北京大学
PEKING UNIVERSITY

SPARKLE: A Unified Single-Loop Primal-Dual Framework for Decentralized Bilevel Optimization

Kun Yuan (袁 坤)

Center for Machine Learning Research @ Peking University

Joint work with

北京大学 PEKING UNIVERSITY

Shuchen Zhu
(PKU)

Boao Kong
(PKU)

Songtao Lu
(IBM Research)

Xinmeng Huang
(Upenn)

Decentralized Bilevel Optimization over Graphs: Loopless Algorithmic Update and Transient Iteration Complexity

Boao Kong[1*] Shuchen Zhu[2*] Songtao Lu[3] Xinmeng Huang[4] Kun Yuan[1†]

[1]Peking University [2]Nankai University [3]IBM Research [4]University of Pennsylvania

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, thereby improving communication efficiency and enhancing algorithmic robustness. 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. D-SOBA achieves the state-of-the-art asymptotic rate, asymptotic gradient/Hessian complexity, and transient iteration complexity under more relaxed assumptions compared to existing methods. Numerical experiments validate our theoretical findings.

1. Introduction

Stochastic bilevel optimization, which tackles problems with nested optimization structures, has gained growing interest. This two-level structure provides a flexible and potent framework for addressing a broad range of tasks, ranging from meta-learning and hyperparameter optimization [21, 54, 4] to reinforcement learning [28], adversarial learning [73], continue learning [5], and imitation learning [2]. State-of-the-art performance in these tasks is typically achieved with extremely large training datasets, which necessitates efficient distributed algorithms for stochastic bilevel optimization across multiple computing nodes.

*Equal contribution.
†Corresponding author <kunyuan@pku.edu.cn>

This paper considers N collaborative nodes connected through a given graph topology. Each node i privately owns a upper-level cost function $f_i : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$ and a lower-level cost function $g_i : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$. The goal of all nodes is to find a solution to the following distributed stochastic bilevel optimization problem:

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

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

where local cost functions f_i and g_i are defined as the expectation of the random function $F(x, y; \xi_i)$ and $G(x, y; \zeta_i)$:

$$f_i(x,y) := \mathbb{E}_{\xi_i\sim\mathcal{D}_{f_i}}[F(x,y;\xi_i)] \tag{2a}$$

$$g_i(x,y) := \mathbb{E}_{\zeta_i\sim\mathcal{D}_{g_i}}[G(x,y;\zeta_i)] \tag{2b}$$

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.

1.1. Decentralized Bilevel Algorithms

Conventional distributed approaches for solving problem (1) typically adopt a centralized paradigm [29, 63, 56]. In these approaches, a central server is employed to synchronize across the entire network, facilitating the evaluation of a globally averaged upper- or lower-level stochastic gradient for updating the model parameters. This global averaging step can be realized through mechanisms such as Parameter Server [53, 33] or Ring-Allreduce [24]. However, these techniques are associated with either substantial bandwidth costs or high latency [65, Table I], which significantly hampers the scalability of centralized bilevel optimization.

arXiv 2024

SPARKLE: A Unified Single-Loop Primal-Dual Framework for Decentralized Bilevel Optimization

