A One-Sample Decentralized Proximal Algorithm for Non-Convex Stochastic Composite Optimization

Tesi Xiao[1] **Xuxing Chen**[2] **Krishnakumar Balasubramanian**[1] **Saeed Ghadimi**[3]

[1]Department of Statistics, University of California, Davis
[2]Department of Mathematics, University of California, Davis
[3]Department of Management Sciences, University of Waterloo

Abstract

We focus on decentralized stochastic non-convex optimization, where n agents work together to optimize a composite objective function which is a sum of a smooth term and a non-smooth convex term. To solve this problem, we propose two single-time scale algorithms: `Prox-DASA` and `Prox-DASA-GT`. These algorithms can find ϵ-stationary points in $\mathcal{O}(n^{-1}\epsilon^{-2})$ iterations using constant batch sizes (i.e., $\mathcal{O}(1)$). Unlike prior work, our algorithms achieve a comparable complexity result without requiring large batch sizes, more complex per-iteration operations (such as double loops), or stronger assumptions. Our theoretical findings are supported by extensive numerical experiments, which demonstrate the superiority of our algorithms over previous approaches. Our code is available at `https://github.com/xuxingc/ProxDASA`.

1 INTRODUCTION

Decentralized optimization is a flexible paradigm for solving complex optimization problems in a distributed manner and has numerous applications in fields such as machine learning, robotics, and control systems. It has attracted increased attention due to the following benefits: (i) *Robustness*: Decentralized optimization is more robust than centralized optimization because each agent can operate independently, making the system more resilient to failures compared to a centralized system where a coordinator failure or overload can halt the entire system. (ii) *Privacy*: Decentralized optimization can provide greater privacy because each agent only has access to a limited subset of observations, which may help to protect sensitive information. (iii) *Scalability*: Decentralized optimization is highly scalable as it can handle large datasets in a distributed manner, thereby solving complex optimization problems that are difficult or even impossible to solve in a centralized setting.

Specifically, we consider the following decentralized composite optimization problems in which n agents collaborate to solve

$$\min_{x\in\mathbb{R}^d} \Phi(x) \coloneqq F(x) + \Psi(x),\ F(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} F_i(x), \quad (1)$$

where each function $F_i(x)$ is a smooth function only known to the agent i; $\Psi(x)$ is non-smooth, convex, and shared across all agents; $\Phi(x)$ is bounded below by $\Phi_* > -\infty$. We consider the stochastic setting where the exact function values and derivatives of F_i's are unavailable. In particular, we assume that $F_i(x) = \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[G_i(x,\xi_i)]$, where ξ_i is a random vector and $\mathcal{D}_i$ is the distribution used to generate samples for agent i. The agents form a connected and undirected network and can communicate with their neighbors to cooperatively solve (1). The communication network can be represented with $\mathbb{G} = (\mathcal{V}, \mathbf{W})$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes all devices and $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n\times n}$ is the weighted adjacency matrix indicating how two agents are connected.

A majority of the existing decentralized stochastic algorithms for solving (1), require large batch sizes to achieve convergence. The few algorithms that operate with constant batch sizes mainly rely on complicated variance reduction techniques and require stronger assumptions to establish convergence results. To the best of our knowledge, the question of whether it is possible to develop decentralized stochastic optimization algorithms to solve (1) without the above mentioned limitations, remains unresolved.

To address this, we propose the two decentralized stochastic proximal algorithms, `Prox-DASA` and `Prox-DASA-GT`, for solving (1) and make the following **contributions**:

- We show that `Prox-DASA` is capable of achieving convergence in both homogenous and bounded heterogeneous settings while `Prox-DASA-GT` works for general decentralized heterogeneous problems.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

- We show that both algorithms find an ϵ-stationary point in $\mathcal{O}(n^{-1}\epsilon^{-2})$ iterations using only $\mathcal{O}(1)$ stochastic gradient samples per agent and m communication rounds at each iteration, where m can be any positive integer. A topology-independent transient time can be achieved by setting $m = \lceil \frac{1}{\sqrt{1-\rho}} \rceil$, where ρ is the second-largest eigenvalue of the communication matrix.
- Through extensive experiments, we demonstrate the superiority of our algorithms over prior works.

A summary of our results and comparison to prior work is provided in Table 1.

Related Works on Decentralized Composite Optimization. Motivated by wide applications in constrained optimization [Lee and Nedic, 2013, Margellos et al., 2017] and non-smooth problems with a composite structure as (1), arising in signal processing [Ling and Tian, 2010, Mateos et al., 2010, Patterson et al., 2014] and machine learning [Facchinei et al., 2015, Hong et al., 2017], several works have studied the decentralized composite optimization problem in (1), a natural generalization of smooth optimization. For example, Shi et al. [2015], Li et al. [2019], Alghunaim et al. [2019], Ye et al. [2020], Xu et al. [2021], Li et al. [2021], Sun et al. [2022], Wu and Lu [2022] studied (1) in the convex setting. Furthermore, Facchinei et al. [2015], Di Lorenzo and Scutari [2016], Hong et al. [2017], Zeng and Yin [2018], Scutari and Sun [2019] studied (1) in the deterministic setting.

Although there has been a lot of research investigating decentralized composite optimization, the stochastic nonconvex setting, which is more broadly applicable, still lacks a full understanding. Wang et al. [2021] proposes `SPPDM`, which uses a proximal primal-dual approach to achieve $\mathcal{O}(\epsilon^{-2})$ sample complexity. `ProxGT-SA` and `ProxGT-SR-O` [Xin et al., 2021a] incorporate stochastic gradient tracking and multi-consensus update in proximal gradient methods and obtain $\mathcal{O}(n^{-1}\epsilon^{-2})$ and $\mathcal{O}(n^{-1}\epsilon^{-1.5})$ sample complexity respectively, where the latter further uses a `SARAH` type variance reduction method [Pham et al., 2020, Wang et al., 2019]. A recent work [Mancino-ball et al., 2023] proposes `DEEPSTORM`, which leverages the momentum-based variance reduction technique and gradient tracking to obtain $\mathcal{O}(n^{-1}\epsilon^{-1.5})$ and $\tilde{\mathcal{O}}(\epsilon^{-1.5})$ sample complexity under different stepsize choices. Nevertheless, existing works either require stronger assumptions [Mancino-ball et al., 2023] or increasing batch sizes [Wang et al., 2021, Xin et al., 2021a].

2 PRELIMINARIES

Notations. $\|\cdot\|$ denotes the ℓ_2-norm for vectors and Frobenius norm for matrices. $\|\cdot\|_2$ denotes the spectral norm for matrices. $\mathbf{1}$ represents the all-one vector, and $\mathbf{I}$ is the identity matrix as a standard practice. We identify vectors at agent i in the subscript and use the superscript for the algorithm step. For example, the optimization variable of agent i at step k is denoted as x_i^k, and z_i^k is the corresponding dual variable. We use uppercase bold letters to represent the matrix that collects all the variables from nodes (corresponding lowercase) as columns. We add an overbar to a letter to denote the average over all nodes. For example, we denote the optimization variables over all nodes at step k as $\mathbf{X}_k = \left[x_1^k, \ldots, x_n^k\right]$. The corresponding average over all nodes can be thereby defined as

$$\bar{x}^k = \frac{1}{n}\sum_{i=1}^n x_i^k = \frac{1}{n}\mathbf{X}_k\mathbf{1},$$

$$\bar{\mathbf{X}}_k = [\bar{x}^k, \ldots, \bar{x}^k] = \bar{x}^k\mathbf{1}^\top = \frac{1}{n}\mathbf{X}_k\mathbf{1}\mathbf{1}^\top.$$

