A Hybrid Variance-Reduced Method for Decentralized Stochastic Non-Convex Optimization

Ran Xin [1] **Usman A. Khan** [2] **Soummya Kar** [1]

Abstract

This paper considers decentralized stochastic optimization over a network of n nodes, where each node possesses a smooth non-convex local cost function and the goal of the networked nodes is to find an ϵ-accurate first-order stationary point of the sum of the local costs. We focus on an online setting, where each node accesses its local cost only by means of a stochastic first-order oracle that returns a noisy version of the exact gradient. In this context, we propose a novel single-loop decentralized hybrid variance-reduced stochastic gradient method, called `GT-HSGD`, that outperforms the existing approaches in terms of both the oracle complexity and practical implementation. The `GT-HSGD` algorithm implements specialized local hybrid stochastic gradient estimators that are fused over the network to track the global gradient. Remarkably, `GT-HSGD` achieves a network topology-independent oracle complexity of $O(n^{-1}\epsilon^{-3})$ when the required error tolerance ϵ is small enough, leading to a linear speedup with respect to the centralized optimal online variance-reduced approaches that operate on a single node. Numerical experiments are provided to illustrate our main technical results.

1. Introduction

We consider n nodes, such as machines or edge devices, communicating over a decentralized network described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, n\}$ is the set of node indices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the collection of ordered pairs (i, j), $i, j \in \mathcal{V}$, such that node j sends information to node i. Each node i possesses a private local cost function $f_i : \mathbb{R}^p \to \mathbb{R}$ and the goal of the networked nodes

is to solve, via local computation and communication, the following optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}).$$

This canonical formulation is known as decentralized optimization (Tsitsiklis et al., 1986; Nedić & Ozdaglar, 2009; Kar et al., 2012; Chen & Sayed, 2015) that has emerged as a promising framework for large-scale data science and machine learning problems (Lian et al., 2017; Assran et al., 2019). Decentralized optimization is essential in scenarios where data is geographically distributed and/or centralized data processing is infeasible due to communication and computation overhead or data privacy concerns. In this paper, we focus on an *online and non-convex* setting. In particular, we assume that each local cost f_i is *non-convex* and each node i only accesses f_i by querying a local *stochastic first-order oracle (`SFO`)* (Nemirovski et al., 2009) that returns a stochastic gradient, i.e., a noisy version of the exact gradient, at the queried point. As a concrete example of practical interest, the `SFO` mechanism applies to many online learning and expected risk minimization problems where the noise in `SFO` lies in the uncertainty of sampling from the underlying streaming data received at each node (Kar et al., 2012; Chen & Sayed, 2015). We are interested in the oracle complexity, i.e., the total number of queries to `SFO` required at each node, to find an ϵ-accurate first-order stationary point $\mathbf{x}^*$ of the global cost F such that $\mathbb{E}[\|\nabla F(\mathbf{x}^*)\|] \leq \epsilon$.

1.1. Related Work

We now briefly review the literature of decentralized non-convex optimization with `SFO`, which has been widely studied recently. Perhaps the most well-known approach is the decentralized stochastic gradient descent (`DSGD`) and its variants (Chen & Sayed, 2015; Kar et al., 2012; Vlaski & Sayed, 2019; Lian et al., 2017; Taheri et al., 2020), which combine average consensus and a local stochastic gradient step. Although being simple and effective, `DSGD` is known to have difficulties in handling heterogeneous data (Xin et al., 2020a). Recent works (Tang et al., 2018; Lu et al., 2019; Xin et al., 2020e; Yi et al., 2020) achieve robustness to heterogeneous environments by leveraging certain

[1]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA [2]Department of Electrical and Computer Engineering, Tufts University, Medford, MA, USA. Correspondence to: Ran Xin <ranx@andrew.cmu.edu>.

decentralized bias-correction techniques such as `EXTRA` (type) (Shi et al., 2015; Yuan et al., 2020; Li & Lin, 2020), gradient tracking (Di Lorenzo & Scutari, 2016; Xu et al., 2015; Pu & Nedich, 2020; Xin et al., 2020f; Nedich et al., 2017; Qu & Li, 2017; Xi et al., 2017), and primal-dual principles (Jakovetić, 2018; Li et al., 2019; Alghunaim et al., 2020; Xu et al., 2020). Built on top of these bias-correction techniques, very recent works (Sun et al., 2020) and (Pan et al., 2020) propose `D-GET` and `D-SPIDER-SFO` respectively that further incorporate online `SARAH`/`SPIDER`-type variance reduction schemes (Fang et al., 2018; Wang et al., 2019; Pham et al., 2020) to achieve lower oracle complexities, when the `SFO` satisfies a mean-squared smoothness property. Finally, we note that the family of decentralized variance reduced methods has been significantly enriched recently, see, for instance, (Mokhtari & Ribeiro, 2016; Yuan et al., 2018; Xin et al., 2020a; Li et al., 2020b;a; Rajawat & Kumar, 2020; Xin et al., 2020b;c; Lü et al., 2020); however, these approaches are explicitly designed for empirical minimization where each local cost f_i is decomposed as a finite-sum of component functions, i.e., $f_i = \frac{1}{m} \sum_{r=1}^{m} f_{i,r}$; it is therefore unclear whether these algorithms can be adapted to the online `SFO` setting, which is the focus of this paper.

1.2. Our Contributions

In this paper, we propose `GT-HSGD`, a novel online variance reduced method for decentralized non-convex optimization with stochastic first-order oracles (`SFO`). To achieve fast and robust performance, the `GT-HSGD` algorithm is built upon global gradient tracking (Di Lorenzo & Scutari, 2016; Xu et al., 2015) and a local hybrid stochastic gradient estimator (Liu et al., 2020; Tran-Dinh et al., 2020; Cutkosky & Orabona, 2019) that can be considered as a convex combination of the vanilla stochastic gradient returned by the `SFO` and a `SARAH`-type variance-reduced stochastic gradient (Nguyen et al., 2017). In the following, we emphasize the key advantages of `GT-HSGD` compared with the existing decentralized online (variance-reduced) approaches, from both theoretical and practical aspects.

Improved oracle complexity. A comparison of the oracle complexity of `GT-HSGD` with related algorithms is provided in Table 1, from which we have the following important observations. First of all, the oracle complexity of `GT-HSGD` is lower than that of `DSGD`, `D2`, `GT-DSGD` and `D-PD-SGD`, which are decentralized online algorithms without variance reduction; however, `GT-HSGD` imposes on the `SFO` an additional mean-squared smoothness (MSS) assumption that is required by all online variance-reduced techniques in the literature (Arjevani et al., 2019; Fang et al., 2018; Wang et al., 2019; Pham et al., 2020; Liu et al., 2020; Tran-Dinh et al., 2020; Cutkosky & Orabona, 2019; Sun et al., 2020; Pan et al., 2020; Zhou et al., 2020). Secondly, `GT-HSGD` further achieves a lower oracle complexity than

the existing decentralized online variance-reduced methods `D-GET` (Sun et al., 2020) and `D-SPIDER-SFO` (Pan et al., 2020), especially in a regime where the required error tolerance ϵ and the network spectral gap $(1 - \lambda)$ are relatively small.[1] Moreover, when ϵ is small enough such that $\epsilon \lesssim \min\{\lambda^{-4}(1-\lambda)^3 n^{-1}, \lambda^{-1}(1-\lambda)^{1.5} n^{-1}\}$, it can be verified that the oracle complexity of `GT-HSGD` reduces to $O(n^{-1}\epsilon^{-3})$, independent of the network topology, and `GT-HSGD` achieves a linear speedup, in terms of the scaling with the network size n, compared with the centralized optimal online variance-reduced approaches that operate on a single node (Fang et al., 2018; Wang et al., 2019; Pham et al., 2020; Liu et al., 2020; Tran-Dinh et al., 2020; Zhou et al., 2020); see Section 3 for a detailed discussion. In sharp contrast, the speedup of `D-GET` (Sun et al., 2020) and `D-SPIDER-SFO` (Pan et al., 2020) is not clear compared with the aforementioned centralized optimal methods even if the network is fully connected, i.e., $\lambda = 0$.

