D-SOBA: A Single-Loop Decentralized Bilevel Algorithm with Transient Complexity Analysis

Boao Kong*, Shuchen Zhu*, Songtao Lu†, Xinmeng Huang‡, Kun Yuan§

*Center for Data Science, Peking University, Beijing, China
†Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR of China
‡Graduate Group in Applied Mathematics and Computational Science, University of Pennsylvania, Philadelphia, USA
§Center for Machine Learning Research, Peking University, Beijing, China
{*kongboao, shuchenzhu*}*@stu.pku.edu.cn, stlu@cse.cuhk.edu.hk, xinmengh@sas.upenn.edu, kunyuan@pku.edu.cn*

***Abstract*—Stochastic bilevel optimization (SBO) is becoming increasingly essential in machine learning due to its versatility in handling nested structures. To address large-scale SBO, decentralized approaches have emerged as effective paradigms in which nodes communicate with immediate neighbors without a central server. However, current decentralized SBO algorithms face challenges, including expensive inner-loop updates and unclear understanding of the influence of network topology, data heterogeneity, and the nested bilevel algorithmic structures. In this paper, we introduce a single-loop decentralized SBO (D-SOBA) algorithm and establish its transient iteration complexity, which, for the first time, clarifies the joint influence of network topology and data heterogeneity on decentralized bilevel algorithms.**

***Index Terms*—bilevel optimization, decentralized optimization, transient iteration, non-asymptotic convergence analysis.**

I. Introduction

Decentralized stochastic bilevel optimization, which tackles problems with nested optimization structures across multiple computing nodes, has gained growing interest. This paper considers N nodes connected through a given graph topology. Each node i privately owns an upper-level loss function $f_i : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$ and a lower-level loss function $g_i : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$. All nodes collaboratively aim to find a solution to the following optimization problem:

$$\min_{x\in\mathbb{R}^d} \quad \Phi(x) := f(x, y^\star(x)) := \frac{1}{N}\sum_{i=1}^{N} f_i\left(x, y^\star(x)\right) \tag{1a}$$

$$\text{s.t.} \quad y^\star(x) := \operatorname*{arg\,min}_{y\in\mathbb{R}^p} \left\{ g(x,y) := \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \right\} \tag{1b}$$

where f_i and g_i are defined as:

$$\begin{aligned} f_i(x,y) &:= \mathbb{E}_{\xi_i\sim\mathcal{D}_{f_i}}[F(x,y;\xi_i)], \\ g_i(x,y) &:= \mathbb{E}_{\zeta_i\sim\mathcal{D}_{g_i}}[G(x,y;\zeta_i)]. \end{aligned} \tag{2}$$

The random variables ξ_i and ζ_i represent data samples available at node i, following local distributions $\mathcal{D}_{f_i}$ and $\mathcal{D}_{g_i}$, respectively. Throughout this paper, we assume local data distributions vary across different nodes, which may result in data heterogeneity issues during the training process.

Limitations in existing literature. Existing works, such as [1]–[5], have developed decentralized bilevel algorithms that offer both theoretical guarantees and empirical effectiveness. However, two key limitations remain in the current literature:

- **Expensive inner-loop updates.** Existing algorithms rely on computationally costly inner-loop updates to estimate the lower-level solution $y^\star(x)$ and the Hessian inverse of $g_i(x,y)$. These updates not only increase computational complexity but also incur significant communication overhead, limiting the practicality of the algorithms.
- **Inadequate Non-Asymptotic Analysis.** While existing studies [3]–[5] show that decentralized and centralized bilevel algorithms achieve the same asymptotic convergence rate, they fail to clarify the non-asymptotic stage, where decentralization-induced slowdowns are observed due to limited iterations in practical scenarios. Current research either overlooks the influence of network topologies or neglects the impact of data heterogeneity, leaving critical questions about when and how decentralized bilevel algorithms slow down unanswered.

Contributions. To address these limitations, we propose a **D**ecentralized **S**tochastic **O**ne-loop **B**ilevel **A**lgorithm (**D-SOBA**). Our contributions are threefold. First, we establish that D-SOBA achieves linear speedup with an asymptotic gradient complexity of $\mathcal{O}(1/(N\varepsilon^2))$, surpassing the current results by at least a factor of $\log(1/\varepsilon)$. Second, we provide a *non-asymptotic*[1] convergence analysis and derive the transient iteration complexity for the D-SOBA framework, quantifying how *network topology* and *data heterogeneity* jointly influence the non-asymptotic convergence stage. Third, we prove that our algorithm achieves the same asymptotic convergence rate and transient complexity as single-level algorithms, implying that the nested structure does not introduce fundamental challenges to decentralized bilevel optimization.