Shuchen Zhu*
Peking University
shuchenzhu@stu.pku.edu.cn

Boao Kong*
Peking University
kongboao@stu.pku.edu.cn

Songtao Lu
IBM Research
songtao@ibm.com

Xinmeng Huang
University of Pennsylvania
xinmengh@sas.upenn.edu

Kun Yuan†
Peking University
kunyuan@pku.edu.cn

Abstract

This paper studies decentralized bilevel optimization, in which multiple agents collaborate to solve problems involving nested optimization structures with neighborhood communications. Most existing literature primarily utilizes gradient tracking to mitigate the influence of data heterogeneity, without exploring other well-known heterogeneity-correction techniques such as EXTRA or Exact Diffusion. Additionally, these studies often employ identical decentralized strategies for both upper- and lower-level problems, neglecting to leverage distinct mechanisms across different levels. To address these limitations, this paper proposes **SPARKLE**, a unified **S**ingle-loop **P**rimal-dual **A**lgo**R**ithm framewor**K** for decentra**L**ized bil**E**vel optimization. SPARKLE offers the flexibility to incorporate various heterogeneity-correction strategies into the algorithm. Moreover, SPARKLE allows for different strategies to solve upper- and lower-level problems. We present a unified convergence analysis for SPARKLE, applicable to all its variants, with state-of-the-art convergence rates compared to existing decentralized bilevel algorithms. Our results further reveal that EXTRA and Exact Diffusion are more suitable for decentralized bilevel optimization, and using mixed strategies in bilevel algorithms brings more benefits than relying solely on gradient tracking.

1 Introduction

Numerous modern machine learning tasks, such as reinforcement learning [25], meta-learning [4], adversarial learning [36], hyper-parameter optimization [19], and imitation learning [3], entail nested optimization formulations that extend beyond the traditional single-level paradigm. For instance, hyper-parameter optimization aims to identify the optimal hyper-parameters for a specific learning task in the upper level by minimizing the validation loss, achieved through training models in the lower-level process. This nested optimization structure has spurred significant attention towards Stochastic Bilevel Optimization (SBO). Since the size of data samples involved in bilevel problems has become increasingly large, this paper investigates decentralized algorithms over a network of n agents (nodes) that collaborate to solve the following distributed bilevel optimization problem:

$$\min_{x\in\mathbb{R}^p} \quad \Phi(x) = f(x, y^\star(x)) := \frac{1}{n}\sum_{i=1}^n f_i(x, y^\star(x)), \qquad \text{(upper-level)} \tag{1a}$$

*Equal Contribution
†Corresponding Author: Kun Yuan. Kun Yuan is also affiliated with National Engineering Labratory for Big Data Analytics and Applications, and AI for Science Institute, Beijing, China.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

NeurIPS 2024

北京大学
PEKING UNIVERSITY

PART 01

Bilevel Optimization

Bilevel Optimization

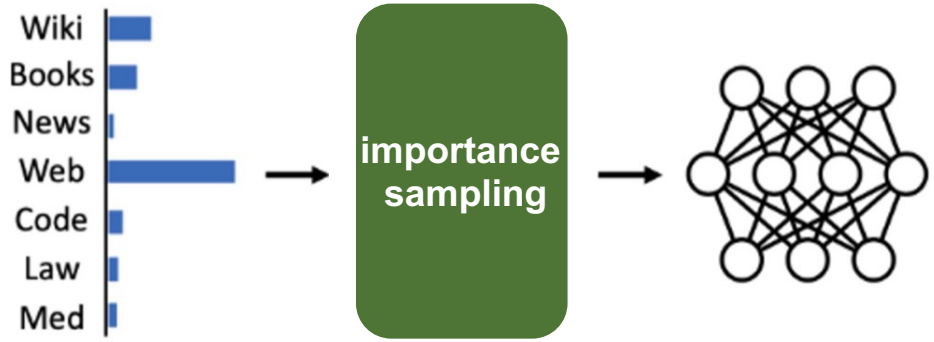
$$\min_{x \in \mathbb{R}^d} \quad \Phi(x) = f(x, y^*(x)) \qquad \textbf{(Upper Level)}$$

$$\text{s.t.} \quad y^*(x) = \arg\min_{y \in \mathbb{R}^p} g(x, y) \qquad \textbf{(Lower Level)}$$