More practical implementation. Both `D-GET` (Sun et al., 2020) and `D-SPIDER-SFO` (Pan et al., 2020) are double-loop algorithms that require very large minibatch sizes. In particular, during each inner loop they execute a fixed number of minibatch stochastic gradient type iterations with $O(\epsilon^{-1})$ oracle queries per update per node, while at every outer loop they obtain a stochastic gradient with mega minibatch size by $O(\epsilon^{-2})$ oracle queries at each node. Clearly, querying the oracles exceedingly, i.e., obtaining a large amount of samples, at each node and every iteration in online steaming data scenarios substantially jeopardizes the actual wall-clock time. This is because the next iteration cannot be performed until all nodes complete the sampling process. Moreover, the double-loop implementation may incur periodic network synchronizations. These issues are especially significant when the working environments of the nodes are heterogeneous. Conversely, the proposed `GT-HSGD` is a single-loop algorithm with $O(1)$ oracle queries per update and only requires a large minibatch size with $O(\epsilon^{-1})$ oracle queries once in the *initialization phase*, i.e., before the update recursion is executed; see Algorithm 1 and Corollary 1 for details.

1.3. Roadmap and Notations

The rest of the paper is organized as follows. In Section 2, we state the problem formulation and develop the proposed `GT-HSGD` algorithm. Section 3 presents the main convergence results of `GT-HSGD` and their implications. Section 4 outlines the convergence analysis of `GT-HSGD`, while the detailed proofs are provided in the Appendix. Section 5 provides numerical experiments to illustrate our theoretical claims. Section 6 concludes the paper.

---

[1] A small network spectral gap $(1 - \lambda)$ implies that the connectivity of the network is weak.

Table 1. A comparison of the oracle complexity of decentralized online stochastic gradient methods. The oracle complexity is in terms of the total number of queries to SFO required *at each node* to obtain an ϵ-accurate stationary point $\mathbf{x}^*$ of the global cost F such that $\mathbb{E}[\|\nabla F(\mathbf{x}^*)\|] \leq \epsilon$. In the table, n is the number of the nodes and $(1-\lambda) \in (0, 1]$ is the spectral gap of the weight matrix associated with the network. We note that the complexity of D2 and D-SPIDER-SFO also depends on the smallest eigenvalue λ_n of the weight matrix; however, since λ_n is less sensitive to the network topology, we omit the dependence of λ_n in the table for conciseness. The MSS column indicates whether the algorithm in question requires the mean-squared smoothness assumption on the SFO. Finally, we emphasize that DSGD requires bounded heterogeneity such that $\sup_{\mathbf{x}} \frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \zeta^2$, for some $\zeta \in \mathbb{R}^+$, while other algorithms in the table do not need this assumption.

Algorithm	Oracle Complexity	MSS	Remarks
DSGD (Lian et al., 2017)	$O\left(\max\left\{\frac{1}{n\epsilon^4}, \frac{\lambda^2 n}{(1-\lambda)^2\epsilon^2}\right\}\right)$	✗	bounded heterogeneity
D2 (Tang et al., 2018)	$O\left(\max\left\{\frac{1}{n\epsilon^4}, \frac{n}{(1-\lambda)^b\epsilon^2}\right\}\right)$	✗	$b \in \mathbb{R}^+$ is not explicitly shown in (Tang et al., 2018)
GT-DSGD (Xin et al., 2020e)	$O\left(\max\left\{\frac{1}{n\epsilon^4}, \frac{\lambda^2 n}{(1-\lambda)^3\epsilon^2}\right\}\right)$	✗	
D-PD-SGD (Yi et al., 2020)	$O\left(\max\left\{\frac{1}{n\epsilon^4}, \frac{n}{(1-\lambda)^c\epsilon^2}\right\}\right)$	✗	$c \in \mathbb{R}^+$ is not explicitly shown in (Yi et al., 2020)
D-GET (Sun et al., 2020)	$O\left(\frac{1}{(1-\lambda)^d\epsilon^3}\right)$	✓	$d \in \mathbb{R}^+$ is not explicitly shown in (Sun et al., 2020)
D-SPIDER-SFO (Pan et al., 2020)	$O\left(\frac{1}{(1-\lambda)^h\epsilon^3}\right)$	✓	$h \in \mathbb{R}^+$ is not explicitly shown in (Pan et al., 2020)
GT-HSGD (**this work**)	$O\left(\max\left\{\frac{1}{n\epsilon^3}, \frac{\lambda^4}{(1-\lambda)^3\epsilon^2}, \frac{\lambda^{1.5}n^{0.5}}{(1-\lambda)^{2.25}\epsilon^{1.5}}\right\}\right)$	✓	

We adopt the following notations throughout the paper. We use lowercase bold letters to denote vectors and uppercase bold letters to denote matrices. The ceiling function is denoted as $\lceil \cdot \rceil$. The matrix $\mathbf{I}_d$ represents the $d \times d$ identity; $\mathbf{1}_d$ and $\mathbf{0}_d$ are the d-dimensional column vectors of all ones and zeros, respectively. We denote $[\mathbf{x}]_i$ as the i-th entry of a vector $\mathbf{x}$. The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. We use $\|\cdot\|$ to denote the Euclidean norm of a vector or the spectral norm of a matrix. We use $\sigma(\cdot)$ to denote the σ-algebra generated by the sets and/or random vectors in its argument.

2. Problem Setup and GT-HSGD

In this section, we introduce the mathematical model of the stochastic first-order oracle (SFO) at each node and the communication network. Based on these formulations, we develop the proposed GT-HSGD algorithm.

2.1. Optimization and Network Model

We work with a rich enough probability space $\{\Omega, \mathbb{P}, \mathcal{F}\}$. We consider decentralized recursive algorithms of interest that generate a sequence of estimates $\{\mathbf{x}_t^i\}_{t\geq 0}$ of the first-order stationary points of F at each node i, where $\mathbf{x}_0^i$ is assumed constant. At each iteration t, each node i observes a random vector $\boldsymbol{\xi}_t^i$ in $\mathbb{R}^q$, which, for instance, may be considered as noise or as an online data sample. We then introduce the natural filtration (an increasing family of sub-

σ-algebras of $\mathcal{F}$) induced by these random vectors observed sequentially by the networked nodes:

$$\mathcal{F}_0 := \{\Omega, \phi\},$$
$$\mathcal{F}_t := \sigma\left(\left\{\boldsymbol{\xi}_0^i, \boldsymbol{\xi}_1^i, \cdots, \boldsymbol{\xi}_{t-1}^i : i \in \mathcal{V}\right\}\right), \qquad \forall t \geq 1, \quad (1)$$

where ϕ is the empty set. We are now ready to define the SFO mechanism in the following. At each iteration t, each node i, given an input random vector $\mathbf{x} \in \mathbb{R}^p$ that is $\mathcal{F}_t$-measurable, is able to query the local SFO to obtain a stochastic gradient of the form $\mathbf{g}_i(\mathbf{x}, \boldsymbol{\xi}_t^i)$, where $\mathbf{g}_i : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p$ is a Borel measurable function. We assume that the SFO satisfies the following four properties.

Assumption 1 (Oracle). For any $\mathcal{F}_t$-measurable random vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, we have the following: $\forall i \in \mathcal{V}, \forall t \geq 0$,

- $\mathbb{E}\left[\mathbf{g}_i(\mathbf{x}, \boldsymbol{\xi}_t^i)|\mathcal{F}_t\right] = \nabla f_i(\mathbf{x})$;

- $\mathbb{E}\left[\|\mathbf{g}_i(\mathbf{x}, \boldsymbol{\xi}_t^i) - \nabla f_i(\mathbf{x})\|^2\right] \leq \nu_i^2, \bar{\nu}^2 := \frac{1}{n}\sum_{i=1}^{n}\nu_i^2$;

- the family $\left\{\boldsymbol{\xi}_t^i : \forall t \geq 0, i \in \mathcal{V}\right\}$ of random vectors is independent;

- $\mathbb{E}\left[\|\mathbf{g}_i(\mathbf{x}, \boldsymbol{\xi}_t^i) - \mathbf{g}_i(\mathbf{y}, \boldsymbol{\xi}_t^i)\|^2\right] \leq L^2\mathbb{E}\left[\|\mathbf{x} - \mathbf{y}\|^2\right]$.