For an extended valued function $\Psi : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$, its effective domain is written as $\mathrm{dom}(\Psi) = \{x \mid \Psi(x) < +\infty\}$. A function Ψ is said to be proper if $\mathrm{dom}(\Psi)$ is nonempty. For any proper closed convex function Ψ, $x \in \mathbb{R}^d$, and scalar $\gamma > 0$, the proximal operator is defined as

$$\mathbf{prox}_\Psi^\gamma(x) = \arg\min_{y\in\mathbb{R}^d}\left\{\frac{1}{2\gamma}\|y-x\|^2 + \Psi(y)\right\}.$$

For $x, z \in \mathbb{R}^d$ and $\gamma > 0$, the proximal gradient mapping of z at x is defined as

$$\mathcal{G}(x, z, \gamma) = \frac{1}{\gamma}\left(x - \mathbf{prox}_\Psi^\gamma(x - \gamma z)\right).$$

All random objects are properly defined in a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and write $x \in \mathcal{H}$ if x is $\mathcal{H}$-measurable given a sub-σ-algebra $\mathcal{H} \subseteq \mathscr{F}$ and a random vector x. We use $\sigma(\cdot)$ to denote the σ-algebra generated by all the argument random vectors.

Assumptions. Next, we list and discuss the assumptions made in this work.

Assumption 1. *The weighted adjacency matrix $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{n\times n}$ is symmetric and doubly stochastic, i.e.,*

$$\mathbf{W} = \mathbf{W}^\top,\ \mathbf{W}\mathbf{1}_n = \mathbf{1}_n,\ w_{ij} \geq 0,\ \forall i, j,$$

and its eigenvalues satisfy $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$ and $\rho := \max\{|\lambda_2|, |\lambda_n|\} < 1$.

Assumption 2. *All functions $\{F_i\}_{1\leq i\leq n}$ have Lipschitz continuous gradients with Lipschitz constants $L_{\nabla F_i}$, respectively. Therefore, ∇F is $L_{\nabla F}$-Lipchitz continous with $L_{\nabla F} = \max_{1\leq i\leq n}\{L_{\nabla F_i}\}$.*

Assumption 3. *The function $\Psi : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ is a closed proper convex function.*

For stochastic oracles, we assume that each node i at every iteration k is able to obtain a local random data vector ξ_i^k. The induced natural filtration is given by $\mathscr{F}_0 = \{\emptyset, \Omega\}$ and

$$\mathscr{F}_k := \sigma\left(\xi_i^t \mid i = 1, \ldots, n,\ t = 1, \ldots, k\right), \forall k \geq 1.$$

Table 1: Comparison of decentralized proximal gradient based algorithms to find an ϵ-stationary solution to stochastic composite optimization in the nonconvex setting. The sample complexity is defined as the number of required samples per agent to obtain an ϵ-stationary point (see Definition 1). We omit a comparison with SPPDM [Wang et al., 2021] as their definition of stationarity differs from ours; see Appendix for further discussions.

Algorithm	Batch Size	Sample Complexity	Communication Complexity	Linear Speedup?	Remark
ProxGT-SA [Xin et al., 2021a]	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(n^{-1}\epsilon^{-2})$	$\mathcal{O}(\log(n)\epsilon^{-1})$	✓	
ProxGT-SR-O [Xin et al., 2021a]	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(n^{-1}\epsilon^{-1.5})$	$\mathcal{O}(\log(n)\epsilon^{-1})$	✓	double-loop; mean-squared smoothness
DEEPSTORM [Mancino-ball et al., 2023]	$\mathcal{O}(\epsilon^{-0.5})$ then $\mathcal{O}(1)^*$	$\mathcal{O}(n^{-1}\epsilon^{-1.5})$	$\mathcal{O}(n^{-1}\epsilon^{-1.5})$	✓	two time-scale; mean-squared smoothness; double gradient evaluations per iteration
	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-1.5}\lvert\log\epsilon\rvert^{-1.5})$	$\mathcal{O}(\epsilon^{-1.5}\lvert\log\epsilon\rvert^{-1.5})$	✗	
Prox-DASA (Alg. 1)	$\mathcal{O}(1)$	$\mathcal{O}(n^{-1}\epsilon^{-2})$	$\mathcal{O}(n^{-1}\epsilon^{-2})$	✓	bounded heterogeneity
Prox-DASA-GT (Alg. 2)	$\mathcal{O}(1)$	$\mathcal{O}(n^{-1}\epsilon^{-2})$	$\mathcal{O}(n^{-1}\epsilon^{-2})$	✓	

* It requires $\mathcal{O}(\epsilon^{-0.5})$ batch size in the first iteration and then $\mathcal{O}(1)$ for the rest (see m_0 in Algorithm 1 in Mancino-ball et al. [2023]).

We require that the stochastic gradient $\nabla G_i(\cdot, \xi_i^{k+1})$ is unbiased conditioned on the filteration $\mathscr{F}_k$.

Assumption 4 (Unbiasness)**.** *For any $k \geq 0, x \in \mathscr{F}_k$, and $1 \leq i \leq n$, $\mathbb{E}\left[\nabla G_i(x, \xi_i^{k+1}) \mid \mathscr{F}_k\right] = \nabla F_i(x)$.*

Assumption 5 (Independence)**.** *For any $k \geq 0, 1 \leq i, j \leq n, i \neq j$, ξ_i^{k+1} is independent of $\mathscr{F}_k$, and ξ_i^{k+1} is independent of ξ_j^{k+1}.*

In addition, we consider two standard assumptions on the variance and heterogeneity of stochastic gradients.

Assumption 6 (Bounded variance)**.** *For any $k \geq 0, x \in \mathscr{F}_k$, and $1 \leq i \leq n$,*

$$\mathbb{E}\left[\left\|\nabla G_i(x, \xi_i^{k+1}) - \nabla F_i(x)\right\|^2 \middle| \mathscr{F}_k\right] \leq \sigma_i^2.$$

Let $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2$.

Assumption 7 (Gradient heterogeneity)**.** *There exists a constant $\nu \geq 0$ such that for all $1 \leq i \leq n, x \in \mathbb{R}^d$,*

$$\|\nabla F_i(x) - \nabla F(x)\| \leq \nu.$$

Remark (Bounded gradient heterogeneity). The above assumption of gradient heterogeneity is standard Lian et al. [2017] and less strict than the bounded second moment assumption on stochastic gradients which implies lipschtizness of functions $\{F_i\}$. However, this assumption is only required for the convergence analysis of Prox-DASA and can be bypassed by employing a gradient tracking step.

Remark (Smoothness and mean-squared smoothness). Our theoretical results of the proposed methods are only built on the smoothness assumption on functions $\{F_i\}$ without further assuming mean-squared smoothness assumptions on $\{G_{i,\xi}\}$, which is required in all variance reduction based methods in the literature, such as ProxGT-SR-O [Xin et al., 2021a] and DEEPSTORM [Mancino-ball et al., 2023]. It is worth noting that a clear distinction in the lower bounds of sample complexity for solving stochastic optimization under two different sets of assumptions has been proven in [Arjevani et al., 2023]. Specifically, when considering the mean-squared smoothness assumption, the optimal sample complexity is $\mathcal{O}(\epsilon^{-1.5})$, whereas under smoothness assumptions, it is $\mathcal{O}(\epsilon^{-2})$. The proposed methods in this work achieve the optimal sample complexity under our weaker assumptions.

