastic gradient $\boldsymbol{g}_n^{r,q}$ is computed like (4) using $\boldsymbol{y}_n^{r,q}$ and a mini-batch $\mathcal{I}_n^{r,q} \subseteq \mathcal{D}_n$.
    end for
    Set $\boldsymbol{y}_n^{r+1} = \boldsymbol{y}_n^{r,E}, \boldsymbol{v}_n^{r+1} = \boldsymbol{v}_n^{r,E}, \boldsymbol{g}_n^{r+1} = \boldsymbol{g}_n^{r,E}$, and send $(\boldsymbol{y}_n^{r+1}, \boldsymbol{v}_n^{r+1})$ to neighbors.
  end for
end for

Remark 1 *For $E = 1$, the LSGT algorithm reduces to the vanilla stochastic GT method. For $E > 1$, the LSGT is in fact equivalent to a periodically time-varying GT method with skipped communications, where the mixing matrix at each t-th iteration can be defined as*

$$\boldsymbol{W}^t = \begin{cases}\boldsymbol{W} & \text{if} \mod (t, E) = 0, \\ \mathbf{I} & \text{otherwise.}\end{cases} \tag{7}$$

Remark 2 *(Comparison with [20]) The recent work [20] proposed the locally updated GT (LU-GT) algorithm which is also a multi-step GT method. However, there are 4 key differences between LU-GT and our proposed LSGT algorithm: i) full gradient: LU-GT considers full gradient, while we consider a computation-efficient SGD updates; ii) extra stepsize: LU-GT introduces an additional stepsize to control the update of tracking variables and guarantee the contraction property of the consensus errors, which is not needed by ours in* (6b)*; iii) symmetric $\boldsymbol{W}$: The analysis of LU-GT is based on a symmetric mixing matrix $\boldsymbol{W}$ whereas our LSGT algorithm does not.*

Remark 3 *(Comparison with [21]) The concurrent work in [21] explores decentralized gradient tracking with multiple local steps, termed as periodical GT. Their theoretical analysis relies on an extra condition for the consensus initialization, namely $\boldsymbol{v}_i^0 = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{g}_n^0, i \in [N]$. However, achieving this consensus may require several rounds of information exchange, leading to significant communication overhead. In contrast, our LSGT algorithm does not require this initialization. We will elaborate on the impact of different initializations, specifically $\boldsymbol{v}_i^0 = \boldsymbol{g}_i^0, i \in [N]$, on LSGT in Remark 5.*

IV. Convergence Analysis

In this section, we present a novel convergence analysis for the LSGT algorithm, which shows the conditions under which LSGT indeed benefits a linear speedup from the local updates.

For ease of presentation, denote $\boldsymbol{Y}^r \triangleq [\boldsymbol{y}_1^r, \ldots, \boldsymbol{y}_N^r]^\top$, $\boldsymbol{V}^r \triangleq [\boldsymbol{v}_1^r, \ldots, \boldsymbol{v}_N^r]^\top$, and define the average of local variables as $\bar{\boldsymbol{y}}^r = \frac{1}{N}\sum_{n=1}^N \boldsymbol{y}_n^r$, and $\bar{\boldsymbol{v}}^r = \frac{1}{N}\sum_{n=1}^N \boldsymbol{v}_n^r$. Then, the consensus and tracking error can be represented as the compact form below

$$\boldsymbol{\phi}^r = \begin{bmatrix}\phi_y^r \\ \phi_v^r\end{bmatrix} = \begin{bmatrix}\mathbb{E}\left[\|\boldsymbol{Y}^r - \mathbf{1}(\bar{\boldsymbol{y}}^r)^\top\|_F^2\right] \\ \mathbb{E}\left[\|\boldsymbol{V}^r - \mathbf{1}(\bar{\boldsymbol{v}}^r)^\top\|_F^2\right]\end{bmatrix} \in \mathbb{R}^2. \tag{8}$$

The main theoretical results for the LSGT method are presented in to the theorem below.

Theorem 1 *Suppose that Assumption 1 to 5 hold. For a sufficiently small $\gamma < 1$ (satisfying the conditions of [28, Theorem 1]), we have*

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^N \nabla f_n(\boldsymbol{y}_n^r)\right\|^2\right] \leq \underbrace{\frac{4(F(\bar{\boldsymbol{y}}^0)-\underline{F})}{\gamma ET} + \frac{40L\gamma\sigma^2}{N|\mathcal{I}|}}_{\text{terms same as centralized SGD}}$$
$$+ \underbrace{\frac{16(1+7\lambda_w^2)^2E^2L^2\gamma^2}{(1-\lambda_w^2)^4}\left(\frac{2577N\sigma^2}{|\mathcal{I}|}+\frac{111\phi_v^0}{T}\right)}_{\text{terms due to decentralized optimization}}. \tag{9}$$