The first three properties above are standard and commonly used to establish the convergence of decentralized stochastic gradient methods. They however do not explicitly impose any structures on the stochastic gradient mapping $\mathbf{g}_i$ other

than the measurability. On the other hand, the last property, the mean-squared smoothness, roughly speaking, requires that $\mathbf{g}_i$ is L-smooth on average with respect to the input arguments $\mathbf{x}$ and $\mathbf{y}$. As a simple example, Assumption 1 holds if $f_i(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}_i \mathbf{x}$ and $\mathbf{g}_i(\mathbf{x}, \boldsymbol{\xi}_i) = \mathbf{Q}_i \mathbf{x} + \boldsymbol{\xi}_i$, where $\mathbf{Q}_i$ is a constant matrix and $\boldsymbol{\xi}_i$ has zero mean and finite second moment. We further note that the mean-squared smoothness of each $\mathbf{g}_i$ implies, by Jensen's inequality, that each f_i is L-smooth, i.e., $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, and consequently the global function F is also L-smooth.

In addition, we make the following assumptions on F and the communication network $\mathcal{G}$.

Assumption 2 (Global Function). F is bounded below, i.e., $F^* := \inf_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}) > -\infty$.

Assumption 3 (Communication Network). The directed network $\mathcal{G}$ admits a primitive and doubly-stochastic weight matrix $\underline{\mathbf{W}} = \{\underline{w}_{ij}\} \in \mathbb{R}^{n \times n}$. Hence, $\underline{\mathbf{W}} \mathbf{1}_n = \underline{\mathbf{W}}^\top \mathbf{1}_n = \mathbf{1}_n$ and $\lambda := \|\underline{\mathbf{W}} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\| \in [0, 1)$.

The weight matrix $\underline{\mathbf{W}}$ that satisfies Assumption 3 may be designed for strongly-connected weight-balanced directed graphs (and thus for arbitrary connected undirected graphs). For example, the family of directed exponential graphs is weight-balanced and plays a key role in decentralized training (Assran et al., 2019). We note that λ is known as the second largest singular value of $\underline{\mathbf{W}}$ and measures the algebraic connectivity of the graph, i.e., a smaller value of λ roughly means a better connectivity. We note that several existing approaches require strictly stronger assumptions on $\underline{\mathbf{W}}$. For instance, D2 (Tang et al., 2018) and D-PD-SGD (Yi et al., 2020) require $\underline{\mathbf{W}}$ to be symmetric and hence are restricted to undirected networks.

2.2. Algorithm Development and Description

We now describe the proposed GT-HSGD algorithm and provide an intuitive construction. Recall that $\mathbf{x}_t^i$ is the estimate of an stationary point of the global cost F at node i and iteration t. Let $\mathbf{g}_i(\mathbf{x}_t^i, \boldsymbol{\xi}_t^i)$ and $\mathbf{g}_i(\mathbf{x}_{t-1}^i, \boldsymbol{\xi}_t^i)$ be the corresponding stochastic gradients returned by the local SFO queried at $\mathbf{x}_t^i$ and $\mathbf{x}_{t-1}^i$ respectively. Motivated by the strong performance of recently introduced decentralized methods that combine gradient tracking and various variance reduction schemes for finite-sum problems (Xin et al., 2020b;c; Li et al., 2020a; Sun et al., 2020), we seek similar variance reduction for decentralized online problems with SFO. In particular, we focus on the following *local* hybrid variance reduced stochastic gradient estimator $\mathbf{v}_t^i$ introduced in (Liu et al., 2020; Tran-Dinh et al., 2020; Cutkosky & Orabona, 2019) for centralized online problems: $\forall t \geq 1$,

$$\mathbf{v}_t^i = \mathbf{g}_i(\mathbf{x}_t^i, \boldsymbol{\xi}_t^i) + (1 - \beta)\left(\mathbf{v}_{t-1}^i - \mathbf{g}_i(\mathbf{x}_{t-1}^i, \boldsymbol{\xi}_t^i)\right), \quad (2)$$

for some applicable weight parameter $\beta \in [0, 1]$. This local gradient estimator $\mathbf{v}_t^i$ is fused, via a gradient tracking

mechanism (Di Lorenzo & Scutari, 2016; Xu et al., 2015), over the network to update the global gradient tracker $\mathbf{y}_t^i$, which is subsequently used as the descent direction in the $\mathbf{x}_t^i$-update. The complete description of GT-HSGD is provided in Algorithm 1. We note that the update (2) of $\mathbf{v}_t^i$ may be equivalently written as

$$\mathbf{v}_t^i = \beta \cdot \underbrace{\mathbf{g}_i(\mathbf{x}_t^i, \boldsymbol{\xi}_t^i)}_{\texttt{Stochastic gradient}}$$
$$+ (1 - \beta) \cdot \underbrace{\left(\mathbf{g}_i(\mathbf{x}_t^i, \boldsymbol{\xi}_t^i) - \mathbf{g}_i(\mathbf{x}_{t-1}^i, \boldsymbol{\xi}_t^i) + \mathbf{v}_{t-1}^i\right)}_{\texttt{SARAH}},$$

which is a convex combination of the local vanilla stochastic gradient returned by the SFO and a SARAH-type (Nguyen et al., 2017; Fang et al., 2018; Wang et al., 2019) gradient estimator. This discussion leads to the fact that GT-HSGD reduces to GT-DSGD (Pu & Nedich, 2020; Xin et al., 2020e; Lu et al., 2019) when $\beta = 1$, and becomes the inner loop of GT-SARAH (Xin et al., 2020b) when $\beta = 0$. However, our convergence analysis shows that GT-HSGD achieves its best oracle complexity and outperforms the existing decentralized online variance-reduced approaches (Sun et al., 2020; Pan et al., 2020) with a weight parameter $\beta \in (0, 1)$. It is then clear that neither GT-DSGD nor the inner loop of GT-SARAH, on their own, are able to outperform the proposed approach, making GT-HSGD a non-trivial algorithmic design for this problem class.

Algorithm 1 GT-HSGD at each node i

Require: $\mathbf{x}_0^i = \overline{\mathbf{x}}_0$; α; β; b_0; $\mathbf{y}_0^i = \mathbf{0}_p$; $\mathbf{v}_{-1}^i = \mathbf{0}_p$; T.

1: Sample $\{\boldsymbol{\xi}_{0,r}^i\}_{r=1}^{b_0}$ and $\mathbf{v}_0^i = \frac{1}{b_0}\sum_{r=1}^{b_0} \mathbf{g}_i(\mathbf{x}_0^i, \boldsymbol{\xi}_{0,r}^i)$;

2: $\mathbf{y}_1^i = \sum_{j=1}^n \underline{w}_{ij}(\mathbf{y}_0^j + \mathbf{v}_0^j - \mathbf{v}_{-1}^j)$;

3: $\mathbf{x}_1^i = \sum_{j=1}^n \underline{w}_{ij}(\mathbf{x}_0^j - \alpha \mathbf{y}_1^j)$;

4: **for** $t = 1, 2, \cdots, T-1$ **do**

5:     Sample $\boldsymbol{\xi}_t^i$;

6:     $\mathbf{v}_t^i = \mathbf{g}_i(\mathbf{x}_t^i, \boldsymbol{\xi}_t^i) + (1 - \beta)(\mathbf{v}_{t-1}^i - \mathbf{g}_i(\mathbf{x}_{t-1}^i, \boldsymbol{\xi}_t^i))$.

7:     $\mathbf{y}_{t+1}^i = \sum_{j=1}^n \underline{w}_{ij}(\mathbf{y}_t^j + \mathbf{v}_t^j - \mathbf{v}_{t-1}^j)$;

8:     $\mathbf{x}_{t+1}^i = \sum_{j=1}^n \underline{w}_{ij}(\mathbf{x}_t^j - \alpha \mathbf{y}_{t+1}^j)$;

9: **end for**

output $\widetilde{\mathbf{x}}_T$ selected uniformly at random from $\{\mathbf{x}_t^i\}_{0 \leq t \leq T}^{i \in \mathcal{V}}$.

Remark 1. *Clearly, each $\mathbf{v}_t^i$ is a conditionally biased estimator of $\nabla f_i(\mathbf{x}_t^i)$, i.e., $\mathbb{E}[\mathbf{v}_t^i|\mathcal{F}_t] \neq \nabla f_i(\mathbf{x}_t^i)$ in general. However, it can be shown that $\mathbb{E}[\mathbf{v}_t^i] = \mathbb{E}[\nabla f_i(\mathbf{x}_t^i)]$, meaning that $\mathbf{v}_t^i$ serves as a surrogate for the underlying exact gradient in the sense of total expectation.*

3. Main Results

In this section, we present the main convergence results of GT-HSGD in this paper and discuss their salient features.