3 ALGORITHM

Several algorithms have been developed to solve Problem (1) in the stochastic setting; see Table 1. However, the most recent two types of algorithms have certain drawbacks: (i) **increasing batch sizes**: ProxGT-SA, Prox-SR-O, and DEEPSTORM with constant step sizes (Theorem 1 in [Mancino-ball et al., 2023]) require batches of stochastic gradients with batch sizes inversely proportional to tolerance ϵ; (ii) **algorithmic complexities**: ProxGT-SR-O and DEEPSTORM are either double-looped or two-time-scale, and require stochastic gradients evaluated at different parameter values over the same sample, i.e., $\nabla G_i(x, \xi)$ and $\nabla G_i(x', \xi)$. These variance reduction techniques are unfavorable when gradient evaluations are computationally expensive such as forward-backward steps for deep neural networks. (iii) **theoretical weakness**: the convergence analyses of ProxGT-SR-O and DEEPSTORM are established under the *stronger* assumption of mean-squared lipschtizness of stochastic gradients. In addition, Theorem 2 in [Mancino-ball et al., 2023] fails to provide linear-speedup results for one-sample variant of DEEPSTORM with diminishing stepsizes.

Algorithm 1: `Prox-DASA`

Input: $x_i^0 = z_i^0 = \mathbf{0}, \gamma, \{\alpha_k\}_{\geq 0}, m$
for $k = 0, 1, \ldots, K-1$ **do**
  `# Local Update`
  for $i = 1, 2, \ldots, n$ *(in parallel)* **do**
    $y_i^k = \mathbf{prox}_\Psi^\gamma \left(x_i^k - \gamma z_i^k\right)$
    $\tilde{x}_i^{k+1} = (1-\alpha_k)x_i^k + \alpha_k y_i^k$
    `# Compute stochastic gradient`
    $v_i^{k+1} = \nabla G_i(x_i^k, \xi_i^{k+1})$
    $\tilde{z}_i^{k+1} = (1-\alpha_k)z_i^k + \alpha_k v_i^{k+1}$
  end
  `# Communication`
  $[x_1^{k+1}, \ldots, x_n^{k+1}] = [\tilde{x}_1^{k+1}, \ldots, \tilde{x}_n^{k+1}]\mathbf{W}^m$
  $[z_1^{k+1}, \ldots, z_n^{k+1}] = [\tilde{z}_1^{k+1}, \ldots, \tilde{z}_n^{k+1}]\mathbf{W}^m$
end

3.1 DECENTRALIZED PROXIMAL AVERAGED STOCHASTIC APPROXIMATION

To address the above limitations, we propose **D**ecentralized **Prox**imal **A**veraged **S**tochastic **A**pproximation (`Prox-DASA`) which leverages a common averaging technique in stochastic optimation [Ruszczyński, 2008, Mokhtari et al., 2018, Ghadimi et al., 2020] to reduce the error of gradient estimation. In particular, the sequences of dual variables $\mathbf{Z}^k = [z_1^k, \ldots, z_n^k]$ that aim to approximate gradients are defined in the following recursion:

$$\mathbf{Z}^{k+1} = \left\{(1-\alpha_k)\mathbf{Z}^k + \alpha_k \mathbf{V}^{k+1}\right\}\mathbf{W}^m$$
$$\mathbf{V}^{k+1} = [v_1^{k+1}, \ldots, v_n^{k+1}],$$

where each v_i^{k+1} is the local stochastic gradient evaluated at the local variable x_i^k. For complete graphs where each pair of graph vertices is connected by an edge and there is no consensus error for optimization variables, i.e., $\mathbf{W} = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ and $x_i^k = x_j^k, \forall i, j$, the averaged dual variable over nodes $\bar{z}^k$ follows the same averaging rule as in centralized algorithms:

$$\bar{z}^{k+1} = (1-\alpha_k)\bar{z}^k + \alpha_k \bar{v}^{k+1}$$
$$\mathbb{E}[\bar{v}^{k+1}|\mathscr{F}_k] = \nabla F(\bar{x}^k).$$

To further control the consensus errors, we employ a multiple consensus step for both primal and dual iterates $\{x_i^k, z_i^k\}$ which multiply the matrix of variables from all nodes by the weight matrix m times. A pseudo code of `Prox-DASA` is given in Algorithm 1.

3.2 GRADIENT TRACKING

The constant ν defined in Assumption 7 measures the heterogeneity between local gradients and global gradients, and hence the variance of datasets of different agents. To remove ν in the complexity bound, Tang et al. [2018] proposed the D^2 algorithm, which modifies the x update in D-PSGD [Lian et al., 2017]. However, it requires one additional assumption on the eigenvalues of the mixing matrix $\mathbf{W}$. Here we adopt the gradient tracking technique, which was first introduced to deterministic distributed optimization to improve the convergence rate [Xu et al., 2015, Di Lorenzo and Scutari, 2016, Nedic et al., 2017, Qu and Li, 2017], and was later proved to be useful in removing the data variance (i.e., ν) dependency in the stochastic case [Zhang and You, 2019, Lu et al., 2019, Pu and Nedić, 2021, Koloskova et al., 2021]. In the convergence analysis of `Prox-DASA`, an essential step is to control the heterogeneity of stochastic gradients, i.e., $\mathbb{E}[\left\|\mathbf{V}^{k+1} - \bar{\mathbf{V}}^{k+1}\right\|^2]$, which requires bounded heterogeneity of local gradients (Assumption 7). To pypass this assumption, we employ a gradient tracking step by replacing $\mathbf{V}^{k+1}$ with pseudo stochastic gradients $\mathbf{U}^{k+1} = [u_1^{k+1}, \ldots, u_n^{k+1}]$, which is updated as follows:

$$\mathbf{U}^{k+1} = \left(\mathbf{U}^k + \mathbf{V}^{k+1} - \mathbf{V}^k\right)\mathbf{W}^m.$$

Provided that $\mathbf{U}^0 = \mathbf{V}^0$ and $\mathbf{W}\mathbf{1} = \mathbf{1}$, one can show that $\bar{u}^k = \bar{v}^k$ at each step k. In addition, with the consensus procedure over $\mathbf{U}^k$, the heterogeneity of pseudo stochastic gradients $\mathbb{E}[\left\|\mathbf{U}^{k+1} - \bar{\mathbf{U}}^{k+1}\right\|^2]$ can be bounded above. The proposed algorithm, named as `Prox-DASA` with Gradient Tracking (`Prox-DASA-GT`), is presented in Algorithm 2.

Algorithm 2: `Prox-DASA-GT`

Input: $x_i^0 = z_i^0 = u_i^0 = \mathbf{0}, \gamma, \{\alpha_k\}_{\geq 0}, m$
for $k = 0, 1, \ldots, K$ **do**
  `# Local Update`
  for $i = 1, 2, \ldots, n$ *(in parallel)* **do**
    $y_i^k = \mathbf{prox}_\Psi^\gamma \left(x_i^k - \gamma z_i^k\right)$
    $\tilde{x}_i^{k+1} = (1-\alpha_k)x_i^k + \alpha_k y_i^k$
    `# Compute stochastic gradient`
    $v_i^{k+1} = \nabla G_i(x_i^k, \xi_i^{k+1})$
    $\tilde{u}_i^{k+1} = u_i^k + v_i^{k+1} - v_i^k$
    $\tilde{z}_i^{k+1} = (1-\alpha_k)z_i^k + \alpha_k \tilde{u}_i^{k+1}$
  end
  `# Communication`
  $[x_1^{k+1}, \ldots, x_n^{k+1}] = [\tilde{x}_1^{k+1}, \ldots, \tilde{x}_n^{k+1}]\mathbf{W}^m$
  $[u_1^{k+1}, \ldots, u_n^{k+1}] = [\tilde{u}_1^{k+1}, \ldots, \tilde{u}_n^{k+1}]\mathbf{W}^m$
  $[z_1^{k+1}, \ldots, z_n^{k+1}] = [\tilde{z}_1^{k+1}, \ldots, \tilde{z}_n^{k+1}]\mathbf{W}^m$
end

3.3 CONSENSUS ALGORITHM

In practice, we can leverage accelerated consensus algorithms, e.g., Liu and Morse [2011], Olshevsky [2017], to speed up the multiple consensus step $\mathbf{W}^m$ to achieve improved communication complexities when $m > 1$. Specifically, we can replace $\mathbf{W}^m$ by a Chebyshev-type polynomial of $\mathbf{W}$ as described in Algorithm 3, which can improve the ρ-dependency of the communication complexity from a factor of $\frac{1}{1-\rho}$ to $\frac{1}{\sqrt{1-\rho}}$.