All our established results as well as those of existing decentralized SBO algorithms are listed in Table I. D-SOBA

*Equal contribution. §Corresponding author.

[1]Following recent conventions in decentralized optimization literature, we use 'non-asymptotic convergence' to characterize the iteration complexity required to achieve ε-approximate stationarity and use 'asymptotic convergence' to characterize the iteration complexity as $T \to +\infty$.

TABLE I

COMPARISON BETWEEN DIFFERENT DECENTRALIZED STOCHASTIC BILEVEL ALGORITHMS. NOTATION T INDICATES THE NUMBER OF (OUTER) ITERATIONS, $1-\rho \in (0,1]$ MEASURES THE CONNECTIVITY OF THE UNDERLYING GRAPH, N IS THE NUMBER OF COMPUTING NODES, G^2 DENOTES GRADIENT UPPER BOUND, AND b^2 DENOTES THE MAGNITUDE OF GRADIENT DISSIMILARITY. WE ALSO LIST THE RESULT OF SINGLE-LEVEL DSGD IN THE BOTTOM LINE FOR REFERENCE.

Algorithm	Single loop	Linear speedup[◊]	Asymptotic complexity[†]	Transient complexity[‡]	Assumption[◁]
DSBO [1]	✗	✗	$\frac{1}{\varepsilon^3}$	N. A.	LC f_i
MA-DSBO [2]	✗	✗	$\frac{1}{\varepsilon^2}\log(\frac{1}{\varepsilon})$	N. A.	LC f_i
SLAM [3]	✗	✓	$\frac{1}{N\varepsilon^2}\log(\frac{1}{\varepsilon})$	N. A.	LC f_i
Gossip DSBO [5]	✗	✓	$\frac{1}{N\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{N^3G^4}{(1-\rho)^4}$ [▷]	BG ∇f_i
MDBO [4, Thm 1][*]	✗	✗	$\frac{1}{(1-\rho)^2\varepsilon^2}\log(\frac{1}{\varepsilon})$	N. A.	BG ∇f_i
MDBO [4, Thm 2][*]	✗	✓	$\frac{1}{N\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{N^3}{(1-\rho)^8}$	BG $\nabla f_i, \nabla g_i, \nabla^2 g_i$
D-SOBA (ours)	✓	✓	$\boldsymbol{\frac{1}{N\varepsilon^2}}$	$\boldsymbol{\max\left\{\frac{N^3}{(1-\rho)^2}, \frac{N^3b^2}{(1-\rho)^4}\right\}}$	**BGD** $\boldsymbol{\nabla f_i, \nabla g_i, \nabla^2 g_i}$
Single-level DSGD [6]	✓	✓	$\frac{1}{N\varepsilon^2}$	$\max\left\{\frac{N^3}{(1-\rho)^2}, \frac{N^3b^2}{(1-\rho)^4}\right\}$	BGD ∇f_i

[◊] Decentralized SBO methods achieve *linear speedup* if they converge at the rate of $1/\sqrt{NT}$ eventually.
[†] #gradient/Hessian evaluations to achieve an ε-stationary solution when $\varepsilon \to 0$ (smaller is better).
[‡] #transient iterations an algorithm experiences before the asymptotic rate dominates (smaller is better).
[◁] Additional assumptions beyond convexity, smoothness, and stochastic variance. The bounded gradient dissimilarity (BGD) assumption is *weaker* than the Lipschitz continuity (LC) and bounded gradients (BG).
[▷] G is the uniform upper bound of gradients (*i.e.*, $\|\nabla f_i\|^2 \le G^2$) assumed in [5]. It typically holds that $b \ll G$ where b gauges the magnitude of the gradient dissimilarity, *i.e.*, $\frac{1}{N}\sum_{i=1}^N \|\nabla f_i - \nabla f\|^2 \le b^2$.
[*] Asymptotic rates and transient complexities are not given in [4]. But one can obtained it by the definition of asymptotic rates.

achieves the state-of-the-art asymptotic rate, asymptotic gradient complexity, and transient iteration complexity under more relaxed assumptions compared to existing methods. Furthermore, our algorithm even achieves the same theoretical convergence rates as single-level DSGD [6].