Proof sketch: Due to limited space, the proof details are relegated to [28, Section 4]. One of the key steps in the analysis is to construct the following linear system for investigating the dynamics and the relationships between the consensus and tracking error matrix.

$$\boldsymbol{\phi}^{r+1} \leq \boldsymbol{A}\boldsymbol{\phi}^r + \boldsymbol{C}\boldsymbol{e}^r, \tag{10}$$

where for some coefficient matrix $\boldsymbol{A}$ and $\boldsymbol{C}$, and the perturbation vector

$$\boldsymbol{e}^r = \begin{bmatrix}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^N \nabla f_n(\boldsymbol{y}_n^r)\right\|^2\right] \\ \frac{\sigma^2}{|\mathcal{I}|}\end{bmatrix}. \tag{11}$$

We then determine the contraction property of $\boldsymbol{\phi}^r$ by proving the spectral radius $\rho(\boldsymbol{A}) < 1$ as long as stepsize γ is sufficiently small. The second step is to investigate how $\boldsymbol{\phi}^r$ influences the descent of the objective value in (1). By combining these two steps, we establish the convergence rate of LSGT in (9). ■

When our LSGT method reduces to the deterministic version, i.e., $\sigma^2 = 0$, then choosing $\gamma \propto 1/E$ in (9), the convergence rate of LSGT can achieve $\mathcal{O}(1/T)$, which is consistent with the convergence rate of LU-GT in [20, Theorem 1].

Theorem 1 illustrates the non-asymptotic mean-square convergence rate of the proposed LSGT algorithm in (9), which determines a solution in the neighborhood of a stationary solution to problem (1). In particular, the first two terms in the right hand side (RHS) of (9) are independent of the network topologies and comparable to those of the centralized SGD scheme [29]. Meanwhile, the impacts of the network connectivity λ_w and the initial tracking error ϕ_v^0 on the decentralized network are reflected in the last two terms.

In addition, it is demonstrated in the following corollary that our proposed LSGT algorithm enjoys the linear speedup with the local update number E and the network size N for a long run.

Corollary 1 *Let $\gamma = \sqrt{\frac{N}{ET}}$ and $E \leq (\frac{T}{N^5})^{\frac{1}{3}}$ where T is sufficiently large so that γ satisfies the conditions in Theorem 1. Then, for the proposed LSGT algorithm, we have*

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{n=1}^N \nabla f_n(\boldsymbol{y}_n^r)\right\|^2\right] \leq \frac{4(F(\bar{\boldsymbol{y}}^0)-\underline{F})}{\sqrt{NET}} + \frac{40L\sigma^2}{\sqrt{NET}|\mathcal{I}|}$$
$$\frac{16(1+7\lambda_w^2)^2L^2}{(1-\lambda_w^2)^4\sqrt{NET}}\left(\frac{2577\sigma^2}{|\mathcal{I}|}+\frac{111\phi_v^0}{NT}\right). \tag{12}$$

Corollary 1 shows that by setting $E \leq (T/N^5)^{\frac{1}{3}}$ with a large enough T, the convergence rate gap between LSGT and centralized SGD narrows up to a constant. It achieves the network-independent linear speedup with the rate of $\mathcal{O}(1/\sqrt{NET})$, which is faster than the convergence rate $\mathcal{O}(1/\sqrt{NT})$ of the vanilla stochastic GT method [19, Corollary 1]. Thus, it demonstrates LSGT indeed benefits from multiple local updates with $E > 1$ for improving communication efficiency.

Remark 4 *(Impact of network connectivity and stochastic gradient error) Both* (9) *and* (12) *suggests that as λ_w decreases, the 3rd and 4th terms of the RHS bound becomes smaller. It is because a smaller λ_w characterizes a better connected network so that $(\boldsymbol{y}_n^r, \boldsymbol{v}_n^r)$ reaches consensus faster. On the other hand, we can also derive that a larger mini-batch size $|\mathcal{I}|$ can reduce the stochastic noise and further improve the convergence performance.*

Remark 5 *(Impact of initialization discrepancy ϕ_v^0) The last term of the bound* (12) *characterizes how the initialization discrepancy ϕ_v^0 influences the convergence of LSGT. Specifically, when the number of the round T is large enough, the impact of initialization discrepancy ϕ_v^0 can be removed under proper constant stepsize. It suggests that LSGT can overcome the data heterogeneity, since ϕ_v^0 measures the difference of the local gradients. In sharp contrast, the theoretical analysis in the concurrent work [21] does not make this point clear and requires $\phi_v^0 = 0$.*

V. Experiment Results

Consider a 20-agent connected network with mixing matrix $\boldsymbol{W}$ generated by the max-degree rule [14], [27]. Assume these agents are connected by a random network graph, which is generated as in [30].