Algorithm 3: Chebyshev Mixing Protocol

Input: Matrix $\mathbf{X}$, mixing matrix $\mathbf{W}$, rounds m
Set $\mathbf{A}_0 = \mathbf{X}, \mathbf{A}_1 = \mathbf{X}\mathbf{W}, \rho = \max\{|\lambda_2(\mathbf{W})|, |\lambda_n(\mathbf{W})|\} < 1, \mu_0 = 1, \mu_1 = \frac{1}{\rho}$
for $t = 1, \ldots, m-1$ **do**
  $\mu_{t+1} = \frac{2}{\rho}\mu_t - \mu_{t-1}$
  $\mathbf{A}_{t+1} = \frac{2\mu_t}{\rho\mu_{t+1}}\mathbf{A}_t\mathbf{W} - \frac{\mu_{t-1}}{\mu_{t+1}}\mathbf{A}_{t-1}$
end
Output: $\mathbf{A}_m$

4 CONVERGENCE ANALYSIS

4.1 NOTION OF STATIONARITY

For centralized optimization problems with non-convex objective function $F(x)$, a standard measure of non-stationarity of a point $\bar{x}$ is the squared norm of proximal gradient mapping of $\nabla F(\bar{x})$ at $\bar{x}$, i.e.,

$$\|\mathcal{G}(\bar{x}, \nabla F(\bar{x}), \gamma)\|^2 = \left\| \frac{1}{\gamma}\left(x - \mathbf{prox}_{\Psi}^{\gamma}(\bar{x} - \gamma\nabla F(\bar{x}))\right)\right\|^2.$$

For the smooth case where $\Psi(x) \equiv 0$, the above measure is reduced to $\|\nabla F(\bar{x})\|^2$.

However, in the decentralized setting with a connected network $\mathbb{G}$, we solve the following equivalent reformulated consensus optimization problem:

$$\begin{aligned} \min_{x_1,\ldots,x_n \in \mathbb{R}^d} \quad & \frac{1}{n}\sum_{i=1}^{n}\{F_i(x_i) + \Psi(x_i)\} \\ \text{s.t.} \quad & x_i = x_j,\ \forall (i,j). \end{aligned} \tag{2}$$

To measure the non-stationarity in Problem (2), one should consider not only the stationarity violation at each node but also the consensus errors over the network. Therefore, Xin et al. [2021a] and Mancino-ball et al. [2023] define an ϵ-stationary point $\mathbf{X} = [x_1, \ldots, x_n]$ of Problem 2 as

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left\{\|\mathcal{G}(x_i, \nabla F(x_i), \gamma)\|^2 + L_{\nabla F}^2\|x_i - \bar{x}\|^2\right\}\right] \le \epsilon. \tag{3}$$

In this work, we use a general measure as follows.

Definition 1. *Let $\mathbf{X} = [x_1, \ldots, x_n]$ be random vectors generated by a decentralized algorithm to solve Problem 2 and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. We say that $\mathbf{X}$ is an ϵ-stationary point of Problem 2 if*

$$\textit{(stationarity violation)} \quad \mathbb{E}\left[\|\mathcal{G}(\bar{x}, \nabla F(\bar{x}), \gamma)\|^2\right] \le \epsilon,$$
$$\textit{(consensus error)} \quad \mathbb{E}\left[\frac{L_{\nabla F}^2}{n}\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|^2\right] \le \epsilon.$$

The next inequality characterizes the difference between the gradient mapping at $\bar{x}$ and x_i, which relates our definition to (3). Noting that by non-expansiveness of the proximal operator, we have $\|\mathcal{G}(x_i, \nabla F(x_i), \gamma) - \mathcal{G}(\bar{x}, \nabla F(\bar{x}), \gamma)\| \le \frac{2+\gamma L_{\nabla F}}{\gamma}\|x_i - \bar{x}\|$, implying

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\|\mathcal{G}(x_i, \nabla F(x_i), \gamma)\|^2 \\ &\lesssim \|\mathcal{G}(\bar{x}, \nabla F(\bar{x}), \gamma)\|^2 + \tfrac{1}{\gamma^2 n}\left\|\mathbf{X} - \bar{\mathbf{X}}\right\|^2. \end{aligned}$$

4.2 MAIN RESULTS

We present the complexity results of our algorithms below.

Theorem 1. *Suppose Assumptions 1, 2, 3, 4, 5, 6 hold and the total number of iterations $K \ge K_0$, where K_0 is a constant that only depends on constants $(n, L_{\nabla F}, \varrho(m), \gamma)$, where $\varrho(m) = \frac{(1+\rho^{2m})\rho^{2m}}{(1-\rho^{2m})^2}$. Let C_0 be some initialization-dependent constant and R be a random integer uniformly distributed over $\{1, 2, \ldots, K\}$. Suppose we set $\alpha_k \asymp \sqrt{\frac{n}{K}}, \gamma \asymp \frac{1}{L_{\nabla F}}$.*

***(Prox-DASA)** Suppose in addition Assumption 7 also holds. The, for Algorithm 1 we have*

$$\begin{aligned} &\mathbb{E}\left[\left\|\mathcal{G}(\bar{x}^R, \nabla F(\bar{x}^R), \gamma)\right\|^2\right] \\ &\quad\lesssim \frac{\gamma^{-1}C_0 + \sigma^2}{\sqrt{nK}} + \frac{n(\sigma^2 + \gamma^{-2}\nu^2)\varrho(m)}{K}, \\ &\mathbb{E}\left[\left\|\bar{z}^R - \nabla F(\bar{x}^R))\right\|^2\right] \\ &\quad\lesssim \frac{\gamma^{-1}C_0 + \sigma^2}{\sqrt{nK}} + \frac{n(\sigma^2 + \gamma^{-2}\nu^2)\varrho(m)}{K}, \\ &\mathbb{E}\left[\frac{L_{\nabla F}^2}{n}\left\|\mathbf{X}_R - \bar{\mathbf{X}}_R\right\|^2 + \frac{1}{n}\left\|\mathbf{Z}_R - \bar{\mathbf{Z}}_R\right\|^2\right] \\ &\quad\lesssim \frac{n(\sigma^2 + \gamma^{-2}\nu^2)\varrho(m)}{K}. \end{aligned}$$

***(Prox-DASA-GT)** For Algorithm 2 we have*

$$\begin{aligned} &\mathbb{E}\left[\left\|\mathcal{G}(\bar{x}^R, \nabla F(\bar{x}^R), \gamma)\right\|^2\right] \lesssim \frac{\gamma^{-1}C_0 + \sigma^2}{\sqrt{nK}} + \frac{n\sigma^2\varrho(m)}{K}, \\ &\mathbb{E}\left[\left\|\bar{z}^R - \nabla F(\bar{x}^R)\right\|^2\right] \lesssim \frac{\gamma^{-1}C_0 + \sigma^2}{\sqrt{nK}} + \frac{n\sigma^2\varrho(m)}{K}, \\ &\mathbb{E}\left[\frac{L_{\nabla F}^2}{n}\left\|\mathbf{X}_R - \bar{\mathbf{X}}_R\right\|^2 + \frac{1}{n}\left\|\mathbf{Z}_R - \bar{\mathbf{Z}}_R\right\|^2\right] \lesssim \frac{n\sigma^2\varrho(m)}{K}. \end{aligned}$$

In Theorem 1 for simplicity we assume $\gamma \asymp \frac{1}{L_{\nabla F}}$, which can be relaxed to $\gamma > 0$. We have the following corollary characterizing the complexity of Algorithm 1 and 2 for finding ϵ-stationary points. The proof is immediate.