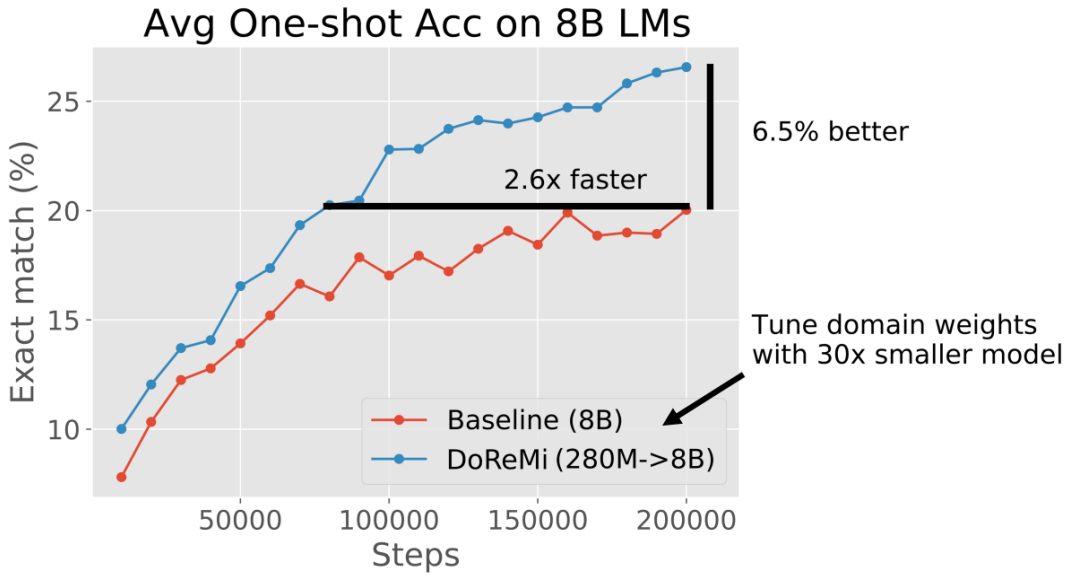
- We assume both $f(\cdot)$ and $g(\cdot)$ are smooth functions
- We further assume $f(\cdot)$ is non-convex w.r.t. x and $g(\cdot)$ is strongly-convex w.r.t. y
- If $g = -f$, the above bilevel optimization problem reduces to a minimax problem:

$$\min_x \max_y f(x, y)$$

Application: Domain reweighting

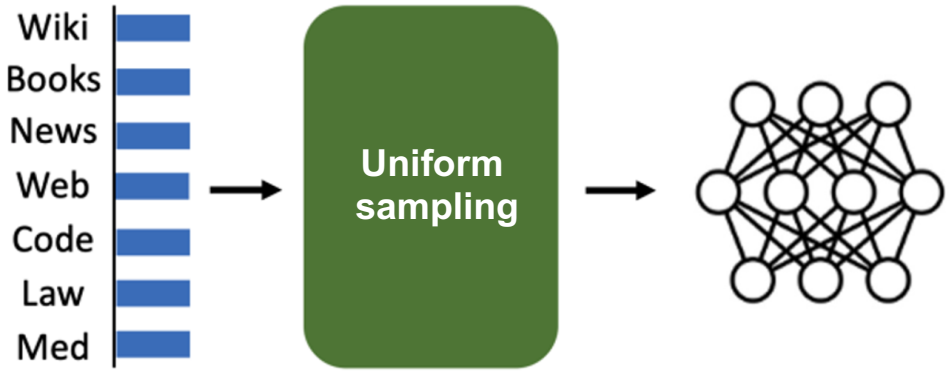


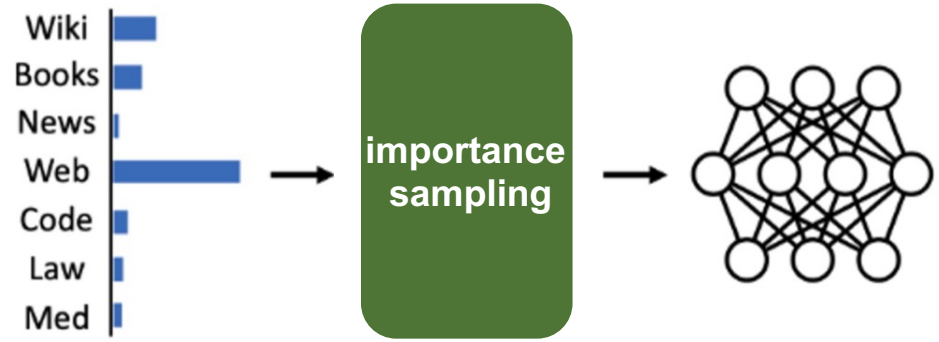
- Data for LLM comes from various sources (Wiki, books, news, code, etc.)
- The mixture proportions of pretraining data domains greatly affect LLM performance



[Xie, et.al., *DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining,* NeurIPS 2023]

Application: Domain reweighting





- How to decide data mixture proportions?

Bilevel Optimization!

$$\min_{p \in \Lambda} \quad \underline{L_{\text{val}}}(w^\star(p))$$

Loss on validation dataset

$$\text{s.t.} \quad w^\star(p) = \arg\min_{w} \sum_{i=1}^{m} \frac{p_i}{n_i} \sum_{j=1}^{n_i} \underline{L_{\text{trn}}}(w; \xi_j^i)$$

Loss on training dataset

[Pan, et.al., *ScaleBiO: Scalable Bilevel Optimization for LLM Data Reweighting,* 2024]

Other applications

北京大学 PEKING UNIVERSITY

- Meta-learning and hyperparameter optimization

[L. Bertinetto, et. al., *Meta-learning with differentiable closed-form solvers,* ICLR, 2019.]

[L. Franceschi, et. al., *Bilevel programming for hyperparameter optimization and meta-learning,* ICML, 2018.]

[J. Snell, et. al., *Prototypical networks for few-shot learning*, NeurIPS, 2017.]

- Reinforcement learning, adversarial learning, continue learning, imitation learning

[M. Hong, et. al., *A two-timescale stochastic algorithm framework for bilevel optimization*, SIAM J. Optim., 2023.]

[Y. Zhang, et. al., *Revisiting and advancing fast adversarial training through the lens of bilevel optimization*, ICML, 2022.]

[Z. Borsos, et. al., *Coresets via bilevel optimization for continual learning and streaming*, NeurIPS, 2020.]

[S. Arora, et. al., *Provable representation learning for imitation learning via bilevel optimization*, ICML, 2020.]

北京大学
PEKING UNIVERSITY

PART 02

Decentralized Stochastic Bilevel Optimization

Distributed Stochastic Bilevel Optimization

北京大学 PEKING UNIVERSITY

- Bilevel optimization may involve **massive** datasets, especially in LLM tasks

data reweighting

$$\min_{p\in\Lambda} \quad L_{\text{val}}(w^\star(p))$$

$$\text{s.t.} \quad w^\star(p) = \arg\min_{w} \sum_{i=1}^{m} \frac{p_i}{n_i} \sum_{j=1}^{n_i} L_{\text{trn}}(w; \xi_j^i)$$

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

[LLaMA: Open and Efficient Foundation Language Models]

- Efficient distributed stochastic bilevel optimization (SBO) algorithms are in urgent need

Distributed Stochastic Bilevel Optimization

- We consider the following distributed stochastic bilevel optimization problem over N nodes

$$\min_{x\in\mathbb{R}^d} \quad \Phi(x) := f(x, y^*(x)) := \frac{1}{N}\sum_{i=1}^{N} f_i(x, y^*(x)) \qquad \textbf{(Upper Level)}$$

$$\text{s.t.} \quad y^*(x) := \arg\min_{y\in\mathbb{R}^p} \left\{ g(x,y) := \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \right\} \qquad \textbf{(Lower Level)}$$