II. PRELIMINARIES

A. Notations

For a second-order differentiable function $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$, we denote $\nabla_1 f(x,y)$ and $\nabla_2 f(x,y)$ as the partial gradients at the x's position and y's position, respectively. Correspondingly, $\nabla^2_{12} f(x,y) \in \mathbb{R}^{d\times p}$ and $\nabla^2_{22} f(x,y) \in \mathbb{R}^{p\times p}$ represent its partial Jacobian matrix. Differently, we use $\nabla_x f(x, y^\star(x))$ to denote the gradient of f with respect to x by viewing y as a function of x. We let $\|\cdot\|$ denote the ℓ_2 norm of both vectors and matrices, $\|\cdot\|_F$ denote the Frobenius norm of a matrix, and $\mathbb{1}_N \in \mathbb{R}^N$ represent the vector with all elements set to 1. For any local variables $\{x_i^{(t)}\}_{i=1}^N$, the subscript i (resp, superscript t) indicates the index of the node (resp, the iteration) and we write their average $\sum_{i=1}^N x_i^{(t)}/N$ as $\bar{x}^{(t)}$. We write $a \lesssim b$ if $a \le Cb$ for a constant $C > 0$.

B. Assumptions

With the notations introduced above, we next state the assumptions used in the paper.

Assumption 1 (SMOOTHNESS): There exist positive constants $L_{\nabla f}$, $L_{\nabla g}$, $L_{\nabla^2 g}$, L_f such that for any $1 \le i \le N$,

1. ∇f_i,∇g_i,$\nabla^2 g_i$ are $L_{\nabla f}$, $L_{\nabla g}$, $L_{\nabla^2 g}$ Lipschitz continuous respectively;
2. $g_i(x,\cdot)$ is μ_g-strongly convex for any given $x \in \mathbb{R}^d$;
3. $\|\nabla_2 f_i(x, y^\star(x))\| \le L_f < \infty$ for all $x \in \mathbb{R}^d$ in which $y^\star(x)$ is defined in problem (1b).

It is noteworthy that the third condition of Assumption 1 relaxes the restrictive assumptions of Lipschitz continuity of f or, equivalently, the boundedness of $\nabla_2 f$ used in [7].

Due to the heterogeneity of local data distributions, the local functions $\{(f_i, g_i)\}_{i=1}^N$ are not identical across different nodes. To tackle this, we assume bounded gradient dissimilarity as follows, which has been widely adopted in prior literature [8].

Assumption 2 (GRADIENT DISSIMILARITY): There exists a constant $b \ge 0$ such that:

$$\frac{1}{N}\sum_{i=1}^N \|\nabla_1 f_i - \nabla_1 f\|^2 \le b^2,\ \frac{1}{N}\sum_{i=1}^N \|\nabla_2 f_i - \nabla_2 f\|^2 \le b^2,$$
$$\frac{1}{N}\sum_{i=1}^N \|\nabla_2 g_i - \nabla_2 g\|^2 \le b^2,\ \frac{1}{N}\sum_{i=1}^N \left\|\nabla^2_{12} g_i - \nabla^2_{12} g\right\|^2 \le b^2,$$
$$\frac{1}{N}\sum_{i=1}^N \left\|\nabla^2_{22} g_i - \nabla^2_{22} g\right\|^2 \le b^2.$$

We also make the following standard assumption for stochastic gradients and Hessians.

Assumption 3 (STOCHASTICITY): There exist constants $\sigma \ge 0$ such that for any given $(x,y) \in \mathbb{R}^d \times \mathbb{R}^p$ and $1 \le i \le N$, the $\nabla_1 F(x,y;\xi_i)$, $\nabla_2 F(x,y;\xi_i)$, $\nabla_2 G(x,y;\zeta_i)$, $\nabla^2_{12} G(x,y;\zeta_i)$, and $\nabla^2_{22} G(x,y;\zeta_i)$ are unbiased estimators of $\nabla_1 f_i(x,y)$, $\nabla_2 f_i(x,y)$, $\nabla_2 g_i(x,y)$, $\nabla^2_{12} g_i(x,y)$, and $\nabla^2_{22} g_i(x,y)$, respectively, with bounded variance σ^2.