Corollary 1. *Under the same conditions of Theorem 1, provided that $K \gtrsim n^3\varrho(m)$, for any $\epsilon > 0$ the sample complexity per agent for finding ϵ-stationary points in Algorithm 1 and 2 are $\mathcal{O}(\max\{n^{-1}\epsilon^{-2}, K_T\})$ where the transient time $K_T \asymp \max\{K_0, n^3\varrho(m)\}$.*

Remark (Sample complexity). For a sufficiently small $\epsilon > 0$, Corrollary 1 implies that the sample complexity of Algorithm 1 and 2 matches the optimal lower bound $\mathcal{O}(n^{-1}\epsilon^{-2})$ in decentralized *smooth* stochastic non-convex optimization [Lu and De Sa, 2021].

Remark (Transient time and communication complexity). Our algorithms can achieve convergence with a single communication round per iteration, i.e., $m = 1$, leading to a topology-independent $\mathcal{O}(n^{-1}\epsilon^{-2})$ communication complexity. In this case, however, the transient time K_T still depends on ρ, as is also the case for smooth optimization problems [Xin et al., 2021b]. When considering multiple consensus steps per iteration with the communication complexity being $\mathcal{O}(mn^{-1}\epsilon^{-2})$, setting $m \asymp \lceil \frac{1}{1-\rho} \rceil$ (or $m \asymp \lceil \frac{1}{\sqrt{1-\rho}} \rceil$ for accelerated consensus algorithms) results in a topology-independent transient time given that $\varrho(m) \asymp 1$.

Remark (Dual convergence). An important aspect to emphasize is that in our proposed methods, the sequence of average dual variables $\bar{z}^k = \frac{1}{n}\sum_{i=1}^n z_i^k$ converges to $\nabla F(\bar{x}^k)$, while the consensus error of $\{z_1^k, \ldots, z_n^k\}$ decreases to zero. Our approach achieves this desirable property, which is commonly observed in modern variance reduction methods [Gower et al., 2020], without the need for complex variance reduction operations in each iteration. As a result, it provides a reliable termination criterion in the stochastic setting without requiring large batch sizes.

4.3 PROOF SKETCH

Here, we present a sketch of our convergence analyses and defer details to Appendix. Our proof relies on the merit function below:

$$W(\bar{x}^k, \bar{z}^k) = \underbrace{\Phi(\bar{x}^k) - \Phi_*}_{\text{function value gap}} + \underbrace{\Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k)}_{\text{primal convergence}} + \lambda \underbrace{\left\|\nabla F(\bar{x}^k) - \bar{z}^k\right\|^2}_{\text{dual convergence}},$$

where $\eta(x, z) = \min_{y \in \mathbb{R}^d} \left\{ \langle z, y - x\rangle + \frac{1}{2\gamma}\|y - x\|^2 + \Psi(y) \right\}$. Let $y_+^k := \mathbf{prox}_\Psi^\gamma\left(\bar{x}^k - \gamma \bar{z}^k\right)$. Then, the proximal gradient mapping of $\bar{z}^k$ at $\bar{x}^k$ is $\mathcal{G}(\bar{x}^k, \bar{z}^k, \gamma) = \frac{1}{\gamma}(\bar{x}^k - y_+^k)$. Since y_+^k is the minimizer of a $1/\gamma$-strongly convex function, we have

$$\left\langle \bar{z}^k, y_+^k - \bar{x}^k \right\rangle + \frac{1}{2\gamma}\|y_+^k - \bar{x}^k\|^2 + \Psi(y_+^k) \le \Psi(\bar{x}^k) - \frac{1}{2\gamma}\|y_+^k - \bar{x}^k\|^2,$$

implying the relation between $\Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k)$ and primal convergence:

$$\Psi(\bar{x}^k) - \eta(\bar{x}^k, \bar{z}^k) \ge \frac{\gamma}{2}\left\|\mathcal{G}(\bar{x}^k, \bar{z}^k, \gamma)\right\|^2.$$

Following standard practices in optimization, we set $\gamma = \frac{1}{L_{\nabla F}}$ below for simplicity. However, our algorithms do not require any restriction on the choice of γ.

Step 1: Leveraging the merit function with $\lambda \asymp \gamma$, we can first obtain an essential lemma (Lemma 11 in Appendix) in our analyses, which says that for sequences $\{x_i^k, z_i^k\}_{1 \le i \le n, k \ge 0}$ generated by `Prox-DASA(-GT)` (Algorithm 1 or 2) with $\alpha_k \lesssim \min\{1, (1+\gamma)^{-2}, \gamma^2(1+\gamma)^{-2}\}$, we have

$$W(\bar{x}^{k+1}, \bar{z}^{k+1}) - W(\bar{x}^k, \bar{z}^k) \le -\alpha_k\left\{\Theta^k + \Upsilon^k + \alpha_k \Lambda^k + r^{k+1}\right\},$$

where $\mathbb{E}[r^{k+1} \mid \mathscr{F}_k] = 0$, $\Lambda^k \asymp \gamma\left\|\bar{\Delta}^{k+1}\right\|^2$,

$$\Theta^k \asymp \frac{1}{\gamma}\|\bar{x}^k - \bar{y}^k\|^2 + \gamma\left\|\nabla F(\bar{x}^k) - \bar{z}^k\right\|^2,$$

$$\Upsilon^k \asymp \frac{\gamma}{n}\left\|\mathbf{Z}_k - \bar{\mathbf{Z}}_k\right\|^2 + \frac{1}{n\gamma}\left\|\mathbf{X}_k - \bar{\mathbf{X}}_k\right\|^2,$$

and $\bar{\Delta}^{k+1} = \bar{v}^{k+1} - \frac{1}{n}\sum_{i=1}^n \nabla F_i(x_i^k) = \bar{u}^{k+1} - \frac{1}{n}\sum_{i=1}^n \nabla F_i(x_i^k)$ (for `Prox-DASA-GT`). Thus, by telescoping and taking expectation with respect to $\mathscr{F}_0$, we have

$$\begin{aligned}
&\sum_{k=0}^K \alpha_k \mathbb{E}\left[\left\|\bar{x}^k - \bar{y}^k\right\|^2 + \gamma^2\left\|\nabla F(\bar{x}^k) - \bar{z}^k\right\|^2\right] \\
&\lesssim \gamma W(\bar{x}^0, \bar{z}^0) + \gamma^2\sigma^2\boxed{\sum_{k=0}^K \frac{\alpha_k^2}{n}} \\
&\quad + \sum_{k=0}^K \frac{\alpha_k\left\{\mathbb{E}\left[\left\|\mathbf{X}_k - \bar{\mathbf{X}}_k\right\|^2 + \gamma^2\left\|\mathbf{Z}_k - \bar{\mathbf{Z}}_k\right\|^2\right]\right\}}{n}.
\end{aligned} \tag{4}$$

Step 2: We then analyze the consensus errors. Without loss of generality, we consider $\mathbf{X}_0 = \bar{\mathbf{X}}_0 = \mathbf{0}$, i.e., all nodes have the same initialization at $\mathbf{0}$. For $m \in \mathbb{N}_+$, define

$$\varrho(m) = \frac{(1+\rho^{2m})\rho^{2m}}{(1-\rho^{2m})^2}.$$

Then, we have the following fact:

- $\varrho(m)$ is monotonically decreasing with the maximum value being $\varrho(1) = \frac{(1+\rho^2)\rho^2}{(1-\rho^2)^2} := \varrho_1$;
- $\varrho(m) \le 1$ if and only if $\rho^{2m} \le \frac{1}{3}$.