The formal convergence analysis is deferred to Section 4.

Theorem 1. *If the weight parameter $\beta = \frac{48L^2\alpha^2}{n}$ and the step-size α is chosen as*

$$0 < \alpha < \min\left\{\frac{(1-\lambda^2)^2}{90\lambda^2}, \frac{\sqrt{n(1-\lambda)}}{26\lambda}, \frac{1}{4\sqrt{3}}\right\}\frac{1}{L},$$

then the output $\widetilde{\mathbf{x}}_T$ of GT-HSGD satisfies: $\forall T \geq 2$,

$$\mathbb{E}\left[\|\nabla F(\widetilde{\mathbf{x}}_T)\|^2\right] \leq \frac{4(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha T} + \frac{8\beta\overline{\nu}^2}{n} + \frac{4\overline{\nu}^2}{\beta b_0 nT}$$
$$+ \frac{64\lambda^4\|\nabla\mathbf{f}(\mathbf{x}_0)\|^2}{(1-\lambda^2)^3 nT} + \frac{96\lambda^2\overline{\nu}^2}{(1-\lambda^2)^3 b_0 T} + \frac{256\lambda^2\beta^2\overline{\nu}^2}{(1-\lambda^2)^3},$$

where $\|\nabla\mathbf{f}(\mathbf{x}_0)\|^2 = \sum_{i=1}^n \|\nabla f_i(\overline{\mathbf{x}}_0)\|^2$

Remark 2. *Theorem 1 holds for GT-HSGD with arbitrary initial minibatch size $b_0 \geq 1$.*

Theorem 1 establishes a non-asymptotic bound, with no hidden constants, on the mean-squared stationary gap of GT-HSGD over any finite time horizon T.

Transient and steady-state performance over infinite time horizon. If α and β are chosen according to Theorem 1, the mean-squared stationary gap $\mathbb{E}\left[\|\nabla F(\widetilde{\mathbf{x}}_T)\|^2\right]$ of GT-HSGD decays sublinearly at a rate of $O(1/T)$ up to a steady-state error (SSE) such that

$$\limsup_{T\to\infty} \mathbb{E}\left[\|\nabla F(\widetilde{\mathbf{x}}_T)\|^2\right] \leq \frac{8\beta\overline{\nu}^2}{n} + \frac{256\lambda^2\beta^2\overline{\nu}^2}{(1-\lambda^2)^3}. \quad (3)$$

In view of (3), the SSE of GT-HSGD is bounded by the sum of two terms: (i) the first term is in the order of $O(\beta)$ and the division by n demonstrates the benefit of increasing the network size[2]; (ii) the second term is in the order of $O(\beta^2)$ and reveals the impact of the spectral gap $(1 - \lambda)$ of the network topology. Clearly, the SSE can be made arbitrarily small by choosing small enough β and α. Moreover, since the spectral gap $(1 - \lambda)$ only appears in a higher order term of β in (3), its impact reduces as β becomes smaller, i.e., as we require a smaller SSE.

The following corollary is concerned with the finite-time convergence rate of GT-HSGD with specific choices of the algorithmic parameters α, β, and b_0.

Corollary 1. *Setting $\alpha = \frac{n^{2/3}}{8LT^{1/3}}$, $\beta = \frac{3n^{1/3}}{4T^{2/3}}$, and $b_0 = \lceil\frac{T^{1/3}}{n^{2/3}}\rceil$ in Theorem 1, we have:*

$$\mathbb{E}\left[\|\nabla F(\widetilde{\mathbf{x}}_T)\|^2\right] \leq \frac{32L(F(\overline{\mathbf{x}}_0) - F^*) + 12\overline{\nu}^2}{(nT)^{2/3}}$$
$$+ \frac{64\lambda^4\|\nabla\mathbf{f}(\mathbf{x}_0)\|^2}{(1-\lambda^2)^3 nT} + \frac{240\lambda^2 n^{2/3}\overline{\nu}^2}{(1-\lambda^2)^3 T^{4/3}},$$

for all $T > \max\left\{\frac{1424\lambda^6 n^2}{(1-\lambda^2)^6}, \frac{35\lambda^3 n^{0.5}}{(1-\lambda)^{1.5}}\right\}$. As a consequence, GT-HSGD achieves an ϵ-accurate stationary point $\mathbf{x}^*$ of the global cost F such that $\mathbb{E}[\|\nabla F(\mathbf{x}^*)\|] \leq \epsilon$ with

$$\mathcal{H} = O\left(\max\{\mathcal{H}_{opt}, \mathcal{H}_{net}\}\right)$$

iterations[3], where $\mathcal{H}_{opt}$ and $\mathcal{H}_{net}$ are given respectively by

$$\mathcal{H}_{opt} = \frac{(L(F(\overline{\mathbf{x}}_0) - F^*) + \overline{\nu}^2)^{1.5}}{n\epsilon^3},$$

$$\mathcal{H}_{net} = \max\left\{\frac{\lambda^4\|\nabla\mathbf{f}(\mathbf{x}_0)\|^2}{(1-\lambda^2)^3 n\epsilon^2}, \frac{\lambda^{1.5} n^{0.5}\overline{\nu}^{1.5}}{(1-\lambda^2)^{2.25}\epsilon^{1.5}}\right\}.$$

The resulting total number of oracle queries at each node is thus $\lceil\mathcal{H} + \mathcal{H}^{1/3}n^{-2/3}\rceil$.

Remark 3. *Since $\mathcal{H}^{1/3}n^{-2/3}$ is much smaller than $\mathcal{H}$, we treat the oracle complexity of GT-HSGD as $\mathcal{H}$ for the ease of exposition in Table 1 and the following discussion.*

An important implication of Corollary 1 is given in the following.

A regime for network topology-independent oracle complexity and linear speedup. According to Corollary 1, the oracle complexity of GT-HSGD at each node is bounded by the maximum of two terms: (i) the first term $\mathcal{H}_{opt}$ is independent of the network topology and, more importantly, is n times smaller than the oracle complexity of the optimal centralized online variance-reduced methods that execute on a single node for this problem class (Fang et al., 2018; Wang et al., 2019; Pham et al., 2020; Tran-Dinh et al., 2020; Liu et al., 2020); (ii) the second term $\mathcal{H}_{net}$ depends on the network spectral gap $1 - \lambda$ and is in the lower order of $1/\epsilon$. These two observations lead to the interesting fact that the oracle complexity of GT-HSGD becomes independent of the network topology, i.e., $\mathcal{H}_{opt}$ dominates $\mathcal{H}_{net}$, if the required error tolerance ϵ is small enough such that[4] $\epsilon \lesssim \min\{\lambda^{-4}(1-\lambda)^3 n^{-1}, \lambda^{-1}(1-\lambda)^{1.5} n^{-1}\}$. In this regime, GT-HSGD thus achieves a network topology-independent oracle complexity $\mathcal{H}_{opt} = O(n^{-1}\epsilon^{-3})$, exhibiting a linear speed up compared with the aforementioned centralized optimal algorithms (Fang et al., 2018; Wang et al., 2019; Pham et al., 2020; Tran-Dinh et al., 2020; Liu et al., 2020; Zhou et al., 2020), in the sense that the total number of oracle queries required to achieve an ϵ-accurate stationary point at each node is reduced by a factor of $1/n$.

Remark 4. *The small error tolerance regime in the above discussion corresponds to a large number of oracle queries, which translates to the scenario where the required total number of iterations T is large. Note that a large T further implies that the step-size α and the weight parameter β are small; see the expression of α and β in Corollary 1.*

---

[2]Since GT-HSGD computes $O(n)$ stochastic gradients in parallel per iteration across the nodes, the network size n can be interpreted as the minibatch size of GT-HSGD.

[3]The $O(\cdot)$ notation here does not absorb any problem parameters, i.e., it only hides universal constants.

[4]This boundary condition follows from basic algebraic manipulations.