To facilitate decentralized communication, we introduce the mixing matrix $W = [w_{ij}]_{i,j=1}^N \in \mathbb{R}^{N\times N}$ to gauge information between nodes in which $w_{ij} = 0$ means node i is not connected to node j. The following assumption on the mixing matrix is widely used for decentralized algorithms [9], [10].

*Assumption 4 (*MIXING MATRIX*)*: The mixing matrix W is doubly stochastic, *i.e.*,

$$\mathbb{1}_N^\top W = \mathbb{1}_N^\top, \quad W\mathbb{1}_N = \mathbb{1}_N.$$

Moreover, we assume $\rho := \left\|W - \mathbb{1}_N\mathbb{1}_N^\top/N\right\|_2 \in [0, 1)$.

*Remark 1 (*SPECTRAL GAP*)*: In decentralized algorithms, the quantity $1-\rho$ is known as the *spectral gap* [11], [12] of W, which serves as a metric for measuring the connectivity of the network topology. Notably, as $1-\rho \to 1$, it indicates that the topology is well-connected. Conversely, as $1-\rho \to 0$, it suggests that the topology is potentially sparse [13].

III. D-SOBA ALGORITHM

The core challenge in stochastic bilevel optimization lies in estimating the hypergradient $\nabla\Phi(x)$, due to the implicit dependence of $y^\star(x)$ on x. Under Assumption 1, $\nabla\Phi(x)$ is:

$$\nabla\Phi(x) = \nabla_1 f(x, y^\star(x)) - \Big(\nabla^2_{12} g(x, y^\star(x)) \cdot \left[\nabla^2_{22} g(x, y^\star(x))\right]^{-1} \cdot \nabla_2 f(x, y^\star(x))\Big) \tag{4}$$

which is computationally expensive due to the need for inverting the partial Hessian. Moreover, the Hessian inversion

$$\left[\nabla^2_{22} g(x, y^\star(x))\right]^{-1} = \left[\frac{1}{N}\sum_{i=1}^N \nabla^2_{22} g_i(x, y^\star(x))\right]^{-1}$$

cannot be easily accessed by decentralized communication. However, these challenges can be effectively addressed by the novel single-node framework known as SOBA [14]. SOBA introduces $z^\star(x) = \left[\nabla^2_{22} g(x, y^\star(x))\right]^{-1}\nabla_2 f(x, y^\star(x))$, which can be interpreted as the solution to minimizing the problem:

$$\frac{1}{N}\sum_{i=1}^N \left\{\frac{1}{2} z^\top \nabla^2_{22} g_i(x, y^\star)\, z - z^\top \nabla_2 f_i(x, y^\star)\right\}. \tag{5}$$

where $y^\star$ denotes $y^\star(x)$. It is worth noting that while $z^\star(x)$ cannot be written as a finite sum across nodes, problem (5) involves only simple sums.

To save computation, we can approximately solve (5) using one-step (stochastic) gradient descent. Thus, combined with one-step (stochastic) gradient descent to update the upper- and lower-level variables (x, y), forms the centralized single-loop framework for solving problem (1):

$$\text{minimize (1a):} \quad x^{(t+1)} = x^{(t)} - \frac{\alpha_t}{N}\sum_{i=1}^N D^{(t)}_{x,i}, \tag{6a}$$

$$\text{minimize (1b):} \quad y^{(t+1)} = y^{(t)} - \frac{\beta_t}{N}\sum_{i=1}^N D^{(t)}_{y,i}, \tag{6b}$$

$$\text{minimize (5):} \quad z^{(t+1)} = z^{(t)} - \frac{\gamma_t}{N}\sum_{i=1}^N D^{(t)}_{z,i}, \tag{6c}$$

where $D^{(t)}_{x,i}, D^{(t)}_{y,i}$ and $D^{(t)}_{z,i}$ are unbiased estimates of

$$D^{(t)}_{x,i}(x,y,z) = \nabla_1 f_i(x^{(t)}, y^{(t)}) - \nabla^2_{12}\, g_i(x^{(t)}, y^{(t)}) z^{(t)},$$
$$D^{(t)}_{y,i}(x,y,z) = \nabla_2\, g_i(x^{(t)}, y^{(t)}),$$
$$D^{(t)}_{z,i}(x,y,z) = \nabla^2_{22}\, g_i(x^{(t)}, y^{(t)}) z^{(t)} - \nabla_2 f_i(x^{(t)}, y^{(t)}),$$

respectively, and α_t, β_t and γ_t are learning rates. We denote recursion (6) as centralized SOBA due to the requirement of a central server. It is noteworthy that (6) does not require any inner loop to approximate the lower-level solution $y^\star(x)$ or directly evaluate the Hessian inversion of $g(x, y)$.

