Proximal Stochastic Recursive Momentum Methods for Nonconvex Composite Decentralized Optimization

Gabriel Mancino-Ball[1], Shengnan Miao[1], Yangyang Xu[1], Jie Chen[2]

[1]Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180
[2]MIT IBM-Watson AI Lab, IBM Research, Cambridge, MA 02142
mancig@rpi.edu, snmiao236@gmail.com, xuy21@rpi.edu, chenjie@us.ibm.com

Abstract

Consider a network of N decentralized computing agents collaboratively solving a nonconvex stochastic composite problem. In this work, we propose a single-loop algorithm, called DEEPSTORM, that achieves optimal sample complexity for this setting. Unlike double-loop algorithms that require a large batch size to compute the (stochastic) gradient once in a while, DEEPSTORM uses a small batch size, creating advantages in occasions such as streaming data and online learning. This is the first method achieving optimal sample complexity for decentralized nonconvex stochastic composite problems, requiring $\mathcal{O}(1)$ batch size. We conduct convergence analysis for DEEPSTORM with both constant and diminishing step sizes. Additionally, under proper initialization and a small enough desired solution error, we show that DEEPSTORM with a constant step size achieves a network-independent sample complexity, with an additional linear speed-up with respect to N over centralized methods. All codes are made available at https://github.com/gmancino/DEEPSTORM.

Introduction

Recent years have seen an increase in designing efficient algorithms for solving large-scale machine learning problems, over a network of N computing agents connected by a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Agents collaboratively solve the following composite problem:

$$\min_{\mathbf{x}} \frac{1}{N} \sum_{i=1}^{N} \Big\{ \phi_i(\mathbf{x}) \triangleq f_i(\mathbf{x}) + r(\mathbf{x}) \Big\}, \tag{1}$$

where the decision variable $\mathbf{x} \in \mathbb{R}^{1\times p}$ is treated as a row vector; f_i is a smooth, possibly nonconvex function known only to agent i; and r is a convex, possibly non-smooth regularizer common to all agents. Agents i and j can communicate only if $(i, j) \in \mathcal{E}$. Many real-world applications in machine learning (Vogels et al. 2021; Ying et al. 2021; Yuan et al. 2021; Chamideh, Tärneberg, and Kihl 2021) and reinforcement learning (Zhang et al. 2018; Qu et al. 2019) fit the form of (1). Such scenarios differ from the centralized setting (McMahan et al. 2017; T. Dinh, Tran, and Nguyen 2020), where the agents are assumed to be able to communicate with one another globally via either a parameter server or a collective communication protocol. This setting arises naturally when data is distributed over a large geographic region or when a centralized communication structure is too costly (Xin, Khan, and Kar 2021a).

Utilizing the communication topology induced by $\mathcal{G}$, we reformulate (1) into the following equivalent *decentralized consensus optimization problem*:

$$\min_{\mathbf{x}_1,\dots,\mathbf{x}_N} \frac{1}{N} \sum_{i=1}^{N} \phi_i(\mathbf{x}_i), \text{ s.t. } \mathbf{x}_i = \mathbf{x}_j, \ \ \forall (i,j) \in \mathcal{E}. \tag{2}$$

Problem (2) allows for agents to maintain and update a local copy of the decision variable by locally computing gradients and performing neighbor communications.

The existence of a non-smooth regularizer r renders many decentralized optimization methods for a smooth objective inappropriate. We assume that r admits an easily computable (e.g. closed form) proximal mapping. Moreover, we are interested in the case where each local function f_i takes the following expectation form:

$$f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi\sim\mathcal{D}_i} \left[f_i(\mathbf{x}; \xi) \right], \tag{3}$$

with a slight abuse of notation for ease of exposition. In such a case, agents locally compute stochastic gradients of f_i. We adapt ideas from recent advances of stochastic optimization to the decentralized setting, by combining variance reduction techniques (Johnson and Zhang 2013; Nguyen et al. 2017; Allen-Zhu 2018; Wang et al. 2019; Cutkosky and Orabona 2019; Tran-Dinh et al. 2022) with gradient tracking (Lorenzo and Scutari 2016; Nedic, Olshevsky, and Shi 2017; Lu et al. 2019; Zhang and You 2020; Koloskova, Lin, and Stich 2021), to produce an algorithmic framework that achieves the optimal sample complexity bounds established in (Arjevani et al. 2022) for nonconvex stochastic methods.

Our framework, coined DEEPSTORM, is a single-loop algorithm with an attractive property that, besides the initial iteration, each agent only needs $\mathcal{O}(1)$ stochastic samples to compute a gradient estimate. Further, when a diminishing step size is used, even the first iteration does not need a large batch, at the expense of an additional logarithmic factor in the sample complexity result. Intuitively, DEEPSTORM utilizes a momentum based variance reduction technique (Cutkosky and Orabona 2019; Xu and Xu 2023; Levy, Kavis, and Cevher 2021; Tran-Dinh et al. 2022)

Method	$r \not\equiv 0$	Batch size	Sample complexity (per agent)
D-PSGD (Lian et al. 2017)	✗	$\mathcal{O}(1)$	$\mathcal{O}\left(\max\left\{\frac{1}{N\varepsilon^2}, \frac{N^2}{(1-\rho)^2\varepsilon}\right\}\right)$
DSGT (Xin, Khan, and Kar 2021b)	✗	$\mathcal{O}(1)$	$\mathcal{O}\left(\max\left\{\frac{1}{N\varepsilon^2}, \frac{\rho N}{(1-\rho)^3\varepsilon}\right\}\right)$
D-GET (Sun, Lu, and Hong 2020)	✗	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ or $\mathcal{O}\left(\frac{1}{\varepsilon^{0.5}}\right)$	$\mathcal{O}\left(\frac{1}{(1-\rho)^a\varepsilon^{1.5}}\right)$
GT-HSGD (Xin, Khan, and Kar 2021a)	✗	$\mathcal{O}\left(\frac{1}{\varepsilon^{0.5}}\right)$ then $\mathcal{O}(1)$	$\mathcal{O}\left(\max\left\{\frac{1}{N\varepsilon^{1.5}}, \frac{\rho^4}{N(1-\rho)^3\varepsilon}, \frac{\rho^{1.5}N^{0.5}}{(1-\rho)^{2.25}\varepsilon^{0.75}}\right\}\right)$
SPPDM (Wang et al. 2021)	✓	$\Omega(\frac{N}{\varepsilon})$	$\mathcal{O}\left(\frac{1}{(1-\rho)^b\varepsilon^2}\right)$
ProxGT-SR-O/E (Xin et al. 2021)	✓	$\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ or $\mathcal{O}\left(\frac{1}{\varepsilon^{0.5}}\right)$	$\mathcal{O}\left(\frac{1}{N\varepsilon^{1.5}}\right)^{\dagger}$
Theorem 1	✓	$\mathcal{O}\left(\frac{1}{\varepsilon^{0.5}}\right)$ then $\mathcal{O}(1)$	$\mathcal{O}\left(\max\left\{\frac{1}{N\varepsilon^{1.5}}, \frac{1}{(1-\rho)^2\varepsilon}, \frac{N^{0.5}}{\varepsilon^{0.75}}\right\}\right)^{\ddagger}$
Theorem 2	✓	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}\left(\frac{1}{\varepsilon^{1.5}}\right)$