With the definition of $\varrho(m)$ and assuming $0 < \alpha_{k+1} \le \alpha_k \le 1$, we can show the consensus errors have the following upper bounds.

`Prox-DASA`: Let $\alpha_k \lesssim \varrho(m)^{-\frac{1}{2}}$, we have

$$\sum_{k=0}^K \frac{\alpha_k}{n}\mathbb{E}\left[\left\|\mathbf{X}_k - \bar{\mathbf{X}}_k\right\|^2\right] \le \sum_{k=0}^K \frac{\gamma^2\alpha_k}{n}\mathbb{E}\left[\left\|\mathbf{Z}_k - \bar{\mathbf{Z}}_k\right\|^2\right]$$

$$\lesssim (\gamma^2\sigma^2+\nu^2)\varrho(m)\boxed{\sum_{k=0}^{K}\alpha_k^3}. \tag{5}$$

`Prox-DASA-GT`: Let $\alpha_k \lesssim \min\{\varrho(m)^{-1}, \varrho(m)^{-\frac{1}{2}}\}$, we have

$$\sum_{k=0}^{K}\frac{\alpha_k}{n}\mathbb{E}\left[\|\mathbf{X}_k-\bar{\mathbf{X}}_k\|^2\right] \le \sum_{k=0}^{K}\frac{\gamma^2\alpha_k}{n}\mathbb{E}\left[\|\mathbf{Z}_k-\bar{\mathbf{Z}}_k\|^2\right]$$
$$\lesssim \varrho(m)^2\boxed{\sum_{k=0}^{K}\alpha_k^3}\left\{\gamma^2\sigma^2+\alpha_k^2\mathbb{E}\left[\|\bar{x}^k-\bar{y}^k\|^2\right]\right\}. \tag{6}$$

We can also see that to obtain a topology-independent iteration complexity, the number of communication rounds can be set as $m=\lceil\frac{\log 3}{2(1-\rho)}\rceil$, which implies $\varrho(m)\le 1$.

In addition, we have the following fact that relates the consensus error of $\mathbf{Y}$ to the consensus errors of $\mathbf{X}$ and $\mathbf{Z}$:

$$\left\|y_+^k-\bar{y}^k\right\|^2+\frac{1}{n}\left\|\mathbf{Y}_k-\bar{\mathbf{Y}}_k\right\|^2=\frac{1}{n}\sum_{i=1}^{n}\left\|y_i^k-y_+^k\right\|^2$$
$$\le \frac{2}{n}\left\{\|\mathbf{X}_k-\bar{\mathbf{X}}_k\|^2+\gamma^2\|\mathbf{Z}_k-\bar{\mathbf{Z}}_k\|^2\right\}.$$

Step 3: Let R be a random integer with

$$\Pr(R=k)=\frac{\alpha_k}{\sum_{k=1}^{K}\alpha_k},\quad k=1,2,\ldots,K,$$

and dividing both sides of (5) by $\sum_{k=1}^{K}\alpha_k$, we can obtain that for `Prox-DASA`, the consensus error of $\mathbf{X}_R$ satisfies

$$\mathbb{E}\left[\frac{1}{n}\left\|\mathbf{X}_R-\bar{\mathbf{X}}_R\right\|^2\right]\lesssim(\gamma^2\sigma^2+\nu^2)\varrho(m)\frac{\sum_{k=0}^{K}\alpha_k^3}{\sum_{k=1}^{K}\alpha_k}.$$

Moreover, noting that

$$\|\mathcal{G}(\bar{x},\nabla F(\bar{x}),\gamma)\|^2\lesssim\frac{1}{\gamma^2}\left\{\left\|\bar{x}^k-\bar{y}^k\right\|^2+\left\|y_+^k-\bar{y}^k\right\|^2\right\}$$
$$+\left\|\nabla F(\bar{x}^k)-\bar{z}^k\right\|^2,$$

and combining (4) with (5), we can get

$$\mathbb{E}\left[\left\|\mathcal{G}(\bar{x}^R,\nabla F(\bar{x}^R),\gamma)\right\|^2\right]\lesssim \underbrace{\frac{W(\bar{x}^0,\bar{z}^0)}{\gamma\sum_{k=1}^{K}\alpha_k}}_{\text{initialization-related term}}$$
$$+\underbrace{\boxed{\sigma^2\frac{\sum_{k=0}^{K}\alpha_k^2}{n\sum_{k=1}^{K}\alpha_k}}}_{\text{variance-related term}}+\underbrace{\boxed{(\sigma^2+\gamma^{-2}\nu^2)\varrho(m)\frac{\sum_{k=0}^{K}\alpha_k^3}{\sum_{k=1}^{K}\alpha_k}}}_{\text{consensus error}}.$$

Thus, setting $\alpha_k \asymp \sqrt{\frac{n}{K}}$, we obtain the convergence results of `Prox-DASA`:

$$\mathbb{E}\left[\left\|\mathcal{G}(\bar{x}^R,\nabla F(\bar{x}^R),\gamma)\right\|^2\right]$$
$$\lesssim\frac{\gamma^{-1}W(\bar{x}^0,\bar{z}^0)+\sigma^2}{\sqrt{nK}}+\frac{n(\sigma^2+\gamma^{-2}\nu^2)\varrho(m)}{K},$$
$$\mathbb{E}\left[\frac{1}{\gamma^2 n}\left\|\mathbf{X}_R-\bar{\mathbf{X}}_R\right\|^2\right]\lesssim\frac{n(\sigma^2+\gamma^{-2}\nu^2)\varrho(m)}{K}.$$

For `Prox-DASA-GT`, we can complete the proof with similar arguments by combining (6) with (4) and noting that $\varrho(m)^2\alpha_k^4\lesssim 1$.

5 EXPERIMENTS

5.1 SYNTHETIC DATA

To demonstrate the effectiveness of our algorithms, we first evaluate our algorithms using synthetic data for solving sparse single index models [Alquier and Biau, 2013] in the decentralized setting. We consider the homogeneous setting where the data sample at each node $\xi=(X,Y)$ is generated from the same single index model $Y=g(X^\top\theta_*)+\varepsilon$, where $X,\theta\in\mathbb{R}^d$ and $\mathbb{E}[\varepsilon|X]=0$. In this case, we solve the following L_1-regularized least square problems:

$$\min_{\theta\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n}\mathop{\mathbb{E}}_{(X,Y)\sim\mathcal{D}}\left[(Y-g(X^\top\theta))^2\right]+\lambda\|\theta\|_1$$

In particular, we set $\theta_*\in\mathbb{R}^{100}$ to be a sparse vector and $g(\cdot)=(\cdot)^2$ which corresponds to the sparse phase retrieval problem [Jaganathan et al., 2016]. We simulate streaming data samples with batch size $=1$ for training and 10,000 data samples per node for evaluations, where X and ϵ are sampled independently from two Gaussian distributions. We employ a ring topology for the network where self-weighting and neighbor weights are set to be $1/3$. We set the penalty parameter $\lambda=0.01$, the total number of iterations $K=10,000$, $\alpha_k=\sqrt{n/K}$, $\gamma=0.01$, and the number of communication rounds per iteration $m=\lceil\frac{1}{1-\rho}\rceil$. We plot the test loss and the norm of proximal gradient mapping in the log scale against the number of iterations in Figure 1, which shows that our decentralized algorithms have an additional linear speed-up with respect to n. In other words, the algorithms become faster as more agents are added to the network.