Inspired by decentralized gradient descent [15], we extend centralized SOBA (6) to decentralized setup:

$$x_i^{(t+1)} = \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^{(t)} - \alpha_t D^{(t)}_{x,j}), \tag{7a}$$

$$y_i^{(t+1)} = \sum_{j\in\mathcal{N}_i} w_{ij}(y_j^{(t)} - \beta_t D^{(t)}_{y,j}), \tag{7b}$$

$$z_i^{(t+1)} = \sum_{j\in\mathcal{N}_i} w_{ij}(z_j^{(t)} - \gamma_t D^{(t)}_{z,j}), \tag{7c}$$

where $x_i^{(t)}, y_i^{(t)}$ and $z_i^{(t)}$ are local variables maintained by each node i at iteration t. The set $\mathcal{N}_i$ includes node i and all its immediate neighbors.

The detailed implementation of recursion (7) is listed in Algorithm 1. For each iteration t, we independently sample a minibatch of data $\xi_i^{(t)} \sim \mathcal{D}_{f_i}$, $\zeta_i^{(t)} \sim \mathcal{D}_{g_i}$ and compute

$$u^{(t)}_{i,x} := \nabla_1 F(x_i^{(t)}, y_i^{(t)}; \xi_i^{(t)}),\ u^{(t)}_{i,y} := \nabla_2 F(x_i^{(t)}, y_i^{(t)}; \xi_i^{(t)});$$
$$v_i^{(t)} := \nabla_2 G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)});$$
$$J_i^{(t)} := \nabla^2_{12} G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)}), H_i^{(t)} := \nabla^2_{22} G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)}).$$

These variables are used in Algorithm 1 at each iteration. Furthermore, we impose a moving average on the update of x_i in Algorithm 1 as it is essential to reduce the order of bias from sample noise in the convergence analysis and relax the technical assumptions.

Algorithm 1 D-SOBA

```
Require: initialize $x^{(0)} = y^{(0)} = z^{(0)} = h^{(0)} = 0$, $\{\alpha_t\}, \{\beta_t\}, \{\gamma_t\}, \{\theta_t\}$, and the mixing matrix $W$
  for $t = 0, 1, \cdots, T-1$ do
    for each node $i = 1, 2, \cdots, N$ in parallel do
      $x_i^{(t+1)} := \sum_{j\in\mathcal{N}_i} w_{ij}(x_j^{(t)} - \alpha_t h_j^{(t)})$;
      $y_i^{(t+1)} := \sum_{j\in\mathcal{N}_i} w_{ij}(y_j^{(t)} - \beta_t v_j^{(t)})$;
      $z_i^{(t+1)} := \sum_{j\in\mathcal{N}_i} w_{ij}\left(z_j^{(t)} - \gamma_t(H_j^{(t)} z_j^{(t)} - u^{(t)}_{i,y})\right)$;
      $\omega_i^{(t+1)} := u^{(t)}_{i,x} - J_i^{(t)} z_i^{(t)}$;
      $h_i^{(t+1)} := (1-\theta_t) h_i^{(t)} + \theta_t \omega_i^{(t+1)}$.
    end for
  end for
```

IV. CONVERGENCE ANALYSIS

In this section, we establish the convergence guarantees for D-SOBA. At the beginning, we introduce the notation $\mathcal{F}^{(t)} = \sigma\left[\bigcup_{\tau=0}^t \left(\bigcup_{i=1}^N \{x_i^{(\tau)}, y_i^{(\tau)}, z_i^{(\tau)}, h_i^{(\tau)}\} \bigcup \{\alpha_t, \beta_t, \gamma_t, \theta_t, \delta_t\}\right)\right]$ to denote the σ-field generated by all items with superscripts in the first t iterations. We also denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}^{(t)}]$ as the conditional expectation with respect to $\mathcal{F}^{(t)}$. Now we present the convergence rate for D-SOBA, which can be described as the following theorem:

Theorem 1: Under Assumptions 1, 2, 3, and 4, there exist $c_1, c_2, c_3 > 0$ such that if $\alpha_t \equiv \Theta(\sqrt{N/T})$, and $\beta_t \equiv c_1\alpha_t$, $\gamma_t \equiv c_2\alpha_t$, $\theta_t \equiv c_3\alpha_t$ for any $0 \le t < T$, the iterators $\bar{y}^{(t)}, \bar{z}^{(t)}$ in D-SOBA satisfy:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\bar{y}^{(t)} - \bar{y}_*^{(t)}\right\|^2 + \left\|\bar{z}^{(t)} - \bar{z}_*^{(t)}\right\|^2\right] \tag{9}$$

$$\lesssim \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\bar{h}^{(t)}\right\|^2\right] + \frac{\alpha}{N} + \frac{\rho^2\alpha^2 b^2}{(1-\rho)^2} + \frac{\rho^2\alpha^2\iota^2}{(1-\rho)^2} + \frac{\rho^2\alpha^2}{(1-\rho)},$$

where $\bar{x}^{(t)} = (1/N)\sum_{i=1}^N x_i^{(t)}$, $\bar{y}^{\star(t)}$ denotes the minimum of (1b) with respect to $\bar{x}^{(t)}$, and $\bar{z}^{\star(t)}$ denotes the minimum of (5) with respect to $\bar{x}^{(t)}, \bar{y}^{\star(t)}$. Furthermore, it holds that

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla\Phi(\bar{x}^{(t)})\right\|^2\right] \lesssim \frac{\kappa^5}{\sqrt{NT}} + \frac{\rho^{\frac{2}{3}}\kappa^6}{(1-\rho)^{\frac{1}{3}}T^{\frac{2}{3}}}$$

$$+ \frac{\rho^{\frac{2}{3}}b^{\frac{2}{3}}\kappa^6}{(1-\rho)^{\frac{2}{3}}T^{\frac{2}{3}}} + \frac{\rho\kappa^6}{(1-\rho)T} + \frac{\kappa^{13}}{T} + \frac{\kappa^7}{NT}, \tag{10}$$

where $\kappa := \max\{L_f, L_{\nabla f}, L_{\nabla g}, L_{\nabla^2 g}\}/\mu_g$ denotes the condition number (Proof is in our long report [16]).

Asymptotic linear speedup. An algorithm reaches the *linear speedup* stage if the term $1/\sqrt{NT}$ dominates the convergence rate [8] eventually (*i.e.*, $T \to \infty$). In this regime, the required iterations to achieve an ε-stationary solution can decrease linearly as the number of computing nodes increases. D-SOBA attains linear speedup eventually while algorithms of [1], [2] only achieve a much slower asymptotic rate $1/\sqrt{T}$.

Transient iteration complexity. Transient iteration complexity [17] refers to the number of iterations an algorithm has to experience before reaching its asymptotic linear-speedup stage, that is, iterations $1, \cdots, T$ where T is relatively small so that non-NT terms still dominate the rate. Transient iteration complexity measures the non-asymptotic stage in decentralized stochastic algorithms. Here, we mainly consider the influence of N, ρ, b on transient complexity analysis. Many existing works, such as [1], [2], fail to establish the transient iteration complexity as their analysis ignores all non-dominant convergence terms. With fine-grained convergence rate (10), the transient iteration complexity of D-SOBA is derived as:

*Corollary 1 (*TRANSIENT ITERATION COMPLEXITY*):* Under the same assumptions as in Theorem 1, the transient iteration complexity of D-SOBA is $\mathcal{O}\left(\max\left\{\frac{N^3}{(1-\rho)^2}, \frac{N^3 b^2}{(1-\rho)^4}\right\}\right)$.

Joint influence of graph and heterogeneity. To our knowledge, Corollary 1 is the first result that quantifies how network topology and data heterogeneity jointly affect the non-asymptotic convergence in decentralized SBO. First, Corollary 1 implies that a sparse topology with $1-\rho \to 0$ can significantly amplify the influence of data heterogeneity b^2. Second, a large data heterogeneity b^2 also exacerbates the adverse impact of sparse topologies from $\mathcal{O}((1-\rho)^{-2})$ to $\mathcal{O}((1-\rho)^{-4})$. Furthermore, Corollary 1 also implies strategies to improve the transient iteration complexity: developing well-connected graphs with $\rho \to 0$, or developing more effective decentralized algorithms that can remove the influence of b^2 (*e.g.*, algorithms built upon gradient tracking [18] or Exact-Diffusion [12]). In contrast, existing transient complexities [4], [5] are worse than our Corollary 1, especially when $b^2 \to 0$ or $(1-\rho) \to 0$, see Table I for detailed comparison.