We aim to solve a image classification task of 10 handwritten digits based on the MNIST dataset [31]. The data samples are partitioned to the agents' local dataset in the IID [32] and non-IID fashion [23]. The agents collaborate to train

TABLE II:

Communication rounds to achieve a certain testing accuracy.

acc.	The IID setting				The non-IID setting			
	$E=1$	$E=5$	$E=10$	$E=50$	$E=1$	$E=5$	$E=10$	$E=50$
85%	62	14	8	**3**	75	16	**10**	11
90%	126	26	14	**4**	136	30	**21**	41
95%	335	68	35	**8**	378	91	**65**	191

"acc." denotes testing accuracy.

a two-layer deep neural network (DNN) comprising a single hidden layer with 30 neurons [33]. The Rectified Linear Unit (ReLU) and softmax functions serve as activation functions for the hidden and output layers, respectively. Cross-entropy serves as the training loss function [34]. To ensure the validity of our results, we conduct five independent trials and compute the average.

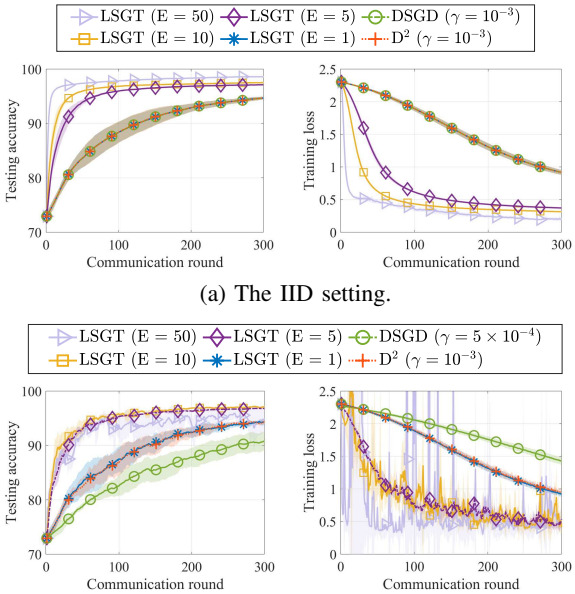
The effects of local iteration number E, stepsize γ, and network topology λ_w are respectively investigated below.

In order to examine the impact of local recursion on the performance of the proposed LSGT algorithm, we conducted an experiment in which we set local updates to $E = 1, 5, 10, 50$ for investigating the effects. The results, as shown in Figure 1(a), reveal that the training loss and testing accuracy improve more quickly as E increases for the IID case. This is due to the fact that in the IID setting, the data shared among decentralized agents are from the same distribution, and thus more local updates aid in the learning of a better common model. Conversely, Figure 1(b) shows that the performance of LSGT first improves from $E = 1$ to $E = 10$, but then worsens when $E = 50$. The reason for this is that a large E can amplify the model discrepancy among agents with non-IID data.

We further explored the relationship between the communication round and local updates E in Table II. It illustrates the communication rounds required by the LSGT algorithm with different values of E to achieve testing accuracy levels of $85\%, 90\%$, and 95%. More local updates decrease the necessary communication rounds in the IID setting, while in the non-IID case, the required communication rounds first decrease and then increase as E grows. This result is consistent with the insight provided by Theorem 1, which indicates that E should not be too large. Additionally, in Figure 1, the LSGT algorithm with $E = 5$ outperforms the DSGD and D^2 methods for both the IID and non-IID settings.

In Fig. 2, one can see that for the IID and non-IID settings, a smaller stepsize such as $\gamma = 10^{-5}$ slows down the convergence of LSGT over both training loss and testing accuracy, which demonstrates the corresponding analysis in Theorem 1.

To evaluate the influence of various network topologies, our proposed LSGT algorithm was applied to three graphs with different orders $\lambda_w(\text{line}) > \lambda_w(\text{random}) > \lambda_w(\text{complete})$, where their connectivity order was reversed. As shown in Figure 3, for the IID setting, the impact of network topology on convergence performance was found to be negligible. However, in the non-IID scenario, the LSGT algorithm exhibited higher testing accuracy when applied



(a) The IID setting.

(b) The non-IID setting.

Fig. 1: Convergence curves of the proposed LSGT algorithm with different E ($\gamma = 10^{-3}$, random graph).