Table 1: Comparison between DEEPSTORM (bottom two rows) and representative decentralized stochastic nonconvex methods. The sample complexity takes into account both the stationarity and consensus violation. Since D-GET and SPPDM do not show the dependence on ρ, we use unspecified powers a and b, following the practice of (Xin, Khan, and Kar 2021a). †The sample complexity of ProxGT-SR-O/E is independent of ρ by *requiring* multiple communications per update; this is similar to our result in Theorem 2. ‡With multiple communications and $\varepsilon \leq N^{-2}$, Theorem 1 guarantees our algorithm attains the optimal $\mathcal{O}\left(N^{-1}\varepsilon^{-1.5}\right)$ sample complexity, but with a smaller batch size than ProxGT-SR-O/E .

to guarantee convergence under a small batch size. The use of momentum simultaneously accelerates the computation and communication complexities over non-momentum based methods in the small batch setting; see Table 1 for a comparison. The recent ProxGT-SR-O/E (Xin et al. 2021) method can also achieve optimal sample complexity for solving (2), but at the expense of performing a double-loop which requires a large (stochastic) gradient computation every time the inner loop is completed. In scenarios where the batch size is uncontrollable, such as streaming or online learning, DEEPSTORM is advantageous.

When discussing sample complexity, it is paramount to specify the impact of the communication graph $\mathcal{G}$. With a constant step size, we show that under a sufficient amount of initial, or *transient*, iterations and proper initialization, DEEPSTORM behaves similarly to its centralized counterparts (Cutkosky and Orabona 2019; Levy, Kavis, and Cevher 2021; Tran-Dinh et al. 2022), while enjoying a linear speed-up with respect to N.

We summarize the contributions of this work below:

- We propose a novel decentralized framework, DEEPSTORM, for nonconvex stochastic composite optimization problems. We show that DEEPSTORM achieves the optimal sample complexity with respect to solution accuracy, where each agent needs only $\mathcal{O}(1)$ samples to compute a local stochastic gradient. To the best of our knowledge, this is the first decentralized method that achieves optimal sample complexity for solving *stochastic composite* problems by using only small batches.
- Additionally, we establish convergence guarantees of DEEPSTORM with both constant and diminishing step sizes. When a constant step size is used, we show that under sufficiently many transient iterations and proper initialization, DEEPSTORM achieves a linear speed-up with respect to N, signifying an advantage over analogous centralized variance reduction methods (Cutkosky and Orabona 2019; Levy, Kavis, and Cevher 2021; Tran-Dinh et al. 2022).

Related Works

A rich body of literature exists for solving the problem (2) in the decentralized setting. We discuss related works below.

Nonconvex decentralized methods. Of particular relevance to this work are methods for nonconvex f_i's. When f_i takes the finite-sum form, deterministic methods (with full gradient computation) such as DGD (Zeng and Yin 2018), Near-DGD (Iakovidou and Wei 2021), Prox-PDA (Hong, Hajinezhad, and Zhao 2017), xFILTER (Sun and Hong 2019), and SONATA (Scutari and Sun 2019) converge to an ε-stationary point in $\mathcal{O}\left(\varepsilon^{-1}\right)$ iterations. They all work for the case $r \equiv 0$ only, except SONATA. For stochastic methods, we summarize a few representative ones in Table 1, including the information of whether they handle $r \not\equiv 0$. Note that D-PSGD (Lian et al. 2017) extends the convergence results of DGD; D^2 (Tang et al. 2018) further improves over D-PSGD by relaxing a dissimilarity assumption.

Gradient tracking (Lorenzo and Scutari 2016; Nedic, Olshevsky, and Shi 2017) has been introduced as a tool to track the gradient of the global objective and has been studied extensively in the nonconvex and stochastic setting, under different names (Zhang and You 2020; Lu et al. 2019; Koloskova, Lin, and Stich 2021; Xin, Khan, and Kar 2021b). Many works now utilize this technique to improve the performance of their methods; those that mimic the SARAH (Nguyen et al. 2017) and Spider (Wang, Yin, and Zeng 2019) updates have become popular for their improved

theoretical convergence rates. D-SPIDER-SFO (Pan, Liu, and Wang 2020) and D-GET (Sun, Lu, and Hong 2020) are two such methods. When f_i takes the finite-sum form, GT-SARAH (Xin, Khan, and Kar 2022) and DESTRESS (Li, Li, and Chi 2022) improve the analysis of D-GET by obtaining an optimal sample complexity and an optimal communication complexity, respectively. All these methods require computing a stochastic gradient with a large batch size every few iterations.

GT-HSGD (Xin, Khan, and Kar 2021a) can be considered a special case of our method. It uses a stochastic gradient estimator of the form proposed in (Cutkosky and Orabona 2019; Levy, Kavis, and Cevher 2021), requiring a large initial batch size, followed by $\mathcal{O}(1)$ batch size subsequently. The convergence analysis of GT-HSGD requires $r \equiv 0$; hence part of our work is to extend it to the case of $r \not\equiv 0$. Similar extensions have been proposed for other methods; for example, ProxGT-SR-O/E (Xin et al. 2021) extends D-GET, GT-SARAH, and DESTRESS. Additionally, the primal-dual method SPPDM (Wang et al. 2021) is shown to converge in $\mathcal{O}\left(\varepsilon^{-1}\right)$ communications, but it requires a large batch size proportional to ε^{-1}. Using such a batch size can negatively impact the performance on machine learning problems (Keskar et al. 2017).

Other decentralized methods. Several other decentralized methods exist for scenarios differing from that considered here. They include methods which study asynchronous updates, communication compression, time-varying network topologies, or convex-only problems. For examples of such works, please refer to the longer version of this paper (Mancino-Ball et al. 2022).

DEEPSTORM Framework

We first state the assumed conditions of each ϕ_i and the communication graph $\mathcal{G}$. They are standard in variance reduction (Cutkosky and Orabona 2019; Xu and Xu 2023; Tran-Dinh et al. 2022) and decentralized methods (Lian et al. 2017; Sun, Lu, and Hong 2020; Xin, Khan, and Kar 2021b).

Assumption 1 *The following conditions hold.*

(i) The regularizer function r is convex and admits an easily computable proximal mapping.