As effective as single-level DSGD. We find that both the asymptotic rate and transient iteration complexity in D-SOBA are identical to that of single-level decentralized SGD [6]. This implies that the nested lower- and upper-level structure does not pose substantial challenges to decentralized stochastic optimization in terms of both asymptotic rate and transient iteration complexity.

V. Experiments

We use the hyper-cleaning problem [19] with the Fashion MNIST dataset [20] to validate our theoretical findings. The 60,000 training images are split into a training set of 50,000 images and a validation set of 10,000 images. The hyper-cleaning problem involves training a classifier in a corrupted setting, where the label of each training sample is replaced by a random class with probability p (i.e., the corruption rate). This is performed on a decentralized network with $N = 10$ clients, which can be formulated as (1), where the loss functions at the upper and lower levels for the i-th node are defined as:

$$f_i(x,y) = \frac{1}{\left|\mathcal{D}_{val}^{(i)}\right|}\sum_{(\xi_e,\zeta_e)\in D_{val}^{(i)}} L(\phi(\xi_e; y), \zeta_e),$$

$$g_i(x,y) = \frac{1}{\left|\mathcal{D}_{tr}^{(i)}\right|}\sum_{(\xi_e,\zeta_e)\in\mathcal{D}_{tr}^{(i)}} \sigma(x_e)L(\phi(\xi_e; y), \zeta_e) + C\|y\|^2,$$

where ϕ denotes the parameters of a two-layer MLP network with a 300-dimensional hidden layer and ReLU activation, while y represents its parameters. L denotes the cross-entropy loss. The sigmoid function $\sigma(x_e)$ assigns reduced weights to corrupted samples through nonlinear activation. $\mathcal{D}_{val}^{(i)}$ and $\mathcal{D}_{tr}^{(i)}$ represent the validation and training sets of client i, which are sampled randomly from a Dirichlet distribution with parameter $\alpha = 0.1$ in the non-i.i.d. case. We set $C = 0.001$ and the step sizes to 0.1. All experiments are repeated 10 times.

We first compare D-SOBA with MA-DSBO [2] and Gossip DSBO [5] over an exponential graph [13] with $p = 0.1, 0.4$, respectively. For all algorithms, the step size is set to 0.1 and the batch size is set to 200. The moving average parameter θ_t for D-SOBA and MA-DSBO is set to 0.8. For MA-DSBO, we set the number of inner-loop iterations to $T = 5$ and the number of outer-loop iterations to $K = 5$. For Gossip DSBO, the number of Hessian-inverse estimation iterations is set to $T = 5$. At the end of the outer parameter update, we use the average of y across all clients for classification on the test set. Figure 1 illustrates the upper-level loss for these algorithms, with D-SOBA converging faster than the other algorithms.

Next, we apply D-SOBA to various topologies, including Adjusted Ring, 2D-Torus, Exponential graph [13], and the centralized case, under a non-i.i.d. setting. We set $p = 0.2$, a batch size of 200, and repeat each case 10 times, reporting

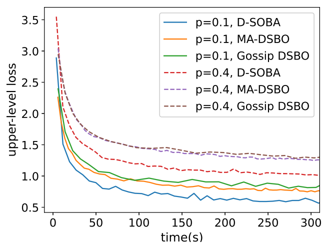

Fig. 1. The upper-level loss of different decentralized stochastic bilevel optimization algorithms.

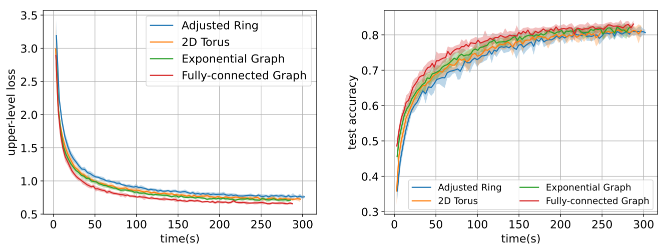

Fig. 2. The validation loss (left) and test accuracy (right) of D-SOBA with different communication topologies.