4. Outline of the Convergence Analysis

In this section, we outlines the proof of Theorem 1, while the detailed proofs are provided in the Appendix. We let Assumptions 1-3 hold throughout the rest of the paper without explicitly stating them. For the ease of exposition, we write the $\mathbf{x}_t$- and $\mathbf{y}_t$-update of GT-HSGD in the following equivalent matrix form: $\forall t \geq 0$,

$$\mathbf{y}_{t+1} = \mathbf{W}(\mathbf{y}_t + \mathbf{v}_t - \mathbf{v}_{t-1}), \quad (4a)$$

$$\mathbf{x}_{t+1} = \mathbf{W}(\mathbf{x}_t - \alpha \mathbf{y}_{t+1}), \quad (4b)$$

where $\mathbf{W} := \underline{\mathbf{W}} \otimes \mathbf{I}_p$ and $\mathbf{x}_t, \mathbf{y}_t, \mathbf{v}_t$ are square-integrable random vectors in $\mathbb{R}^{np}$ that respectively concatenate the local estimates $\{\mathbf{x}_t^i\}_{i=1}^n$ of a stationary point of F, gradient trackers $\{\mathbf{y}_t^i\}_{i=1}^n$, stochastic gradient estimators $\{\mathbf{v}_t^i\}_{i=1}^n$. It is straightforward to verify that $\mathbf{x}_t$ and $\mathbf{y}_t$ are $\mathcal{F}_t$-measurable while $\mathbf{v}_t$ is $\mathcal{F}_{t+1}$-measurable for all $t \geq 0$. For convenience, we also denote

$$\nabla \mathbf{f}(\mathbf{x}_t) := \left[\nabla f_1(\mathbf{x}_t^1)^\top, \cdots, \nabla f_n(\mathbf{x}_t^n)^\top\right]^\top$$

and introduce the following quantities:

$$\mathbf{J} := \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\right) \otimes \mathbf{I}_p$$

$$\overline{\mathbf{x}}_t := \frac{1}{n}(\mathbf{1}_n^\top \otimes \mathbf{I}_p)\mathbf{x}_t, \quad \overline{\mathbf{y}}_t := \frac{1}{n}(\mathbf{1}_n^\top \otimes \mathbf{I}_p)\mathbf{y}_t,$$

$$\overline{\mathbf{v}}_t := \frac{1}{n}(\mathbf{1}_n^\top \otimes \mathbf{I}_p)\mathbf{v}_t, \quad \overline{\nabla \mathbf{f}}(\mathbf{x}_t) := \frac{1}{n}(\mathbf{1}_n^\top \otimes \mathbf{I}_p)\nabla \mathbf{f}(\mathbf{x}_t).$$

In the following lemma, we enlist several well-known results in the context of gradient tracking-based algorithms for decentralized stochastic optimization, whose proofs may be found in (Di Lorenzo & Scutari, 2016; Qu & Li, 2017; Xin et al., 2020d; Pu & Nedich, 2020).

Lemma 1. *The following relationships hold.*

(a) $\|\mathbf{W}\mathbf{x} - \mathbf{J}\mathbf{x}\| \leq \lambda \|\mathbf{x} - \mathbf{J}\mathbf{x}\|, \forall \mathbf{x} \in \mathbb{R}^{np}$.

(b) $\overline{\mathbf{y}}_{t+1} = \overline{\mathbf{v}}_t, \forall t \geq 0$.

(c) $\left\|\overline{\nabla \mathbf{f}}(\mathbf{x}_t) - \nabla F(\overline{\mathbf{x}}_t)\right\|^2 \leq \frac{L^2}{n}\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2, \forall t \geq 0$.

We note that Lemma 1(a) holds since $\underline{\mathbf{W}}$ is primitive and doubly-stochastic, Lemma 1(b) is a direct consequence of the gradient tracking update (4a) and Lemma 1(c) is due to the L-smoothness of each f_i. By the estimate update of GT-HSGD described in (4b) and Lemma 1(b), it is straightforward to obtain:

$$\overline{\mathbf{x}}_{t+1} = \overline{\mathbf{x}}_t - \alpha \overline{\mathbf{y}}_{t+1} = \overline{\mathbf{x}}_t - \alpha \overline{\mathbf{v}}_t, \qquad \forall t \geq 0. \quad (5)$$

Hence, the mean state $\overline{\mathbf{x}}_t$ proceeds in the direction of the average of local stochastic gradient estimators $\overline{\mathbf{v}}_t$. With the help of (5) and the L-smoothness of F and each f_i, we establish the following descent inequality which sheds light on the overall convergence analysis.

Lemma 2. *If* $0 < \alpha \leq \frac{1}{2L}$, *then we have:* $\forall T \geq 0$,

$$\sum_{t=0}^{T}\left\|\nabla F(\overline{\mathbf{x}}_t)\right\|^2 \leq \frac{2(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha} - \frac{1}{2}\sum_{t=0}^{T}\|\overline{\mathbf{v}}_t\|^2$$

$$+ 2\sum_{t=0}^{T}\left\|\overline{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\right\|^2 + \frac{2L^2}{n}\sum_{t=0}^{T}\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2.$$

In light of Lemma 2, our approach to establishing the convergence of GT-HSGD is to seek the conditions on the algorithmic parameters of GT-HSGD, i.e., the step-size α and the weight parameter β, such that

$$-\frac{1}{2T}\sum_{t=0}^{T}\mathbb{E}\left[\|\overline{\mathbf{v}}_t\|^2\right] + \frac{2}{T}\sum_{t=0}^{T}\mathbb{E}\left[\|\overline{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\|^2\right]$$

$$+ \frac{2L^2}{nT}\sum_{t=0}^{T}\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right] = O\left(\alpha, \beta, \frac{1}{b_0}, \frac{1}{T}\right), \quad (6)$$

where $O(\alpha, \beta, 1/b_0, 1/T)$ represents a nonnegative quantity which may be made arbitrarily small by choosing small enough α and β along with large enough T and b_0. If (6) holds, then Lemma 2 reduces to

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}\left[\|\nabla F(\overline{\mathbf{x}}_t)\|^2\right]$$

$$\leq \frac{2(F(\overline{\mathbf{x}}_0) - F^*)}{\alpha T} + O\left(\alpha, \beta, \frac{1}{b_0}, \frac{1}{T}\right),$$

which leads to the convergence arguments of GT-HSGD. For these purposes, we quantify $\sum_{t=0}^{T}\mathbb{E}\left[\|\overline{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\|^2\right]$ and $\sum_{t=0}^{T}\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right]$ next.

4.1. Contraction Relationships

First of all, we establish upper bounds on the gradient variances $\mathbb{E}\left[\|\overline{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\|^2\right]$ and $\mathbb{E}\left[\|\mathbf{v}_t - \nabla \mathbf{f}(\mathbf{x}_t)\|^2\right]$ by exploiting the hybrid and recursive update of $\mathbf{v}_t$.

Lemma 3. *The following inequalities hold:* $\forall t \geq 1$,

$$\mathbb{E}\left[\|\overline{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\|^2\right]$$
$$\leq (1-\beta)^2 \mathbb{E}\left[\|\overline{\mathbf{v}}_{t-1} - \overline{\nabla \mathbf{f}}(\mathbf{x}_{t-1})\|^2\right]$$
$$+ \frac{6L^2\alpha^2}{n}(1-\beta)^2\mathbb{E}\left[\|\overline{\mathbf{v}}_{t-1}\|^2\right] + \frac{2\beta^2\overline{\nu}^2}{n}$$
$$+ \frac{6L^2}{n^2}(1-\beta)^2\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2 + \|\mathbf{x}_{t-1} - \mathbf{J}\mathbf{x}_{t-1}\|^2\right],$$
$$\quad (7)$$

and, $\forall t \geq 1$,

$$\mathbb{E}\left[\|\mathbf{v}_t - \nabla \mathbf{f}(\mathbf{x}_t)\|^2\right]$$
$$\leq (1-\beta)^2 \mathbb{E}\left[\|\mathbf{v}_{t-1} - \nabla \mathbf{f}(\mathbf{x}_{t-1})\|^2\right]$$
$$+ 6nL^2\alpha^2(1-\beta)^2\mathbb{E}\left[\|\overline{\mathbf{v}}_{t-1}\|^2\right] + 2n\beta^2\overline{\nu}^2$$
$$+ 6L^2(1-\beta)^2\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2 + \|\mathbf{x}_{t-1} - \mathbf{J}\mathbf{x}_{t-1}\|^2\right].$$
$$\quad (8)$$

Remark 5. *Since* $\mathbf{v}_t$ *is a conditionally biased estimator of* $\nabla \mathbf{f}(\mathbf{x}_t)$, (7) *and* (8) *do not directly imply each other and need to be established separately.*

We emphasize that the contraction structure of the gradient variances shown in Lemma 3 plays a crucial role in the convergence analysis. The following contraction bounds on the consensus errors $\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right]$ are standard in decentralized algorithms based on gradient tracking, e.g., (Pu & Nedich, 2020; Xin et al., 2020b); in particular, it follows directly from the $\mathbf{x}_t$-update (4b) and Young's inequality.