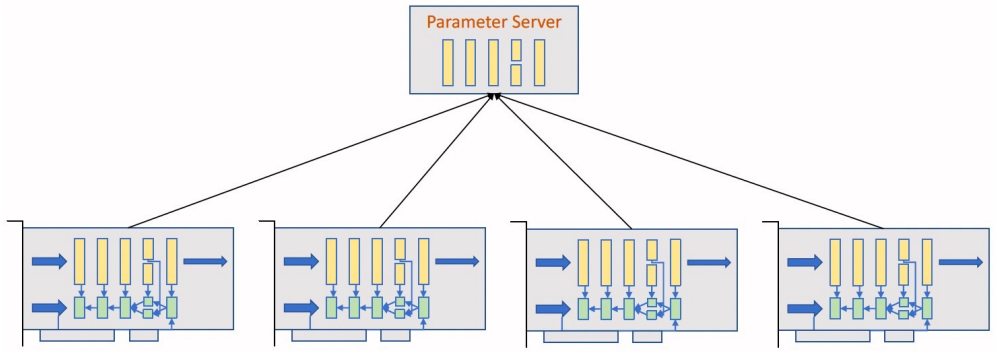
  - Each $f_i(x,y) = \mathbb{E}_{\phi\sim\mathcal{D}_{f_i}}[F_i(x,y;\phi)]$ is a local upper level stochastic cost function
  - Each $g_i(x,y) = \mathbb{E}_{\xi\sim\mathcal{D}_{g_i}}[G_i(x,y;\xi)]$ is a local lower level stochastic cost function
  - Random variables ϕ and ξ denotes the local data

Distributed Stochastic Bilevel Optimization

- Traditional distributed bilevel algorithms rely on **centralized** averaging

$$x_i^{(t+\frac{1}{2})} = x_i^{(t)} - \gamma h_i^{(t)} \qquad \text{(Local compt.)}$$

$$x_i^{(t+1)} = \frac{1}{N}\sum_{j=1}^{N} x_j^{(t+\frac{1}{2})} \qquad \text{(Global comm.)}$$

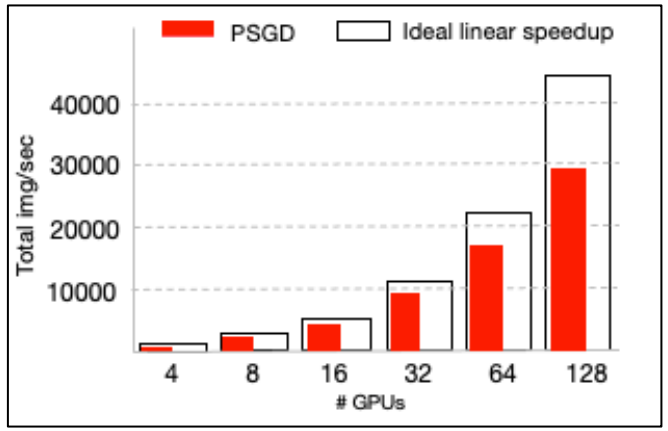


- Centralized averaging involves N averages; incurs significant comm. time

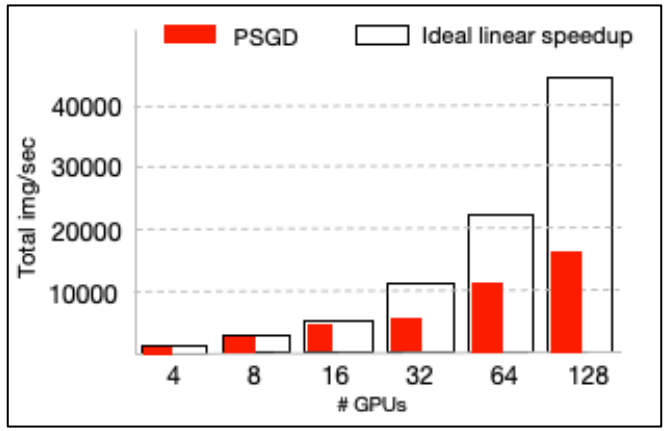
Distributed Stochastic Bilevel Optimization

北京大学 PEKING UNIVERSITY

- Centralized averaging **cannot** achieve ideal linear speedup in throughput due to comm. overhead
- Larger comm-to-compt ratio leads to worse performance in PSGD



Small comm.-to-compt. ratio



Large comm.-to-compt. ratio

Ring-Allreduce is used in each experiment

- How can we reduce the communication overhead in distributed bilevel optimization?