5.2 REAL-WORLD DATA

Following Mancino-ball et al. [2023], we consider solving the classification problem

$$\min_{\theta\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathcal{D}_i|}\sum_{(x,y)\in\mathcal{D}_i}\ell_i(f(x;\theta),y)+\lambda\|\theta\|_1, \tag{7}$$

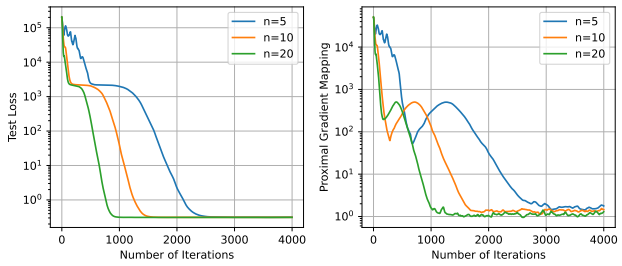

Figure 1: Linear-speedup performance of `Prox-DASA` for decentralized online sparse phase retrievel problems. (`Prox-DASA-GT` has relatively the same plots)

on a9a and MNIST datasets[1]. Here, ℓ_i denotes the cross-entropy loss, and f represents a neural network parameterized by θ with x being its input. $\mathcal{D}_i$ is the training set only available to agent i. The L_1 regularization term is used to impose a sparsity structure on the neural network. We use the code in Mancino-ball et al. [2023] for `SPPDM`, `ProxGT-SR-O/E`, `DEEPSTORM`, and then implement `Prox-DASA` and `Prox-DASA-GT` under their framework, which mainly utilizes PyTorch [Paszke et al., 2019] and mpi4py [Dalcin and Fang, 2021]. We use a 2-layer perception model on a9a and the LeNet architecture [LeCun et al., 2015] for the MNIST dataset. We have 8 agents ($n = 8$) which connect in the form of a ring for a9a and a random graph for MNIST. To demonstrate the performance of our algorithms in the constant batch size setting, the batch size is chosen to be 4 for a9a and 32 for MNIST for all algorithms. The learning rates provided in the code of Mancino-ball et al. [2023] are adjusted accordingly, and we select the ones with the best performance. For `Prox-DASA` and `Prox-DASA-GT` we choose a diminishing stepsize sequence, namely, $\alpha_k = \min\left\{\alpha\sqrt{\frac{n}{k}}, 1\right\}$ for all $k \geq 0$. Note that the same complexity (up to logarithmic factors) bounds can be obtained by directly plugging in the aforementioned expressions for α_k in Section 4.3. Then we tune $\gamma \in \{1, 3, 10\}$ and $\alpha \in \{0.3, 1.0, 3.0\}$. The penalty parameter λ is chosen to be 0.0001 for all experiments. The number of communication rounds per iteration m is set to be 1 for all algorithms. We evaluate the model performance periodically during training and then plot the results in Figure 2, from which we observe that both `Prox-DASA` and `Prox-DASA-GT` have considerably good performance with small variance in terms of test accuracy, training loss, and stationarity. In particular, it should be noted that although `DEEPSTORM` achieves better stationarity in Figure 2(l) and 2(i), training a neural network by using `DEEPSTORM` takes longer time than `Prox-DASA` and `Prox-DASA-GT` since it uses the momentum-based variance reduction technique, which requires **two forward-backward passes** (see, e.g., Eq. (10) and Algorithm 1 in Mancino-ball et al. [2023]) to compute the gradients in one iteration per agent. In contrast, ours only require **one**, which saves a large amount of time (see Table 1 in Appendix). We include further details of our experiments in the Appendix.

[1] Available at https://www.openml.org.

6 CONCLUSION

In this work, we propose and analyze a class of single time-scale decentralized proximal algorithms (`Prox-DASA-(GT)`) for non-convex stochastic composite optimization in the form of (1). We show that our algorithms achieve linear speed-up with respect to the number of agents using an $\mathcal{O}(1)$ batch size per iteration under mild assumptions. Furthermore, we demonstrate the efficiency and effectiveness of our algorithms through extensive experiments, in which our algorithms achieve relatively better results with less training time using a small batch size compared to existing methods. In future research, it would be intriguing to expand our work in the context of dependent and heavy-tailed stochastic gradient scenarios [Wai, 2020, Li and Liu, 2022].

Author Contributions

TX and XC contributed equally to the paper. TX was responsible for conceptualizing the idea and writing the paper. TX and XC worked together to complete the proof. XC took charge of creating the code and conducting the experiments. The paper was further revised by KB and SG.

Acknowledgements

We thank the authors of [Mancino-ball et al., 2023] for kindly providing the code framework to support our experiments. The research of KB is supported by NSF grant DMS-2053918. The research of SG is partially supported by NSERC grant RGPIN-2021-02644.

References

Sulaiman Alghunaim, Kun Yuan, and Ali H Sayed. A linearly convergent proximal gradient algorithm for decentralized optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Pierre Alquier and Gérard Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14(1), 2013.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1-2):165–214, 2023.

Lisandro Dalcin and Yao-Lung L Fang. mpi4py: Status update after 12 years of development. *Computing in Science & Engineering*, 23(4):47–54, 2021.