(ii) Each component function f_i is mean-squared L-smooth; i.e. there exists a constant $0 < L < \infty$ such that $\forall\, \mathbf{a}, \mathbf{b} \in \mathbb{R}^{1\times p}$ and $\forall\, i = 1, \ldots, N$,

$$\mathbb{E}_\xi \left\|\nabla f_i(\mathbf{a};\xi) - \nabla f_i(\mathbf{b};\xi)\right\|_2^2 \le L^2 \left\|\mathbf{a}-\mathbf{b}\right\|_2^2. \quad (4)$$

(iii) There exists $\sigma > 0$ such that $\forall\, \mathbf{a} \in \mathbb{R}^{1\times p}$,

$$\begin{aligned} &\mathbb{E}_\xi[\nabla f_i(\mathbf{a};\xi)] = \nabla f_i(\mathbf{a}), \\ &\mathbb{E}\left\|\nabla f_i(\mathbf{a};\xi) - \nabla f_i(\mathbf{a})\right\|_2^2 \le \sigma^2. \end{aligned} \quad (5)$$

(iv) The global function $\phi = \frac{1}{N}\sum_{i=1}^N \phi_i$ is lower bounded; i.e. there exists a constant ϕ^* such that

$$-\infty < \phi^* \le \phi(\mathbf{a}), \ \ \forall \mathbf{a} \in \mathbb{R}^{1\times p}. \quad (6)$$

Assumption 2 *The graph $\mathcal{G}$ is connected and undirected. It can be represented by a* mixing matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ *such that:*

Algorithm 1: DEEPSTORM

Input: Initial $\mathbf{X}^{(0)}$, mixing rounds T_0, T, iteration K, and $\{\alpha_k\}$, $\{\beta_k\}$

Compute $\mathbf{d}_i^{(0)} = \frac{1}{m_0}\sum_{\xi\in B_i^{(0)}} \nabla f_i(\mathbf{x}_i^{(0)};\xi)\ \forall i$
Communicate to obtain $\mathbf{Y}^{(0)} = \boldsymbol{\mathcal{W}}_{T_0}(\mathbf{D}^{(0)})$
for $k = 0, \ldots, K-1$ **do**
  Communicate to obtain $\mathbf{Z}^{(k)} = \boldsymbol{\mathcal{W}}_T(\mathbf{X}^{(k)})$
  Update local decision variables by (9)
  Obtain local gradient estimator by (10)
  Communicate to update gradient tracking variable $\mathbf{Y}^{(k+1)} = \boldsymbol{\mathcal{W}}_T(\mathbf{Y}^{(k)} + \mathbf{D}^{(k+1)} - \mathbf{D}^{(k)})$
end for

Output: $\mathbf{Z}^{(\tau)}$ with τ chosen randomly from $\{0, \ldots, K-1\}$

*(i) **(Decentralized property)** $w_{ij} > 0$ if $(i,j) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise;*

*(ii) **(Symmetric property)** $\mathbf{W} = \mathbf{W}^\top$;*

*(iii) **(Null-space property)** $\mathrm{null}\,(\mathbf{I}-\mathbf{W}) = \mathrm{span}\{\mathbf{e}\}$, where $\mathbf{e} \in \mathbb{R}^N$ is the vector of all ones; and*

*(iv) **(Spectral property)** the eigenvalues of $\mathbf{W}$ lie in the range $(-1, 1]$ with*

$$\rho \triangleq \left\|\mathbf{W} - \frac{1}{N}\mathbf{e}\mathbf{e}^\top\right\|_2 < 1. \quad (7)$$

Note that the entry values of $\mathbf{W}$ can be flexibly designed as long as Assumption 2 holds. One example is $\mathbf{W} = \mathbf{I} - \mathbf{L}/\tau$, where $\mathbf{L}$ is the combinatorial Laplacian of $\mathcal{G}$ and τ is a value greater than half of its largest eigenvalue. It is not hard to see that the consensus constraint $\mathbf{x}_i = \mathbf{x}_j$ for all $(i,j) \in \mathcal{E}$ in (2) is equivalent to $\mathbf{W}\mathbf{X} = \mathbf{X}$, where the i-th row of $\mathbf{X}$ is $\mathbf{x}_i$. The value ρ in (7) indicates the connectedness of the graph. The quantity $1-\rho$ is sometimes referred to as the *spectral gap*; a higher value suggests that the graph is more connected and consensus of the $\mathbf{x}_i$'s is easier to achieve.

Under Assumptions 1 and 2, we now present the DEEPSTORM framework. We start with the basic algorithm and later generalize the simple communication (using $\mathbf{W}$) with a more general communication operator, denoted by $\boldsymbol{\mathcal{W}}_T$.

Basic algorithm. Let $\mathbf{x}_i^{(k)}$ be the k-th iterate for agent i, and let the matrix $\mathbf{X}^{(k)}$ contain all the k-th iterates among agents, stacked as a matrix. We will similarly use such vector and matrix notations for other variables. Our **DE**c**E**ntralized **P**roximal **STO**chastic **R**ecursive **M**omentum framework, DEEPSTORM, uses a variance reduction variable $\mathbf{d}_i^{(k)}$ and a gradient tracking variable $\mathbf{y}_i^{(k)}$ to improve the convergence of $\mathbf{x}_i^{(k)}$. DEEPSTORM contains the following steps in each iteration k:

1. Communicate the local variables:

$$\mathbf{Z}^{(k)} = \mathbf{W}\mathbf{X}^{(k)}. \quad (8)$$

2. Update each local variable (e.g. by using proximal mappings):

$$\mathbf{x}_i^{(k+1)} = \underset{\mathbf{x}_i}{\mathrm{argmin}}\{\alpha_k r(\mathbf{x}_i) + \frac{1}{2}\left\|\mathbf{x}_i - \left(\mathbf{z}_i^{(k)} - \alpha_k \mathbf{y}_i^{(k)}\right)\right\|^2\}. \quad (9)$$

3. Update the variance reduction variable:

$$\mathbf{d}_i^{(k+1)} = (1-\beta_k)\left(\mathbf{d}_i^{(k)} + \mathbf{v}_i^{(k+1)} - \mathbf{u}_i^{(k+1)}\right) + \beta_k \tilde{\mathbf{v}}_i^{(k+1)}, \tag{10}$$

where

$$\begin{aligned} \mathbf{v}_i^{(k+1)} &= \frac{1}{m}\sum_{\xi \in B_i^{(k+1)}} \nabla f_i(\mathbf{x}_i^{(k+1)}; \xi), \\ \mathbf{u}_i^{(k+1)} &= \frac{1}{m}\sum_{\xi \in B_i^{(k+1)}} \nabla f_i(\mathbf{x}_i^{(k)}; \xi). \end{aligned} \tag{11}$$

Here, $B_i^{(k+1)}$ is a batch of m samples at the current iteration. Note that while $\mathbf{v}_i^{(k+1)}$ is evaluated at the current iterate, $\mathbf{u}_i^{(k+1)}$ is evaluated at the previous iterate. We make the assumption that for all k and all agents i and j, $B_i^{(k+1)}$ and $B_j^{(k+1)}$ contain independent and mutually independent random variables. The part $\tilde{\mathbf{v}}_i^{(k+1)}$ can be any unbiased estimate of $\nabla f_i(\mathbf{x}_i^{(k+1)})$ with bounded variance; its details will be elaborated soon.

4. Update the gradient tracking variable via communication:

$$\mathbf{Y}^{(k+1)} = \mathbf{W}\left(\mathbf{Y}^{(k)} + \mathbf{D}^{(k+1)} - \mathbf{D}^{(k)}\right). \tag{12}$$

The step that updates the variance reduction variable, (10), is motivated by Hybrid-SGD (Tran-Dinh et al. 2022), which allows for a single-loop update. Intuitively, this variable is a convex combination of the SARAH (Nguyen et al. 2017) update and $\tilde{\mathbf{v}}_i^{(k+1)}$, allowing for strong variance reduction and meanwhile flexibility in design. By doing so, a constant batch size m suffices for convergence. This is a useful property in scenarios of online learning and real-time decision making, where it is unrealistic to obtain and store mega batches for training (Xu and Xu 2023; Xin, Khan, and Kar 2021a).