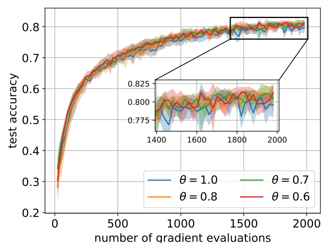

Fig. 3. The test accuracy of D-SOBA with different moving-average parameter θ_t.

the mean of all trials. Figure 2 presents the upper-level loss and test accuracy for the different cases. From these results, we observe that topologies with smaller spectral gaps converge more rapidly and achieve higher test accuracy in less time.

Finally, we investigate the influence of the moving average parameter θ on convergence. With a batch size of 100 and $p = 0.2$, we run D-SOBA on an Adjusted Ring with $\theta \in \{1.0, 0.8, 0.7, 0.6\}$, where the weight matrix $W = [w_{ij}]_{N\times N}$ of the Adjusted Ring satisfies:

$$w_{ij} = \begin{cases} 0.4, & \text{if } (j-i)\%N \in \{\pm 1\}, \\ 0.2, & \text{if } j = i, \\ 0, & \text{else.} \end{cases}$$

We repeat each case 10 times to obtain the test accuracy. From Figure 3, the cases with $\theta = 0.7, 0.8$ achieve higher average test accuracy than those with $\theta = 0.6, 1.0$, suggesting a trade-off in the choice of θ with respect to convergence.

VI. Conclusion

This paper introduces D-SOBA, a single-loop algorithm for decentralized stochastic bilevel optimization that achieves a state-of-the-art asymptotic convergence rate, as well as improved gradient and Hessian complexity, under more relaxed assumptions compared to existing methods. Additionally, we provide the first analysis of the joint influence of network topology and data heterogeneity on bilevel algorithms, focusing on transient iteration complexity.

References

[1] X. Chen, M. Huang, and S. Ma, "Decentralized bilevel optimization," *arXiv preprint arXiv:2206.05670*, 2022.

[2] X. Chen, M. Huang, S. Ma, and K. Balasubramanian, "Decentralized stochastic bilevel optimization with improved per-iteration complexity," in *International Conference on Machine Learning*. PMLR, 2023, pp. 4641–4671.

[3] S. Lu, S. Zeng, X. Cui, M. Squillante, L. Horesh, B. Kingsbury, J. Liu, and M. Hong, "A stochastic linearized augmented lagrangian method for decentralized bilevel optimization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 638–30 650, 2022.

[4] H. Gao, B. Gu, and M. T. Thai, "On the convergence of distributed stochastic bilevel optimization algorithms over a network," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 9238–9281.

[5] S. Yang, X. Zhang, and M. Wang, "Decentralized gossip-based stochastic bilevel optimization over communication networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 238–252, 2022.

[6] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Accelerating gossip sgd with periodic global averaging," in *International Conference on Machine Learning (ICML)*, 2021.

[7] X. Chen, T. Xiao, and K. Balasubramanian, "Optimal algorithms for stochastic bilevel optimization under relaxed smoothness conditions," *arXiv preprint arXiv:2306.12067*, 2023.

[8] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[9] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International Conference on Machine Learning (ICML)*, 2020, pp. 1–12.

[10] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[11] Y. Lu and C. De Sa, "Optimal complexity in decentralized training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7111–7123.

[12] K. Yuan, S. A. Alghunaim, and X. Huang, "Removing data heterogeneity influence enhances network topology dependence of decentralized SGD," *Journal of Machine Learning Research*, vol. 24, no. 280, pp. 1–53, 2023.

[13] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin, "Exponential graph is provably efficient for decentralized deep training," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[14] M. Dagréou, P. Ablin, S. Vaiter, and T. Moreau, "A framework for bilevel optimization that enables stochastic and global variance reduction algorithms," *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 698–26 710, 2022.

[15] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[16] B. Kong, S. Zhu, S. Lu, X. Huang, and K. Yuan, "Decentralized bilevel optimization over graphs: Loopless algorithmic update and transient iteration complexity," *arXiv preprint arXiv:2402.03167*, 2024.

[17] S. Pu, A. Olshevsky, and I. C. Paschalidis, "A sharp estimate on the transient time of distributed stochastic gradient descent," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 5900–5915, 2021.

[18] S. A. Alghunaim and K. Yuan, "A unified and refined convergence analysis for non-convex decentralized learning," *IEEE Transactions on Signal Processing*, 2022.

[19] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1723–1732.

[20] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.