Lemma 4. *The following inequalities hold:* $\forall t \geq 0$,

$$\|\mathbf{x}_{t+1} - \mathbf{J}\mathbf{x}_{t+1}\|^2 \leq \frac{1+\lambda^2}{2}\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2$$
$$+ \frac{2\alpha^2\lambda^2}{1-\lambda^2}\|\mathbf{y}_{t+1} - \mathbf{J}\mathbf{y}_{t+1}\|^2. \quad (9)$$

$$\|\mathbf{x}_{t+1} - \mathbf{J}\mathbf{x}_{t+1}\|^2 \leq 2\lambda^2\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2$$
$$+ 2\alpha^2\lambda^2\|\mathbf{y}_{t+1} - \mathbf{J}\mathbf{y}_{t+1}\|^2. \quad (10)$$

It is then clear from Lemma 4 that we need to further quantify the gradient tracking errors $\mathbb{E}\left[\|\mathbf{y}_t - \mathbf{J}\mathbf{y}_t\|^2\right]$ in order to bound the consensus errors. These error bounds are shown in the following lemma.

Lemma 5. *We have the following.*

(a) $\mathbb{E}\left[\|\mathbf{y}_1 - \mathbf{J}\mathbf{y}_1\|^2\right] \leq \lambda^2 \left\|\nabla \mathbf{f}(\mathbf{x}_0)\right\|^2 + \lambda^2 n \bar{\nu}^2 / b_0.$

(b) *If* $0 < \alpha \leq \frac{1-\lambda^2}{2\sqrt{42}\lambda^2 L}$, *then* $\forall t \geq 1$,

$$\mathbb{E}\left[\|\mathbf{y}_{t+1} - \mathbf{J}\mathbf{y}_{t+1}\|^2\right]$$
$$\leq \frac{3+\lambda^2}{4}\mathbb{E}\left[\|\mathbf{y}_t - \mathbf{J}\mathbf{y}_t\|^2\right] + \frac{21\lambda^2 nL^2\alpha^2}{1-\lambda^2}\mathbb{E}[\|\bar{\mathbf{v}}_{t-1}\|^2]$$
$$+ \frac{63\lambda^2 L^2}{1-\lambda^2}\mathbb{E}\left[\|\mathbf{x}_{t-1} - \mathbf{J}\mathbf{x}_{t-1}\|^2\right]$$
$$+ \frac{7\lambda^2\beta^2}{1-\lambda^2}\mathbb{E}\left[\|\mathbf{v}_{t-1} - \nabla \mathbf{f}(\mathbf{x}_{t-1})\|^2\right] + 3\lambda^2 n\beta^2\bar{\nu}^2.$$

We note that establishing the contraction argument of gradient tracking errors in Lemma 5 requires a careful examination of the structure of the $\mathbf{v}_t$-update.

4.2. Error Accumulations

To proceed, we observe, from Lemma 3, 4, and 5, that the recursions of the gradient variances, consensus, and gradient tracking errors admit similar forms. Therefore, we abstract out formulas for the accumulation of the error recursions of this type in the following lemma.

Lemma 6. *Let* $\{V_t\}_{t\geq 0}$, $\{R_t\}_{t\geq 0}$ *and* $\{Q_t\}_{t\geq 0}$ *be non-negative sequences and* $C \geq 0$ *be some constant such*

that $V_t \leq qV_{t-1} + qR_{t-1} + Q_t + C$, $\forall t \geq 1$, *where* $q \in (0,1)$. *Then the following inequality holds:* $\forall T \geq 1$,

$$\sum_{t=0}^{T} V_t \leq \frac{V_0}{1-q} + \frac{1}{1-q}\sum_{t=0}^{T-1} R_t + \frac{1}{1-q}\sum_{t=1}^{T} Q_t + \frac{CT}{1-q}.$$
(11)

Similarly, if $V_{t+1} \leq qV_t + R_{t-1} + C$, $\forall t \geq 1$, *then we have:* $\forall T \geq 2$,

$$\sum_{t=1}^{T} V_t \leq \frac{V_1}{1-q} + \frac{1}{1-q}\sum_{t=0}^{T-2} R_t + \frac{CT}{1-q}.$$
(12)

Applying Lemma 6 to Lemma 3 leads to the following upper bounds on the accumulated variances.

Lemma 7. *For any* $\beta \in (0,1)$, *the following inequalities hold:* $\forall T \geq 1$,

$$\sum_{t=0}^{T} \mathbb{E}\left[\|\bar{\mathbf{v}}_t - \overline{\nabla \mathbf{f}}(\mathbf{x}_t)\|^2\right]$$
$$\leq \frac{\bar{\nu}^2}{\beta b_0 n} + \frac{6L^2\alpha^2}{n\beta}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\bar{\mathbf{v}}_t\|^2\right]$$
$$+ \frac{12L^2}{n^2\beta}\sum_{t=0}^{T} \mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right] + \frac{2\beta\bar{\nu}^2 T}{n}, \quad (13)$$

and, $\forall T \geq 1$,

$$\sum_{t=0}^{T} \mathbb{E}\left[\|\mathbf{v}_t - \nabla \mathbf{f}(\mathbf{x}_t)\|^2\right]$$
$$\leq \frac{n\bar{\nu}^2}{\beta b_0} + \frac{6nL^2\alpha^2}{\beta}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\bar{\mathbf{v}}_t\|^2\right]$$
$$+ \frac{12L^2}{\beta}\sum_{t=0}^{T} \mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right] + 2n\beta\bar{\nu}^2 T. \quad (14)$$

It can be observed that (13) in Lemma 7 may be used to refine the left hand side of (6). The remaining step, naturally, is to bound $\sum_t \mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right]$ in terms of $\sum_t \mathbb{E}\left[\|\bar{\mathbf{v}}_t\|^2\right]$. This result is provided in the following lemma that is obtained with the help of Lemma 4, 5, 6, and 7.

Lemma 8. *If* $0 < \alpha \leq \frac{(1-\lambda^2)^2}{70\lambda^2 L}$ *and* $\beta \in (0,1)$, *then the following inequality holds:* $\forall T \geq 2$,

$$\sum_{t=0}^{T} \frac{\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{J}\mathbf{x}_t\|^2\right]}{n} \leq \frac{2016\lambda^4 L^2\alpha^4}{(1-\lambda^2)^4}\sum_{t=0}^{T-2} \mathbb{E}\left[\|\bar{\mathbf{v}}_t\|^2\right]$$
$$+ \frac{32\lambda^4\alpha^2}{(1-\lambda^2)^3}\frac{\|\nabla \mathbf{f}(\mathbf{x}_0)\|^2}{n} + \left(\frac{7\beta}{1-\lambda^2} + 1\right)\frac{32\lambda^4\bar{\nu}^2\alpha^2}{(1-\lambda^2)^3 b_0}$$
$$+ \left(\frac{14\beta}{1-\lambda^2} + 3\right)\frac{32\lambda^4\beta^2\bar{\nu}^2\alpha^2 T}{(1-\lambda^2)^3}.$$