Examples of $\tilde{\mathbf{v}}_i^{(k+1)}$. The vector $\tilde{\mathbf{v}}_i^{(k+1)}$ in (10) can be any unbiased local gradient estimate. In this work, we consider two cases: either $\tilde{\mathbf{v}}_i^{(k+1)}$ is evaluated on another set of samples $\tilde{B}_i^{(k+1)}$, defined analogously to $B_i^{(k+1)}$ that is used to compute $\mathbf{v}_i^{(k+1)}$ in (11), such that

$$\begin{aligned} &\tilde{B}_i^{(k+1)} \text{ is independent of } B_i^{(k+1)} \text{ with} \\ &\mathbb{E}\left\|\tilde{\mathbf{v}}_i^{(k+1)} - \nabla f_i(\mathbf{x}_i^{(k+1)})\right\|_2^2 \le \hat{\sigma}^2; \end{aligned} \tag{v1}$$

or simply

$$\tilde{\mathbf{v}}_i^{(k+1)} = \mathbf{v}_i^{(k+1)} \text{ with } \mathbb{E}\left\|\mathbf{v}_i^{(k+1)} - \nabla f_i(\mathbf{x}_i^{(k+1)})\right\|_2^2 \le \hat{\sigma}^2, \tag{v2}$$

for some $\hat{\sigma} > 0$. Two possible unbiased estimators that satisfy (v1) are

$$\tilde{\mathbf{v}}_i^{(k+1)} = \frac{1}{m}\sum_{\tilde{\xi} \in \tilde{B}_i^{(k+1)}} \nabla f_i(\mathbf{x}_i^{(k+1)}; \tilde{\xi}), \tag{v1-SG}$$

$$\begin{aligned} \tilde{\mathbf{v}}_i^{(k+1)} &= \frac{1}{m}\sum_{\tilde{\xi} \in \tilde{B}_i^{(k+1)}} \nabla f_i(\mathbf{x}_i^{(k+1)}; \tilde{\xi}) \\ &+ \frac{1}{m}\sum_{\tilde{\tilde{\xi}} \in \tilde{\tilde{B}}_i^{(\tau_{k+1})}} \nabla f_i(\mathbf{x}_i^{(\tau_{k+1})}; \tilde{\tilde{\xi}}) - \frac{1}{m}\sum_{\tilde{\xi} \in \tilde{B}_i^{(k+1)}} \nabla f_i(\mathbf{x}_i^{(\tau_{k+1})}; \tilde{\xi}), \end{aligned} \tag{v1-SVRG}$$

for some $\tau_{k+1} < k+1$. The first estimator is a standard one, evaluated by using a batch $\tilde{B}_i^{(k+1)}$ independent of $B_i^{(k+1)}$. The second estimator, which introduces further variance reduction, uses an additional past-time iterate $\mathbf{x}_i^{(\tau_{k+1})}$ and a batch $\tilde{\tilde{B}}_i^{(\tau_{k+1})}$, whose size is generally greater than m. Such an update is inspired by the SVRG method (Johnson and Zhang 2013). Here, we have $\hat{\sigma}^2 = m^{-1}\sigma^2$ for the estimators (v1-SG) and (v2); while $\hat{\sigma}^2 = \left(3m^{-1} + 6\left|\tilde{\tilde{B}}_i^{(\tau_{k+1})}\right|^{-1}\right)\sigma^2$ for (v1-SVRG), where we recall that σ^2 comes from (5). Note that beyond the two examples, our proof techniques hold for any unbiased estimator satisfying (v1), leaving more open designs.

Generalized communication. Steps (8) and (12) use the mixing matrix to perform weighted averaging of neighbor information. The closer $\mathbf{W}$ is to $\frac{1}{N}\mathbf{e}\mathbf{e}^\top$, the more uniform the rows of $\mathbf{X}^{(k+1)}$ are, implying agents are closer to consensus. Hence, to improve convergence, we can apply multiple mixing rounds in each iteration. To this end, we generalize the network communication by using an operator $\boldsymbol{\mathcal{W}}_T$, which is a degree-$T$ polynomial in $\mathbf{W}$ that must satisfy Assumption 2 parts (*ii*)–(*iv*). We adopt Chebyshev acceleration (Auzinger and Melenk 2011; Scaman et al. 2017; Xin et al. 2021; Li, Li, and Chi 2022), which defines for any input matrix $\mathbf{B}_0$, $\mathbf{B}_T = \boldsymbol{\mathcal{W}}_T(\mathbf{B}_0)$, where $\mathbf{B}_1 = \mathbf{W}\mathbf{B}_0$, $\mu_0 = 1$, $\mu_1 = \frac{1}{\rho}$ for ρ defined in (7), and recursively,

$$\begin{aligned} \mu_{t+1} &= \frac{2}{\rho}\mu_t - \mu_{t-1} \text{ and} \\ \mathbf{B}_{t+1} &= \frac{2\mu_t}{\rho\mu_{t+1}}\mathbf{W}\mathbf{B}_t - \frac{\mu_{t-1}}{\mu_{t+1}}\mathbf{B}_{t-1}, \text{ for } t \le T-1. \end{aligned} \tag{13}$$

It is not hard to see that $\mathbf{e}$ is an eigenvector of $\boldsymbol{\mathcal{W}}_T$, associated to eigenvalue 1, whose algebraic multiplicity is 1. Therefore,

$$\tilde{\rho} \triangleq \left\|\boldsymbol{\mathcal{W}}_T - \frac{1}{N}\mathbf{e}\mathbf{e}^\top\right\|_2 < 1. \tag{14}$$

Moreover, $\tilde{\rho}$ converges to zero exponentially with T, bringing $\boldsymbol{\mathcal{W}}_T$ rather close to an averaging operator (for details, see Appendix B in (Mancino-Ball et al. 2022)). Notice with $T = 1$, $\boldsymbol{\mathcal{W}}_T$ reduces to $\mathbf{W}$.

We summarize the overall algorithm in Algorithm 1, by replacing $\mathbf{W}$ in (8) and (12) with $\boldsymbol{\mathcal{W}}_T$. Additionally, see the discussions after Theorems 1 and 2 regarding the probability distribution for choosing the output of Algorithm 1.