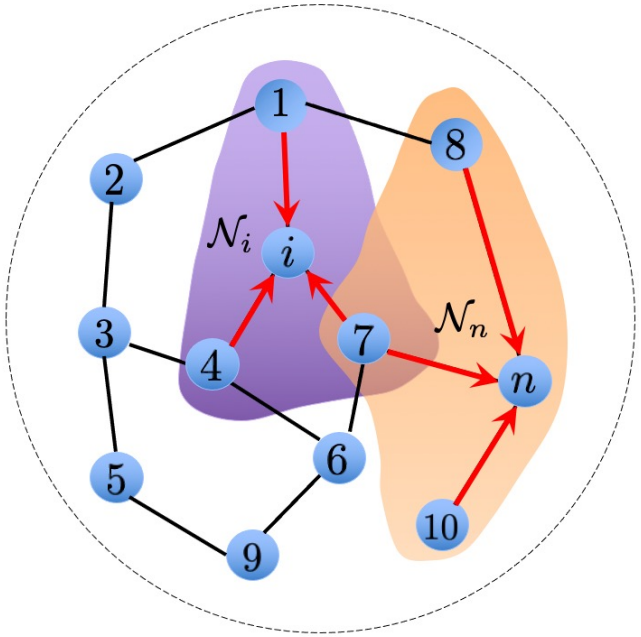
B. Ying, K. Yuan, H. Hu, Y. Chen and W. Yin, “BlueFog: Make decentralized algorithms practical for optimization and deep learning”, arXiv: 2111. 04287, 2021

Decentralized Stochastic Bilevel Optimization

- Decentralized bilevel algorithms rely on **partial** averaging

$$x_i^{(t+\frac{1}{2})} = x_i^{(t)} - \gamma h_i^{(t)} \qquad \text{(Local compt.)}$$

$$x_i^{(t+1)} = \sum_{j\in\mathcal{N}_i} w_{ij} x_j^{(t+\frac{1}{2})} \qquad \text{(Partial comm.)}$$



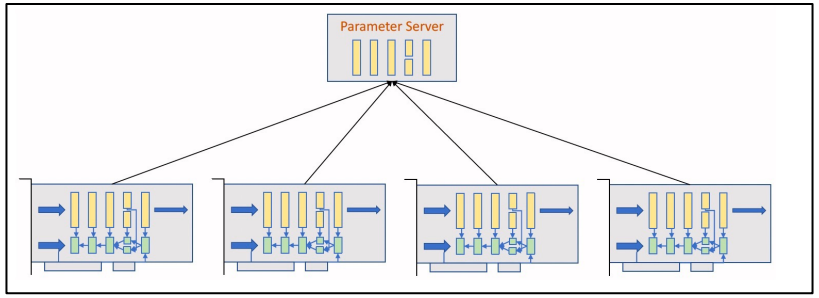
  - $\mathcal{N}_i$ is the set of neighbors at node i
  - w_{ij} scales information from node j to node i;satisfying $\sum_{j\in\mathcal{N}_i} w_{ij} = 1$

- Significant communication saving with partial averaging; from $O(n)$ to $O(d_{\max})$