Finally, we note that Lemma 7 and 8 suffice to establish (6) and hence lead to Theorem 1; see the Appendix for details.
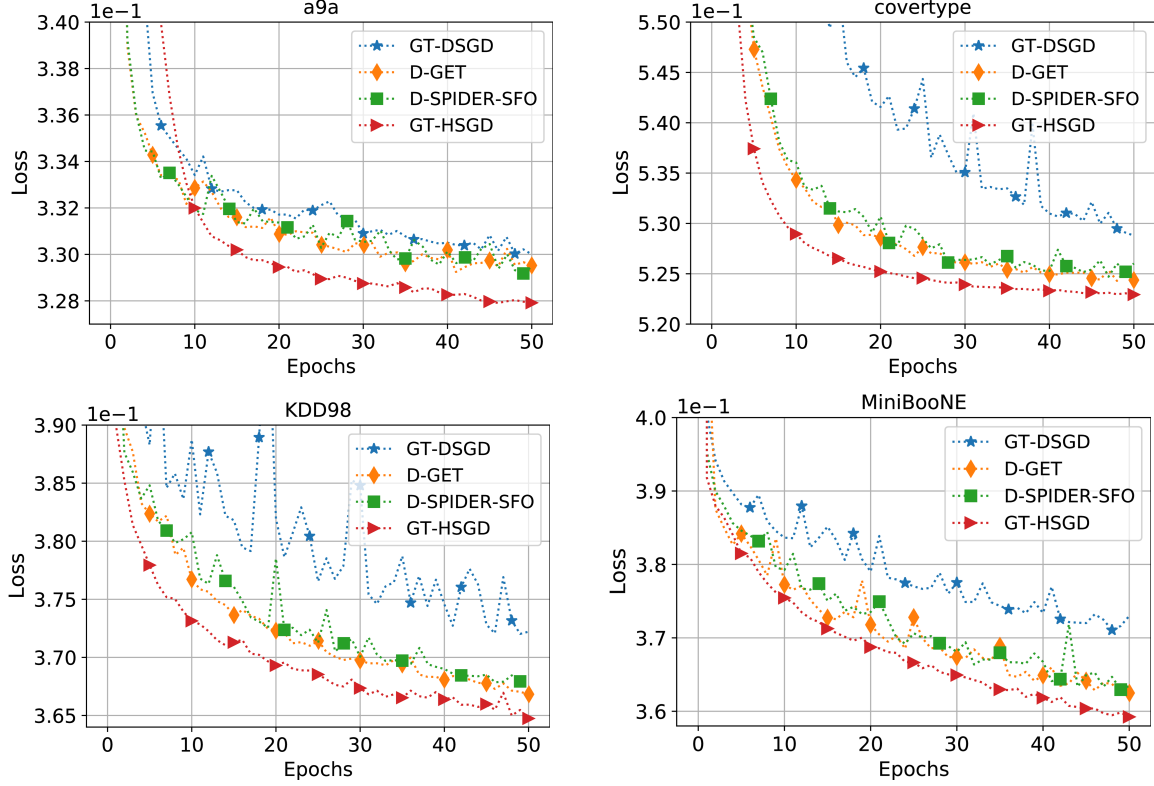
Figure 1. A comparison of `GT-HSGD` with other decentralized online stochastic gradient algorithms over the undirected exponential graph of 20 nodes on the a9a, covertype, KDD98, and MiniBooNE datasets.

5. Numerical Experiments

In this section, we illustrate our theoretical results on the convergence of the proposed `GT-HSGD` algorithm with the help of numerical experiments.

Model. We consider a non-convex logistic regression model (Antoniadis et al., 2011) for decentralized binary classification. In particular, the decentralized non-convex optimization problem of interest takes the form $\min_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}) + r(\mathbf{x})$, such that

$$f_i(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} \log \left[1 + e^{-\langle \mathbf{x}, \boldsymbol{\theta}_{ij} \rangle l_{ij}} \right]$$

and

$$r(\mathbf{x}) = R \sum_{k=1}^{p} \frac{[\mathbf{x}]_k^2}{1 + [\mathbf{x}]_k^2},$$

where $\boldsymbol{\theta}_{i,j}$ is the feature vector, $l_{i,j} \in \{-1, +1\}$ is the corresponding binary label, and $r(\mathbf{x})$ is a non-convex regularizer. To simulate the online `SFO` setting described in Section 2, each node i is only able to *sample with replacement* from its local data $\{\boldsymbol{\theta}_{i,j}, l_{i,j}\}_{j=1}^{m}$ and compute the corresponding (minibatch) stochastic gradient. Throughout all experiments, we set the number of the nodes to $n = 20$ and the regularization parameter to $R = 10^{-4}$.

Data. To test the performance of the applicable decentralized algorithms, we distribute the a9a, covertype, KDD98, MiniBooNE datasets uniformly over the nodes and normalize the feature vectors such that $\|\boldsymbol{\theta}_{i,j}\| = 1, \forall i, j$. The statistics of these datasets are provided in Table 2.

Table 2. Datasets used in numerical experiments, all available at https://www.openml.org/.

Dataset	train (nm)	dimension (p)
a9a	48,840	123
covertype	100,000	54
KDD98	75,000	477
MiniBooNE	100,000	11

Network topology. We consider the following network topologies: the undirected ring graph, the undirected and directed exponential graphs, and the complete graph; see (Nedić et al., 2018; Xin et al., 2020f; Assran et al., 2019; Lian et al., 2017) for detailed configurations of these graphs. For all graphs, the associated doubly stochastic weights are set to be equal. The resulting second largest singular value λ of the weight matrices are $0.98, 0.75, 0.67, 0$, respectively, demonstrating a significant difference in the algebraic connectivity of these graphs.

Performance measure. We measure the performance of the decentralized algorithms in question by the decrease of the global cost function value $F(\overline{\mathbf{x}})$, to which we refer as loss, versus epochs, where $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ with $\mathbf{x}_i$ being the model at node i and each epoch contains m stochastic gradient computations at each node.

5.1. Comparison with the Existing Decentralized Stochastic Gradient Methods.

We conduct a performance comparison of GT-HSGD with GT-DSGD (Pu & Nedich, 2020; Lu et al., 2019; Xin et al., 2020e), D-GET (Sun et al., 2020), and D-SPIDER-SFO (Pan et al., 2020) over the undirected exponential graph of 20 nodes. Note that we use GT-DSGD to represent methods that do not incorporate online variance reduction techniques, since it in general matches or outperforms DSGD (Lian et al., 2017) and has a similar performance with D2 (Tang et al., 2018) and D-PD-SGD (Yi et al., 2020).

Parameter tuning. We set the parameters of GT-HSGD, GT-DSGD, D-GET, and D-SPIDER-SFO according to the following procedures. *First*, we find a very large step-size candidate set for each algorithm in comparison. *Second*, we choose the minibatch size candidate set for all algorithms as $\mathcal{B} := \{1, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$: the minibatch size of GT-DSGD, the minibatch size of GT-HSGD at $t = 0$, the minibatch size of D-GET and D-SPIDER-SFO at inner- and outer-loop are all chosen from $\mathcal{B}$. *Third*, for D-GET and D-SPIDER-SFO, we choose the inner-loop length candidate set as $\left\{\frac{m}{20b}, \frac{m}{19b}, \cdots, \frac{m}{b}, \frac{2m}{b}, \cdots, \frac{20m}{b}\right\}$, where m is the local data size and b is the minibatch size at the inner-loop. *Fourth*, we iterate over all combinations of parameters for each algorithm to find its best performance. In particular, we find that the best performance of GT-HSGD is attained with a small β and a relatively large α as Corollary 1 suggests.

The experimental results are provided in Fig. 1, where we observe that GT-HSGD achieves faster convergence than the other algorithms in comparison on those four datasets. This observation is coherent with our main convergence results that GT-HSGD achieves a lower oracle complexity than the existing approaches; see Table 1.

5.2. Topology-Independent Rate of GT-HSGD

We test the performance of GT-HSGD over different network topologies. In particular, we follow the procedures described in Section 5.1 to find the best set of parameters for GT-HSGD over the *complete graph* and then use this parameter set for other graphs. The corresponding experimental results are presented in Fig. 2. Clearly, it can be observed that when the number of iterations is large enough, that is to say, the required error tolerance is small enough, the convergence rate of GT-HSGD is not affected by the
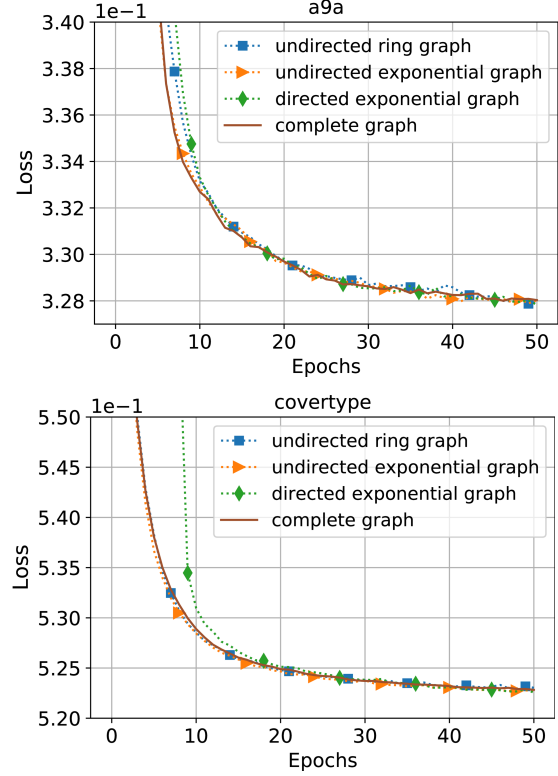

Figure 2. Convergence behaviors of GT-HSGD over different network topologies on the a9a and covertype datasets.

underlying network topology. This interesting phenomenon is consistent with our convergence theory; see Corollary 1 and the related discussion in Section 3.