Method	Train loss	Stationarity	% Non-zeros	Test accuracy
		a9a		
DSGT	0.3308±1.272e-4	0.0003±1.819e-4	74.18±160.09e-4	84.89±271.02e-4
SPPDM	0.5457±20.014e-4	0.001±2.99e-4	46.19±51.04e-4	76.38±0.0e-4
ProxGT-SR-E	0.545±85.017e-4	0.0491±64.099e-4	98.04±15.035e-4	76.38±0.0e-4
DEEPSTORM v1-SG	0.3306±9.46e-4	0.0002±1.292e-4	2.99±60.066e-4	84.96±1235.0e-4
DEEPSTORM v1-SVRG	0.3308±7.689e-4	**0.0001**±0.21278e-4	2.86±45.018e-4	84.94±929.04e-4
DEEPSTORM v2	**0.3277**±7.461e-4	**0.0001**±0.8179e-4	**1.92**±53.073e-4	**85.11**±478.03e-4
		MiniBooNE		
DSGT	0.3735±3.844e-4	0.0003±2.076e-4	81.83±227.0e-4	84.24±202.07e-4
SPPDM	0.5699±61.016e-4	0.0025±5.565e-4	35.32±77.02e-4	72.02±0.0e-4
ProxGT-SR-E	0.5663±32.027e-4	0.0115±7.57e-4	97.88±17.017e-4	72.02±0.0e-4
DEEPSTORM v1-SG	**0.3637**±19.015e-4	0.0002±0.6464e-4	4.34±60.07e-4	84.24±1902.0e-4
DEEPSTORM v1-SVRG	0.3653±23.054e-4	0.0002±0.9716e-4	4.42±65.068e-4	84.15±1974.0e-4
DEEPSTORM v2	**0.3637**±18.046e-4	**0.0001**±0.4136e-4	**4.2**±61.073e-4	**84.25**±1752.0e-4
		MNIST		
DSGT	0.1055±24.03e-4	0.0024±3.554e-4	51.05±896.0e-4	97.61±1346.0e-4
SPPDM	0.1851±55.065e-4	0.0051±2.058e-4	66.81±616.03e-4	95.55±1488.0e-4
ProxGT-SR-E	1.699±903.07e-4	0.21299±268.0e-4	91.4±70.087e-4	52.25±41480.0e-4
DEEPSTORM v1-SG	0.081±33.014e-4	0.0027±5.376e-4	10.31±70.031e-4	97.97±1261.0e-4
DEEPSTORM v1-SVRG	0.078±34.022e-4	0.0031±7.366e-4	10.99±82.095e-4	98.08±1485.0e-4
DEEPSTORM v2	**0.0768**±29.095e-4	**0.0016**±1.83e-4	**7.36**±50.07e-4	**98.15**±659.04e-4

Table 2: Comparisons of different methods by running them with the same number of data passes. Bold values indicate the best results and underlined values indicate the second best.

Convergence Results

For the convergence of DEEPSTORM, we start with the following standard definitions (Xu and Xu 2023; Xin et al. 2021).

Definition 1 *Given* $\mathbf{x} \in \mathrm{dom}(r)$, $\mathbf{y}$, *and* $\eta > 0$, *define the* proximal gradient mapping *of* $\mathbf{y}$ *at* $\mathbf{x}$ *to be*

$$P(\mathbf{x}, \mathbf{y}, \eta) \triangleq \frac{1}{\eta}\left(\mathbf{x} - \mathrm{prox}_{\eta r}(\mathbf{x} - \eta \mathbf{y})\right), \tag{15}$$

where prox *denotes the proximal operator* $\mathrm{prox}_g(\mathbf{v}) = \mathrm{argmin}_{\mathbf{u}}\left\{g(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \mathbf{v}\|_2^2\right\}$.

Definition 2 *A stochastic matrix* $\mathbf{X} \in \mathbb{R}^{N \times p}$ *is called a* stochastic ε-stationary point *of (2) if*

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|P(\mathbf{x}_i, \nabla f(\mathbf{x}_i), \eta)\|_2^2 + \frac{L^2}{N}\|\mathbf{X}_\perp\|_F^2\right] \leq \varepsilon, \tag{16}$$

where $\eta > 0$, $\nabla f \triangleq \frac{1}{N}\sum_{j=1}^{N}\nabla f_j$, $\mathbf{x}_i$ *is the* i*-th row of* $\mathbf{X}$, *and* $\mathbf{X}_\perp \triangleq \mathbf{X} - \frac{1}{N}\mathbf{e}\mathbf{e}^\top\mathbf{X}$ *is the difference between all* $\mathbf{x}_i$ *and their average* $\frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_j$.

Our analyses rely on the construction of two novel Lyapunov functions as indicated by Theorems 1 and 2 below. These Lyapunov functions guarantee convergence through the careful design of function coefficients which result from solving non-linear systems of inequalities in either the constant or diminishing step size case. We first consider the use of a constant step size. The convergence rate result is given in the following theorem. Its proof is given in Appendix C.2 in (Mancino-Ball et al. 2022).

Theorem 1 *Under Assumptions 1 and 2, let* $\left\{\left(\mathbf{X}^{(k)}, \mathbf{D}^{(k)}, \mathbf{Y}^{(k)}, \mathbf{Z}^{(k)}\right)\right\}$ *be obtained by Algorithm 1 via* (9), (12), *and* (10) *such that* $\tilde{\mathbf{v}}_i^{(k+1)}$ *is any unbiased gradient estimator that satisfies either* (v1) *or* (v2). *Further, let* α_k *and* β_k *be chosen as*

$$\alpha_k = \frac{\alpha}{K^{\frac{1}{3}}},\ \beta_k = \frac{144L^2\alpha^2}{NK^{\frac{2}{3}}},\ \textit{with}\ \alpha \leq \min\left\{\frac{K^{\frac{1}{3}}}{32L}, \frac{(1-\tilde{\rho})^2K^{\frac{1}{3}}}{64L}\right\}, \tag{17}$$

for all $k = 0, \ldots, K-1$. *Then, it holds that* $\beta_k \in (0,1)$ *for all* $k \geq 0$ *and that*

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\left\|P(\mathbf{z}_i^{(k)}, \nabla f(\mathbf{z}_i^{(k)}), \alpha_k)\right\|_2^2 + \frac{L^2}{N}\left\|\mathbf{Z}_\perp^{(k)}\right\|_F^2\right)\\ \leq \frac{512}{\alpha K^{\frac{2}{3}}}\left(\Phi^{(0)} - \phi^*\right) + \left(\frac{2048}{L(1-\tilde{\rho})^2K}\right)\frac{144^2L^4\alpha^3\hat{\sigma}^2}{N^2}\\ + \left(\frac{128}{3L^2\alpha K^{\frac{2}{3}}} + \frac{8192\alpha}{K^{\frac{4}{3}}} + \frac{2048\alpha}{NK^{\frac{4}{3}}}\right)\frac{144^2L^4\alpha^3\hat{\sigma}^2}{N^2},\end{aligned} \tag{18}$$

for some $\Phi^{(0)} > \phi^*$ *that depends on the initialization. Note that* $\Phi^{(k)}$ *is defined in* (C.43) *in (Mancino-Ball et al. 2022) for any* $k \geq 0$.