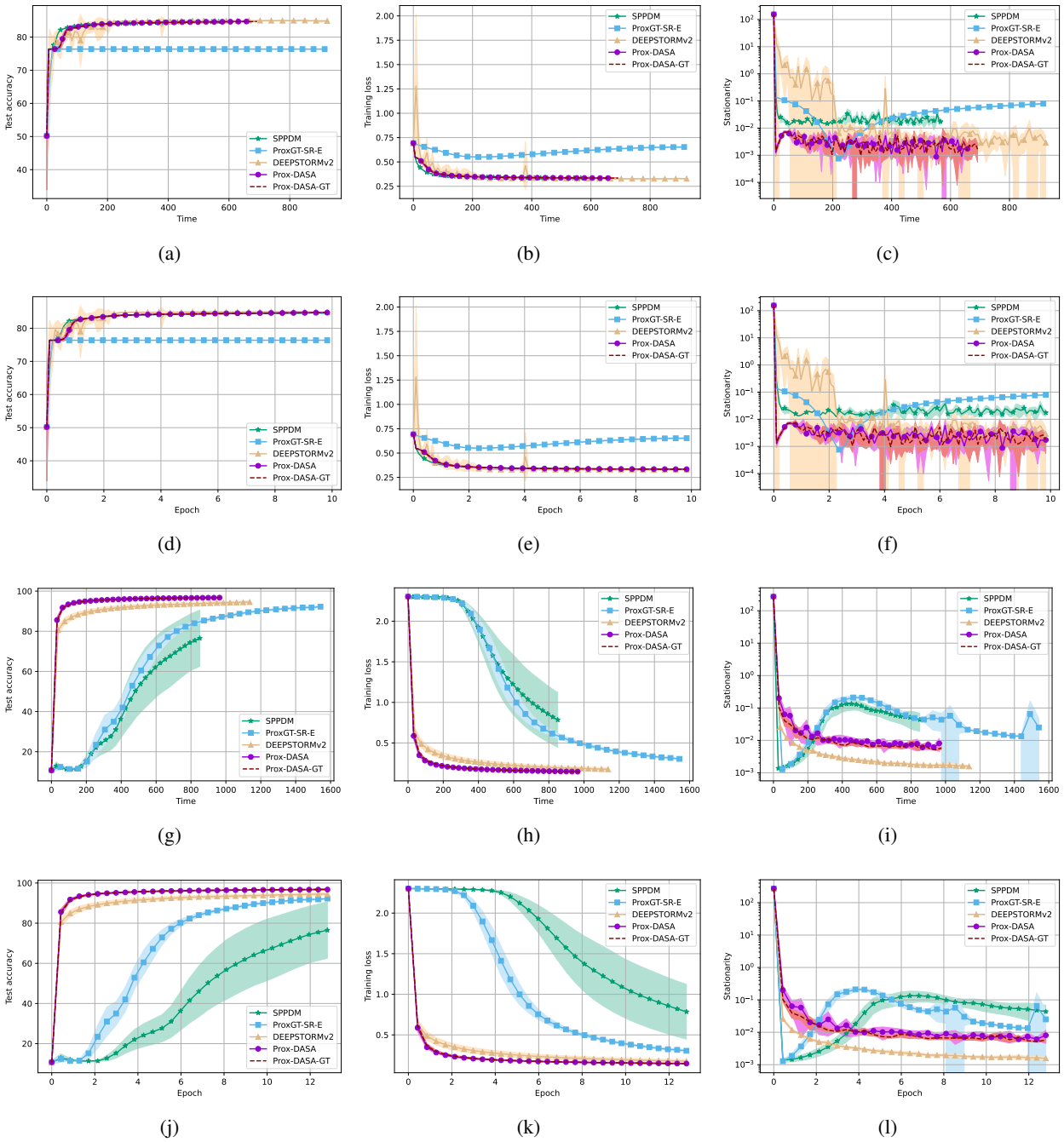

Figure 2: Comparisons between `SPPDM` [Wang et al., 2021], `ProxGT-SR-E` [Xin et al., 2021a], `DEEPSTORM` [Mancino-ball et al., 2023], `Prox-DASA` 1, and `Prox-DASA-GT` 2. The first two rows correspond to a9a and the last two rows correspond to MNIST. The results are averaged over 10 trials, and the shaded regions represent confidence intervals. The vertical axes in the third column are log-scale. It should be noted that `ProxGT-SR-E` maintains another hyperparameter q (see, e.g., Algorithm 4 and Theorem 3 in [Xin et al., 2021a]) and computes gradients using a full batch every q iterations. For simplicity, we do not include that amount of epochs when we plot this figure. In other words, the real number of epochs required to obtain a point on `ProxGT-SR` is larger than plotted in the figures in the second and fourth rows. We include the plots that take q into account in Appendix.

Paolo Di Lorenzo and Gesualdo Scutari. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

Francisco Facchinei, Gesualdo Scutari, and Simone Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, 2015.

Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

Robert M Gower, Mark Schmidt, Francis Bach, and Peter Richtárik. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao. Prox-pda: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pages 1529–1538. PMLR, 2017.

Kishore Jaganathan, Yonina C Eldar, and Babak Hassibi. Phase retrieval: An overview of recent developments. *Optical Compressive Imaging*, pages 279–312, 2016.

Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

Soomin Lee and Angelia Nedic. Distributed random projection algorithm for convex optimization. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):221–229, 2013.

Shaojie Li and Yong Liu. High probability guarantees for nonconvex stochastic gradient descent with heavy tails. In *International Conference on Machine Learning*, pages 12931–12963. PMLR, 2022.

Yao Li, Xiaorui Liu, Jiliang Tang, Ming Yan, and Kun Yuan. Decentralized composite optimization with compression. *arXiv preprint arXiv:2108.04448*, 2021.

Zhi Li, Wei Shi, and Ming Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Qing Ling and Zhi Tian. Decentralized sparse signal recovery for compressive sleeping wireless sensor networks. *IEEE Transactions on Signal Processing*, 58(7):3816–3827, 2010.

Ji Liu and A Stephen Morse. Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35 (2):160–165, 2011.

Songtao Lu, Xinwei Zhang, Haoran Sun, and Mingyi Hong. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 315–321. IEEE, 2019.

Yucheng Lu and Christopher De Sa. Optimal complexity in decentralized training. In *International Conference on Machine Learning*, pages 7111–7123. PMLR, 2021.

Gabriel Mancino-ball, Shengnan Miao, Yangyang Xu, and Jie Chen. Proximal stochastic recursive momentum methods for nonconvex composite decentralized optimization. In *AAAI Conference on Artificial Intelligence*, 2023.

Kostas Margellos, Alessandro Falsone, Simone Garatti, and Maria Prandini. Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Transactions on Automatic Control*, 63(5): 1372–1387, 2017.

Gonzalo Mateos, Juan Andrés Bazerque, and Georgios B Giannakis. Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, 58(10):5262–5276, 2010.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895. PMLR, 2018.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27 (4):2597–2633, 2017.

Alex Olshevsky. Linear time average consensus and distributed optimization on fixed graphs. *SIAM Journal on Control and Optimization*, 55(6):3990–4014, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors,

Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

Stacy Patterson, Yonina C Eldar, and Idit Keidar. Distributed compressed sensing for static and time-varying networks. *IEEE Transactions on Signal Processing*, 62(19):4931–4946, 2014.

Nhan H Pham, Lam M Nguyen, Dzung T Phan, and Quoc Tran-Dinh. Proxsarah: An efficient algorithmic framework for stochastic composite nonconvex optimization. *The Journal of Machine Learning Research*, 21(1):4455–4502, 2020.

Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187(1): 409–457, 2021.

Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2017.

Andrzej Ruszczyński. A merit function approach to the subgradient method with averaging. *Optimisation Methods and Software*, 23(1):161–172, 2008.

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

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63(22): 6013–6023, 2015.

Ying Sun, Gesualdo Scutari, and Amir Daneshmand. Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation. *SIAM Journal on Optimization*, 32(2):354–385, 2022.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. d^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pages 4848–4856. PMLR, 2018.

Hoi-To Wai. On the convergence of consensus algorithms with markovian noise and gradient bias. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4897–4902. IEEE, 2020.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhiguo Wang, Jiawei Zhang, Tsung-Hui Chang, Jian Li, and Zhi-Quan Luo. Distributed stochastic consensus optimization with momentum for nonconvex nonsmooth problems. *IEEE Transactions on Signal Processing*, 69: 4486–4501, 2021.

Xuyang Wu and Jie Lu. A unifying approximate method of multipliers for distributed composite optimization. *IEEE Transactions on Automatic Control*, 2022.

Ran Xin, Subhro Das, Usman A Khan, and Soummya Kar. A stochastic proximal gradient framework for decentralized non-convex composite optimization: Topology-independent sample complexity and communication efficiency. *arXiv preprint arXiv:2110.01594*, 2021a.

Ran Xin, Usman A Khan, and Soummya Kar. An improved convergence analysis for decentralized online stochastic non-convex optimization. *IEEE Transactions on Signal Processing*, 69:1842–1858, 2021b.

Jinming Xu, Shanying Zhu, Yeng Chai Soh, and Lihua Xie. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2055–2060. IEEE, 2015.

Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. Distributed algorithms for composite optimization: Unified framework and convergence analysis. *IEEE Transactions on Signal Processing*, 69:3555–3570, 2021.

Haishan Ye, Ziang Zhou, Luo Luo, and Tong Zhang. Decentralized accelerated proximal gradient descent. *Advances in Neural Information Processing Systems*, 33: 18308–18317, 2020.

Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

Jiaqi Zhang and Keyou You. Decentralized stochastic gradient tracking for non-convex empirical risk minimization. *arXiv preprint arXiv:1909.02712*, 2019.