6. Conclusion

In this paper, we investigate decentralized stochastic optimization to minimize a sum of smooth non-convex cost functions over a network of nodes. Assuming that each node has access to a stochastic first-order oracle, we propose GT-HSGD, a novel single-loop decentralized algorithm that leverages local hybrid variance reduction and gradient tracking to achieve provably fast convergence and robust performance. Compared with the existing online variance-reduced methods, GT-HSGD achieves a lower oracle complexity with a more practical implementation. We further show that GT-HSGD achieves a network topology-independent oracle complexity, when the required error tolerance is small enough, leading to a linear speedup with respect to the centralized optimal methods that execute on a single node.

Acknowledgments

References

Alghunaim, S. A., Ryu, E., Yuan, K., and Sayed, A. H. Decentralized proximal gradient algorithms with linear convergence rates. *IEEE Trans. Autom. Control*, 2020.

Antoniadis, A., Gijbels, I., and Nikolova, M. Penalized likelihood regression for generalized linear models with non-quadratic penalties. *Annals of the Institute of Statistical Mathematics*, 63(3):585–615, 2011.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. Stochastic gradient push for distributed deep learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 97: 344–353, 2019.

Chen, J. and Sayed, A. H. On the learning behavior of adaptive networks—part i: Transient analysis. *IEEE Transactions on Information Theory*, 61(6):3487–3517, 2015.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. In *Adv. Neural Inf. Process. Syst.*, pp. 15236–15245, 2019.

Di Lorenzo, P. and Scutari, G. NEXT: In-network nonconvex optimization. *IEEE Trans. Signal Inf. Process. Netw. Process.*, 2(2):120–136, 2016.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. SPIDER: near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 689–699, 2018.

Jakovetić, D. A unification and generalization of exact distributed first-order methods. *IEEE Trans. Signal Inf. Process. Netw. Process.*, 5(1):31–46, 2018.

Kar, S., Moura, J. M. F., and Ramanan, K. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Trans. Inf. Theory*, 58(6):3575–3605, 2012.

Li, B., Cen, S., Chen, Y., and Chi, Y. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *J. Mach. Learn. Res.*, 21(180):1–51, 2020a.

Li, H. and Lin, Z. Revisiting extra for smooth distributed optimization. *SIAM J. Optim.*, 30(3):1795–1821, 2020.

Li, H., Lin, Z., and Fang, Y. Optimal accelerated variance reduced EXTRA and DIGing for strongly convex and smooth decentralized optimization. *arXiv preprint arXiv:2009.04373*, 2020b.

Li, Z., Shi, W., and Yan, M. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Trans. Signal Process.*, 67(17):4494–4506, 2019.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Adv. Neural Inf. Process. Syst.*, pp. 5330–5340, 2017.

Liu, D., Nguyen, L. M., and Tran-Dinh, Q. An optimal hybrid variance-reduced algorithm for stochastic composite nonconvex optimization. *arXiv preprint arXiv:2008.09055*, 2020.

Lü, Q., Liao, X., Li, H., and Huang, T. A computation-efficient decentralized algorithm for composite constrained optimization. *IEEE Trans. Signal Inf. Process. Netw.*, 6:774–789, 2020.

Lu, S., Zhang, X., Sun, H., and Hong, M. GNSD: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop*, pp. 315–321, 2019.

Mokhtari, A. and Ribeiro, A. DSA: Decentralized double stochastic averaging gradient algorithm. *J. Mach. Learn. Res.*, 17(1):2165–2199, 2016.

Nedić, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48, 2009.

Nedić, A., Olshevsky, A., and Rabbat, M. G. Network topology and communication-computation tradeoffs in decentralized optimization. *P. IEEE*, 106(5):953–976, 2018.

Nedich, A., Olshevsky, A., and Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.*, 27(4):2597–2633, 2017.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2009.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takac, M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proc. 34th Int. Conf. Mach. Learn.*, pp. 2613–2621, 2017.

Pan, T., Liu, J., and Wang, J. D-SPIDER-SFO: A decentralized optimization algorithm with faster convergence rate

for nonconvex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1619–1626, 2020.

Pham, N. H., Nguyen, L. M., Phan, D. T., and Tran-Dinh, Q. ProxSARAH: an efficient algorithmic framework for stochastic composite nonconvex optimization. *J. Mach. Learn. Res.*, 21(110):1–48, 2020.

Pu, S. and Nedich, A. Distributed stochastic gradient tracking methods. *Math. Program.*, pp. 1–49, 2020.

Qu, G. and Li, N. Harnessing smoothness to accelerate distributed optimization. *IEEE Trans. Control. Netw. Syst.*, 5(3):1245–1260, 2017.

Rajawat, K. and Kumar, C. A primal-dual framework for decentralized stochastic optimization. *arXiv preprint arXiv:2012.04402*, 2020.

Shi, W., Ling, Q., Wu, G., and Yin, W. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.*, 25(2):944–966, 2015.

Sun, H., Lu, S., and Hong, M. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9217–9228, 13–18 Jul 2020.

Taheri, H., Mokhtari, A., Hassani, H., and Pedarsani, R. Quantized decentralized stochastic learning over directed graphs. In *International Conference on Machine Learning*, pp. 9324–9333, 2020.

Tang, H., Lian, X., Yan, M., Zhang, C., and Liu, J. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pp. 4848–4856, 2018.

Tran-Dinh, Q., Pham, N. H., Phan, D. T., and Nguyen, L. M. A hybrid stochastic optimization framework for stochastic composite nonconvex optimization. *Math. Program.*, 2020.

Tsitsiklis, J., Bertsekas, D., and Athans, M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Autom. Control*, 31(9): 803–812, 1986.

Vlaski, S. and Sayed, A. H. Distributed learning in nonconvex environments–Part II: Polynomial escape from saddle-points. *arXiv:1907.01849*, 2019.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost and momentum: Faster variance reduction algorithms. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2403–2413, 2019.

Xi, C., Xin, R., and Khan, U. A. ADD-OPT: Accelerated distributed directed optimization. *IEEE Trans. Autom. Control*, 63(5):1329–1339, 2017.

Xin, R., Kar, S., and Khan, U. A. Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence. *IEEE Signal Process. Mag.*, 37(3):102–113, 2020a.

Xin, R., Khan, U. A., and Kar, S. A near-optimal stochastic gradient method for decentralized non-convex finite-sum optimization. *arXiv preprint arXiv:2008.07428*, 2020b.

Xin, R., Khan, U. A., and Kar, S. A fast randomized incremental gradient method for decentralized non-convex optimization. *arXiv preprint arXiv:2011.03853*, 2020c.

Xin, R., Khan, U. A., and Kar, S. Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Trans. Signal Process.*, 68:6255–6271, 2020d.

Xin, R., Khan, U. A., and Kar, S. An improved convergence analysis for decentralized online stochastic non-convex optimization. *arXiv preprint arXiv:2008.04195*, 2020e.

Xin, R., Pu, S., Nedić, A., and Khan, U. A. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11):1869–1889, 2020f.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *Proc. IEEE Conf. Decis. Control*, pp. 2055–2060, 2015.

Xu, J., Tian, Y., Sun, Y., and Scutari, G. Distributed algorithms for composite optimization: Unified and tight convergence analysis. *arXiv:2002.11534*, 2020.

Yi, X., Zhang, S., Yang, T., Chai, T., and Johansson, K. H. A primal-dual SGD algorithm for distributed nonconvex optimization. *arXiv preprint arXiv:2006.03474*, 2020.

Yuan, K., Ying, B., Liu, J., and Sayed, A. H. Variance-reduced stochastic learning by networked agents under random reshuffling. *IEEE Trans. Signal Process.*, 67(2): 351–366, 2018.

Yuan, K., Alghunaim, S. A., Ying, B., and Sayed, A. H. On the influence of bias-correction on distributed stochastic optimization. *IEEE Trans. Signal Process.*, 2020.

Zhou, D., Xu, P., and Gu, Q. Stochastic nested variance reduction for nonconvex optimization. *J. Mach. Learn. Res.*, 2020.