Network-independent sample complexity, linear speed-up, and communication complexity. Theorem 1 establishes convergence based on the sequence $\{\mathbf{Z}^{(k)}\}$ defined in (8). As a consequence, if we let each agent start with the same initial variable $\mathbf{x}^{(0)}$, set $\alpha = \frac{N^{\frac{2}{3}}}{64L}$ and

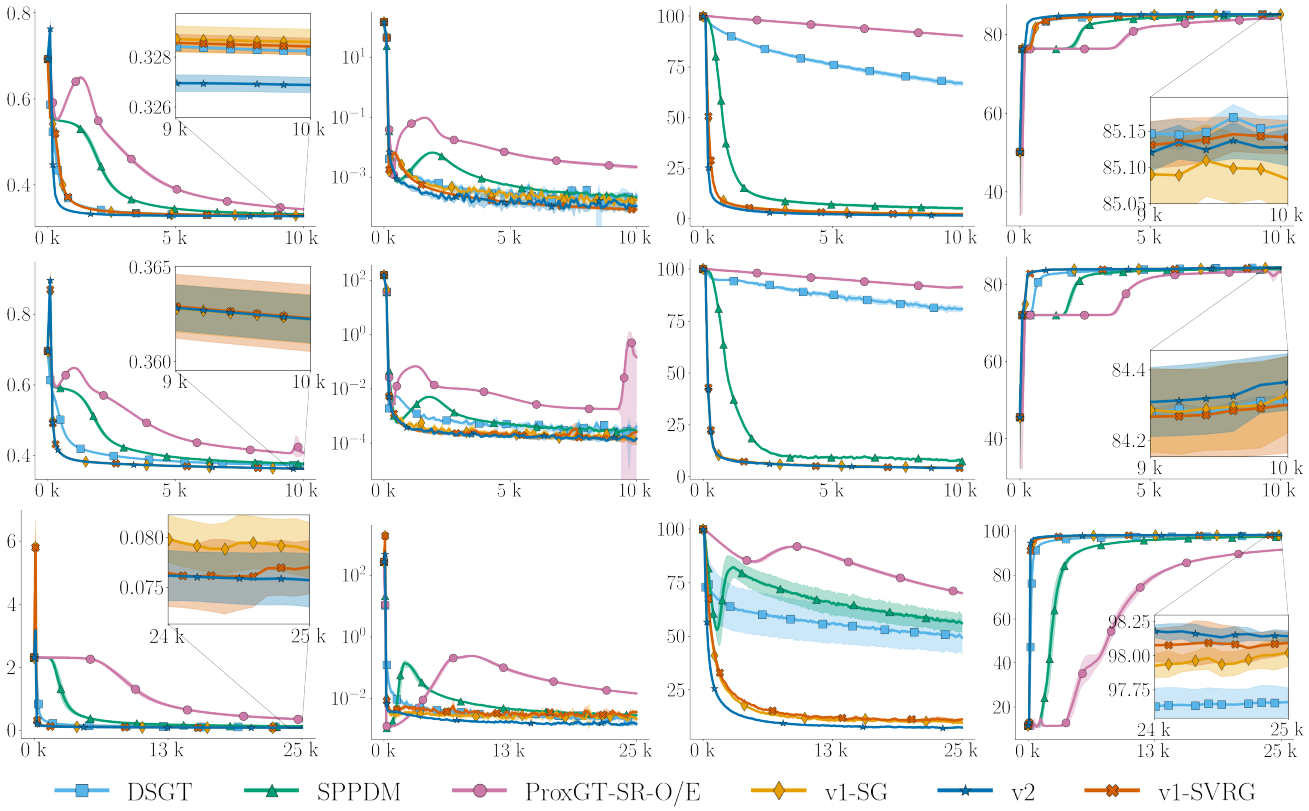

Figure 1: Comparison of different methods by running them with the same number of iterations (x-axis is iteration number). From top to bottom: a9a, MiniBooNE, MNIST. From left to right: training loss, stationarity violation (log scale), average percentage non-zeros, testing accuracy. The shaded regions indicate standard deviations (with some being small and unnoticeable).

the initial batch size $m_0 = \sqrt[3]{NK}$, and choose initial communication rounds $T_0 = \tilde{\mathcal{O}}\left((1-\rho)^{-0.5}\right)$ for $\mathbf{Y}^{(0)}$, then for all $K \geq \frac{N^2}{(1-\tilde{\rho})^6}$, DEEPSTORM achieves stochastic ε-stationarity for some iterate $\mathbf{Z}^{(\tau)}$, where τ is selected uniformly from $\{0, \ldots, K-1\}$, by using

$$\mathcal{O}\left(\max\left\{\frac{(L\Delta)^{\frac{3}{2}} + \hat{\sigma}^3}{N\varepsilon^{\frac{3}{2}}}, \frac{\hat{\sigma}^2}{(1-\tilde{\rho})^2\varepsilon}, \frac{\sqrt{N}\hat{\sigma}^{\frac{3}{2}}}{\varepsilon^{\frac{3}{4}}}\right\}\right) \tag{19}$$

local stochastic gradient computations. For the formal statement, see Corollary 1 in Appendix C.2 in (Mancino-Ball et al. 2022). Here, $\Delta = \Phi^{(0)} - \phi^*$ denotes an initial function gap, which is independent of $\tilde{\rho}$, N, and K. Moreover, when $\varepsilon \leq N^{-2}(1-\tilde{\rho})^4$, we see that $\mathcal{O}\left(N^{-1}\varepsilon^{-1.5}\right)$ dominates in (19); hence, this result manifests a linear speed-up with respect to N over the centralized counterparts (Cutkosky and Orabona 2019; Tran-Dinh et al. 2022) of DEEPSTORM. Furthermore, if the number of Chebyshev mixing rounds is $T = \lceil \frac{2}{\sqrt{1-\rho}} \rceil$, we have $(1-\tilde{\rho}) \geq \frac{1}{\sqrt{2}}$, which suggests that ε does not need to be small for the linear speed-up to hold. For details, see Lemma B.1 and Remark C.2 in (Mancino-Ball et al. 2022). The communication cost is $\mathcal{O}\left(T_0 + TK\right)$.

In parallel, we state a result for the case of diminishing step size. Its proof is given in Appendix C.3 in (Mancino-Ball et al. 2022).

Theorem 2 *Under the same assumptions as Theorem 1, let α_k and β_k be chosen as*

$$\alpha_k = \frac{\alpha}{(k+k_0)^{\frac{1}{3}}}, \quad \beta_k = 1 - \frac{\alpha_{k+1}}{\alpha_k} + 48L^2\alpha_{k+1}^2, \text{ with}$$
$$\alpha \leq \min\left\{\frac{k_0^{\frac{1}{3}}}{32L}, \frac{(1-\tilde{\rho})^2 k_0^{\frac{1}{3}}}{64L}\right\}, \tag{20}$$

for all $k = 0, \ldots, K-1$, where $k_0 \geq \lceil \frac{2}{1-\tilde{\rho}^3} \rceil$. Then, it holds that $\beta_k \in (0,1)$ for all $k \geq 0$ and that

$$\sum_{k=0}^{K-1} c\alpha_k \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\left\|P(\mathbf{z}_i^{(k)}, \nabla f(\mathbf{z}_i^{(k)}), \alpha_k)\right\|_2^2 + \frac{L^2}{N}\left\|\mathbf{Z}_\perp^{(k)}\right\|_F^2\right)$$
$$\leq 12\left(\hat{\Phi}^{(0)} - \phi^*\right) + \sum_{k=0}^{K-1}\left(\frac{1}{L^2\alpha_{k+1}} + \frac{48}{L(1-\tilde{\rho})^2}\right)\beta_k^2\hat{\sigma}^2, \tag{21}$$

for some $\hat{\Phi}^{(0)} > \phi^*$ that depends on initialization and $c \triangleq \frac{k_0^{\frac{1}{3}}}{2k_0^{\frac{1}{3}} + (k_0+1)^{\frac{1}{3}}} > \frac{1}{4}$. Note that $\hat{\Phi}^{(k)}$ is defined in (C.69) *in Appendix C in (Mancino-Ball et al. 2022) for any $k \geq 0$.*

Sample complexity. Theorem 2 establishes the convergence rate of DEEPSTORM with diminishing step sizes.