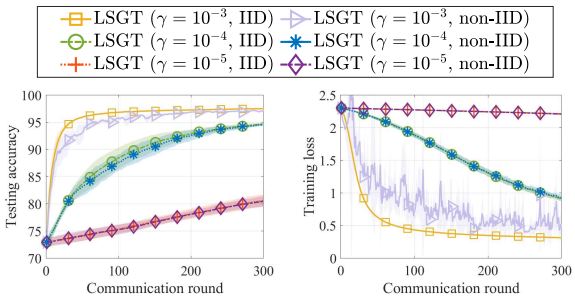


Fig. 2: Convergence curves of the proposed LSGT algorithm with different stepsize γ ($E = 10$, random graph).

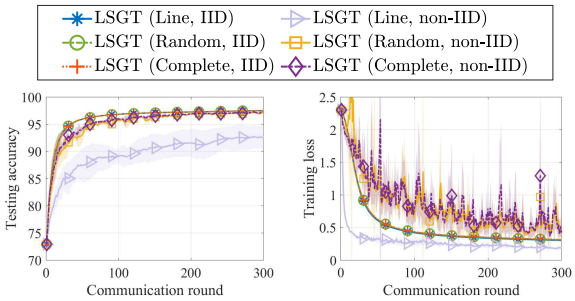


Fig. 3: Convergence curves of the proposed LSGT algorithm with different topologies ($E = 10$, $\gamma = 10^{-3}$).

to a better-connected network, such as a complete graph. This result aligns with the analysis provided in Remark 4. Notably, the proposed LSGT over a a network with stronger connectivity converged to a larger training loss. This can be attributed to the fact that as the network topology's connectivity increases, the training model approaches the global model more closely. However, when this training model is applied to local data, its efficacy may be reduced

due to the heterogeneity of the data, as discussed in [35].

VI. Conclusion

In this paper, we have proposed a novel algorithm named LSGT, which integrates the local SGD technique into the stochastic GT method to improve the communication efficiency of the current GT method for solving problem (1). We have established the convergence conditions theoretically, under which the LSGT algorithm (Theorem 1) achieves a linear speedup with the number of local SGD updates E (Corollary 1), while still being capable of handling heterogeneous data. This is in contrast to existing GT methods that either do not consider multiple steps of local SGD or fail to fully explore the benefits of local SGD for GT methods. Furthermore, our theoretical analysis is supported by the observations in the simulation results.

APPENDIX

References

[1] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 461–497, Springer, 2016.

[2] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning- Parallel and Distributed Approaches*. Cambridge University Press, 2012.

[3] G. Scutari and Y. Sun, "Parallel and distributed successive convex approximation methods for big-data optimization," in *Multi-agent Optimization*, pp. 141–308, Springer, 2018.

[4] T.-H. Chang, M. Hong, H.-T. Wai, X. Zhang, and S. Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 26–38, 2020.

[5] J. Zhang, S. Ge, T.-H. Chang, and Z.-Q. Luo, "Decentralized nonconvex learning with linearly coupled constraints: Algorithm designs and application to vertical learning problem," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3312–3327, 2022.

[6] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[7] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[8] S. S. Ram, A. Nedich, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *arXiv preprint arXiv:0811.2595*, 2008.

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

[10] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D^2: Decentralized training over decentralized data," in *Proc. Int. Conf. on Mach. Learn.*, pp. 4848–4856, 2018.

[12] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2017.

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

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

[15] S. Lu, X. Zhang, H. Sun, and M. Hong, "GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization," in *Proc. IEEE Data Sci. Workshop*, pp. 315–321, 2019.

[16] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *Proc. IEEE Conf. Decis. Control*, pp. 8353–8358, 2019.

[17] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, 2013.

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

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

[20] E. D. H. Nguyen, S. A. Alghunaim, K. Yuan, and C. A. Uribe, "On the performance of gradient tracking with local updates," *arXiv preprint arXiv:2210.04757*, 2022.

[21] Y. Liu, T. Lin, A. Koloskova, and S. U. Stich, "Decentralized gradient tracking with local steps," *arXiv preprint arXiv:2301.01313*, 2023.

[22] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[24] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[25] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik, "Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally!," in *International Conference on Machine Learning*, pp. 15750–15769, PMLR, 2022.

[26] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*, pp. 5132–5143, 2020.

[27] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[28] S. Ge and T.-H. Chang, "Gradient and variable tracking with multiple local sgd for decentralized non-convex learning," *arXiv preprint arXiv:2302.01537*, 2023.

[29] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[30] M. E. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3753–3764, 2008.

[31] Y. LeCun and C. Cortes, "Mnist handwritten digit database," *Available: http://yann.lecun.com/exdb/mnist/*, 2010.

[32] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proc. IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, pp. 1–9, 2020.

[33] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 323–341, 2021.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[35] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *Proc. IEEE 4th World Conf. on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 794–797, 2020.