If we choose $k_0 = \lceil \frac{2}{(1-\tilde{\rho})^6} \rceil$ in (20), then DEEPSTORM achieves stochastic ε-stationarity for some iterate $\mathbf{Z}^{(\tau)}$, where τ is chosen according to (C.87) in (Mancino-Ball et al. 2022), by using $\tilde{\mathcal{O}}\left((1-\tilde{\rho})^{-3}\varepsilon^{-1.5}\right)$ local stochastic gradient computations; this sample complexity is network-dependent. However, by using an initialization technique similar to the case of constant step sizes above and letting the initial batch size be $\mathcal{O}(1)$, we can set the Chebyshev mixing rounds to be $T = \lceil \frac{2}{\sqrt{1-\rho}} \rceil$, so that $(1-\tilde{\rho})^{-1} \leq \sqrt{2}$. This leads to the network-independent sample complexity reported in Table 1. For a full statement of the complexity results, see Corollary 2 in Appendix C.3 and Remark C.4 in (Mancino-Ball et al. 2022).

Experiments

In this section, we empirically validate the convergence theory of DEEPSTORM and demonstrate its effectiveness in comparison with representative decentralized methods. We compare all versions of DEEPSTORM with DSGT (Lu et al. 2019; Zhang and You 2020; Koloskova, Lin, and Stich 2021; Xin, Khan, and Kar 2021b), SPPDM (Wang et al. 2021), and ProxGT-SR-O/E (Xin et al. 2021). DSGT uses gradient tracking but it is not designed for non-smooth objectives; nevertheless, it outperforms strong competitors (e.g. D-PSGD (Lian et al. 2017) and D^2 (Tang et al. 2018)) in practice (Zhang and You 2020; Xin, Khan, and Kar 2021b). SPPDM is a primal-dual method, but it does not utilize gradient tracking and its convergence theory requires a large batch size. ProxGT-SR-O/E is a double-loop algorithm, which requires using a mega-batch to compute the (stochastic) gradient at each outer iteration. All experiments are conducted using the AiMOS[1] supercomputer with eight NVIDIA Tesla V100 GPUs in total, with code implemented in PyTorch (v1.6.0) and OpenMPI (v3.1.4).

Problems. We conduct tests on three classification problems. Each local agent i has the objective $\phi_i(\mathbf{x}_i) = \frac{1}{M}\sum_{j=1}^{M} \ell\left(g\left(\mathbf{x}_i, \mathbf{a}_j\right), \mathbf{b}_j\right) + \lambda \left\|\mathbf{x}_i\right\|_1$, where $g(\mathbf{x}, \mathbf{a})$ is the output of a neural network with parameters $\mathbf{x}$ on data $\mathbf{a}$, and ℓ is the cross-entropy loss function between the output and the true label $\mathbf{b}$. The data is uniformly randomly split among the agents, each obtaining M training examples. The L_1 regularization promotes sparsity of the trained network. The regularization strength λ is set to 0.0001 following general practice.

Data sets and neural networks. The three data sets we experiment with are summarized in Table 2 in Appendix A in (Mancino-Ball et al. 2022). Two of them are tabular data and we use the standard multi-layer perceptron for g (one hidden layer with 64 units). The other data set contains images; thus, we use a convolutional neural network. Both neural networks use the tanh activation to satisfy the smoothness condition of the objective function.

Communication graphs. Each data set is paired with a different communication graph, indicated by, and visualized in, Table 2 in Appendix A in (Mancino-Ball et al. 2022). For the ladder and random graphs, the mixing matrix is set as $\mathbf{W} = \mathbf{I} - \gamma\mathbf{L}$, where γ is reciprocal of the maximum eigenvalue of the combinatorial Laplacian $\mathbf{L}$. For the ring graph, self-weighting and neighbor weights are set to be $\frac{1}{3}$.

Performance metrics. We evaluate on four metrics: training loss, stationarity violation, solution sparsity, and test accuracy. Further, we compare the methods with respect to data passes and algorithm iterations, which reflect the sample complexity and communication complexity, respectively. Note that for each iteration, all methods except SPPDM communicate two variables. For the training loss, stationarity violation, and test accuracy, we evaluate on the average solution $\bar{\mathbf{x}}$. The stationarity violation is defined as $\|\bar{\mathbf{x}} - \text{prox}_r\left(\bar{\mathbf{x}} - \nabla f(\bar{\mathbf{x}})\right)\|_2^2 + \sum_{i=1}^{N} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2$, which measures both optimality and consensus. For sparsity, we use the average percentage of non-zeros in each $\mathbf{x}_i$ prior to local communication.

Protocols. For hyperparameter selection, see Appendix A in (Mancino-Ball et al. 2022). We perform ten runs with different starting points for each dataset. In several runs for the MNIST dataset, DSGT and SPPDM converge to solutions with $\ll 1\%$ non-zero entries, but the training loss and test accuracy are not competitive at all. We keep only the five best runs for reporting the (averaged) performance.

Results. Table 2 summarizes the results for all performance metrics, by using the same number of data passes for all methods when convergence has been observed. For a9a and MiniBooNE, the results are averaged over passes 80 to 100; while for MNIST, over passes 180 to 200. Figure 1 compares different methods by using the same number of algorithm iterations.

Overall, we see that DEEPSTORM (all variants) generally yields a lower training loss and significantly fewer non-zeros in the solution than the other decentralized algorithms. This observation suggests that DEEPSTORM indeed solves the optimization problem (2) much more efficiently in terms of both data passes and iterations. Moreover, the test accuracy is also highly competitive, concluding the practical usefulness of DEEPSTORM.

Conclusion

We have presented a novel decentralized algorithm for solving the nonconvex stochastic composite problem (2) by leveraging variance reduction and gradient tracking. It is the first such work that achieves optimal sample complexity for this class of problems by using $\mathcal{O}(1)$ batch sizes. Our algorithm is a framework with an open term (see (10)), for which we analyze two examples that allow the framework to achieve network-independent complexity bounds, suggesting no sacrifice over centralized variance reduction methods. Our proof technique can be used to analyze more designs of the open term. While our work is one of the few studies on the nonconvex stochastic composite problem (2), our analysis is for the synchronous setting with a static communication graph. Analysis (or adaptation of the algorithm) for asynchronous or time-varying settings is an avenue of future investigation.

[1] See: https://cci.rpi.edu/aimos

Acknowledgments

This work was supported by the Rensselaer-IBM AI Research Collaboration, part of the IBM AI Horizons Network, NSF grants DMS-2053493 and DMS-2208394, and the ONR award N00014-22-1-2573.

References

Allen-Zhu, Z. 2018. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. *Journal of Machine Learning Research*, 18(221): 1–51.

Arjevani, Y.; Carmon, Y.; Duchi, J. C.; Foster, D. J.; Srebro, N.; and Woodworth, B. 2022. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*.

Auzinger, W.; and Melenk, J. M. 2011. Iterative Solution of Large Linear Systems. *TU Wien, Lecture Notes*.

Chamideh, S.; Tärneberg, W.; and Kihl, M. 2021. Evaluation of Decentralized Algorithms for Coordination of Autonomous Vehicles at Intersections. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 1954–1961.

Cutkosky, A.; and Orabona, F. 2019. Momentum-Based Variance Reduction in Non-Convex SGD. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hong, M.; Hajinezhad, D.; and Zhao, M.-M. 2017. Prox-PDA: The Proximal Primal-Dual Algorithm for Fast Distributed Nonconvex Optimization and Learning Over Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1529–1538. International Convention Centre, Sydney, Australia: PMLR.

Iakovidou, C.; and Wei, E. 2021. On the Convergence of NEAR-DGD for Nonconvex Optimization with Second Order Guarantees. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 259–264.

Johnson, R.; and Zhang, T. 2013. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Koloskova, A.; Lin, T.; and Stich, S. U. 2021. An Improved Analysis of Gradient Tracking for Decentralized Machine Learning. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Levy, K. Y.; Kavis, A.; and Cevher, V. 2021. STORM+: Fully Adaptive SGD with Recursive Momentum for Nonconvex Optimization. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Li, B.; Li, Z.; and Chi, Y. 2022. DESTRESS: Computation-Optimal and Communication-Efficient Decentralized Nonconvex Finite-Sum Optimization. *SIAM Journal on Mathematics of Data Science*, 4(3): 1031–1051.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30, 5330–5340. Curran Associates, Inc.

Lorenzo, P. D.; and Scutari, G. 2016. NEXT: In-Network Nonconvex Optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2): 120–136.

Lu, S.; Zhang, X.; Sun, H.; and Hong, M. 2019. GNSD: a Gradient-Tracking Based Nonconvex Stochastic Algorithm for Decentralized Optimization. In *2019 IEEE Data Science Workshop (DSW)*, 315–321.

Mancino-Ball, G.; Miao, S.; Xu, Y.; and Chen, J. 2022. Proximal Stochastic Recursive Momentum Methods for Nonconvex Composite Decentralized Optimization. *arXiv preprint arXiv:2211.11954*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

Nedic, A.; Olshevsky, A.; and Shi, W. 2017. Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs. *SIAM Journal on Optimization*, 27: 2597 – 2633.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 2613–2621. International Convention Centre, Sydney, Australia: PMLR.

Pan, T.; Liu, J.; and Wang, J. 2020. D-SPIDER-SFO: A Decentralized Optimization Algorithm with Faster Convergence Rate for Nonconvex Problems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 1619–1626. AAAI Press.

Qu, C.; Mannor, S.; Xu, H.; Qi, Y.; Song, L.; and Xiong, J. 2019. Value Propagation for Decentralized Networked Deep Multi-agent Reinforcement Learning. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.;

and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Scaman, K.; Bach, F.; Bubeck, S.; Lee, Y. T.; and Massoulié, L. 2017. Optimal Algorithms for Smooth and Strongly Convex Distributed Optimization in Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3027–3036. International Convention Centre, Sydney, Australia: PMLR.

Scutari, G.; and Sun, Y. 2019. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176(1): 497–544.

Sun, H.; and Hong, M. 2019. Distributed Non-Convex First-Order Optimization and Information Processing: Lower Complexity Bounds and Rate Optimal Algorithms. *IEEE Transactions on Signal Processing*, 67(22): 5912–5928.

Sun, H.; Lu, S.; and Hong, M. 2020. Improving the Sample and Communication Complexity for Decentralized Non-Convex Optimization: Joint Gradient Estimation and Tracking. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 9217–9228. Virtual: PMLR.

T. Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized Federated Learning with Moreau Envelopes. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 21394–21405. Curran Associates, Inc.

Tang, H.; Lian, X.; Yan, M.; Zhang, C.; and Liu, J. 2018. D^2: Decentralized Training over Decentralized Data. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4848–4856. Stockholmsmässan, Stockholm Sweden: PMLR.

Tran-Dinh, Q.; Pham, N. H.; Phan, D. T.; and Nguyen, L. M. 2022. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 191(2): 1005–1071.

Vogels, T.; He, L.; Koloskova, A.; Karimireddy, S. P.; Lin, T.; Stich, S. U.; and Jaggi, M. 2021. RelaySum for Decentralized Deep Learning on Heterogeneous Data. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 28004–28015. Curran Associates, Inc.

Wang, Y.; Yin, W.; and Zeng, J. 2019. Global Convergence of ADMM in Nonconvex Nonsmooth Optimization. *Journal of Scientific Computing*, 78(1): 29–63.

Wang, Z.; Ji, K.; Zhou, Y.; Liang, Y.; and Tarokh, V. 2019. SpiderBoost and Momentum: Faster Variance Reduction Algorithms. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d' Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wang, Z.; Zhang, J.; Chang, T.-H.; Li, J.; and Luo, Z.-Q. 2021. Distributed Stochastic Consensus Optimization With Momentum for Nonconvex Nonsmooth Problems. *IEEE Transactions on Signal Processing*, 69: 4486–4501.

Xin, R.; Das, S.; Khan, U. A.; and Kar, S. 2021. A Stochastic Proximal Gradient Framework for Decentralized Non-Convex Composite Optimization: Topology-Independent Sample Complexity and Communication Efficiency. *arXiv preprint arXiv:2110.01594*.

Xin, R.; Khan, U.; and Kar, S. 2021a. A Hybrid Variance-Reduced Method for Decentralized Stochastic Non-Convex Optimization. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 11459–11469. PMLR.

Xin, R.; Khan, U. A.; and Kar, S. 2021b. An Improved Convergence Analysis for Decentralized Online Stochastic Non-Convex Optimization. *IEEE Transactions on Signal Processing*, 69: 1842–1858.

Xin, R.; Khan, U. A.; and Kar, S. 2022. Fast Decentralized Nonconvex Finite-Sum Optimization with Recursive Variance Reduction. *SIAM Journal on Optimization*, 32(1): 1–28.

Xu, Y.; and Xu, Y. 2023. Momentum-Based Variance-Reduced Proximal Stochastic Gradient Method for Composite Nonconvex Stochastic Optimization. *Journal of Optimization Theory and Applications*, 196(1): 266–297.

Ying, B.; Yuan, K.; Chen, Y.; Hu, H.; PAN, P.; and Yin, W. 2021. Exponential Graph is Provably Efficient for Decentralized Deep Training. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 13975–13987. Curran Associates, Inc.

Yuan, K.; Chen, Y.; Huang, X.; Zhang, Y.; Pan, P.; Xu, Y.; and Yin, W. 2021. DecentLaM: Decentralized Momentum SGD for Large-Batch Deep Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3029–3039.

Zeng, J.; and Yin, W. 2018. On Nonconvex Decentralized Gradient Descent. *IEEE Transactions on Signal Processing*, 66: 2834 – 2848.

Zhang, J.; and You, K. 2020. Decentralized Stochastic Gradient Tracking for Non-convex Empirical Risk Minimization. *arXiv preprint arXiv:1909.02712*.

Zhang, K.; Yang, Z.; Liu, H.; Zhang, T.; and Basar, T. 2018. Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5872–5881. PMLR.