Decentralized Stochastic Bilevel Optimization

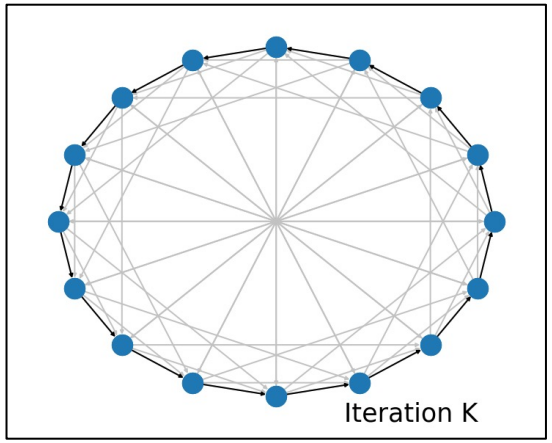
北京大学 PEKING UNIVERSITY

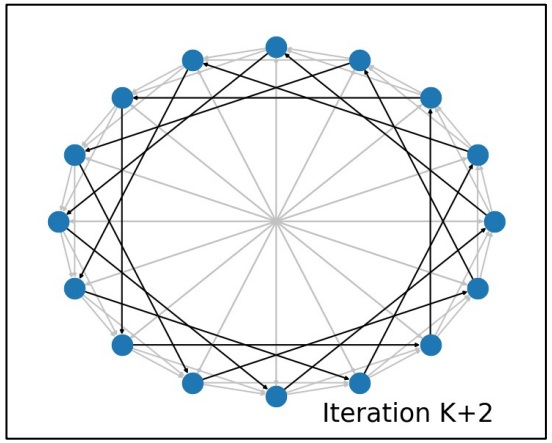
- Incurs $O(1)$ comm. overhead on **sparse** topologies; much less than global average $O(n)$



Global averaging over centralized network

comm. overhead $O(n)$





Partial averaging over ring or grid

comm. overhead $O(1)$

Decentralized Stochastic Bilevel Optimization

北京大学 PEKING UNIVERSITY

- A real experiment on a 256-GPUs cluster [1]

Model	Ring-Allreduce	Partial average
ResNet-50 (25.5M)	278 ms	150 ms

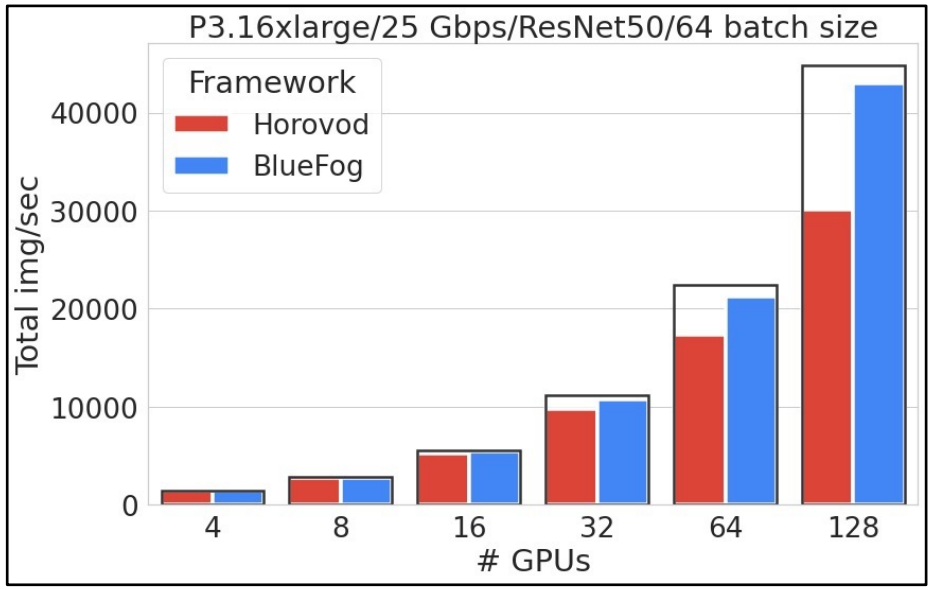
Table. Comparison of per-iter comm. time in terms of runtime with 256 GPUs

- DSGD saves more communications per iteration for larger models

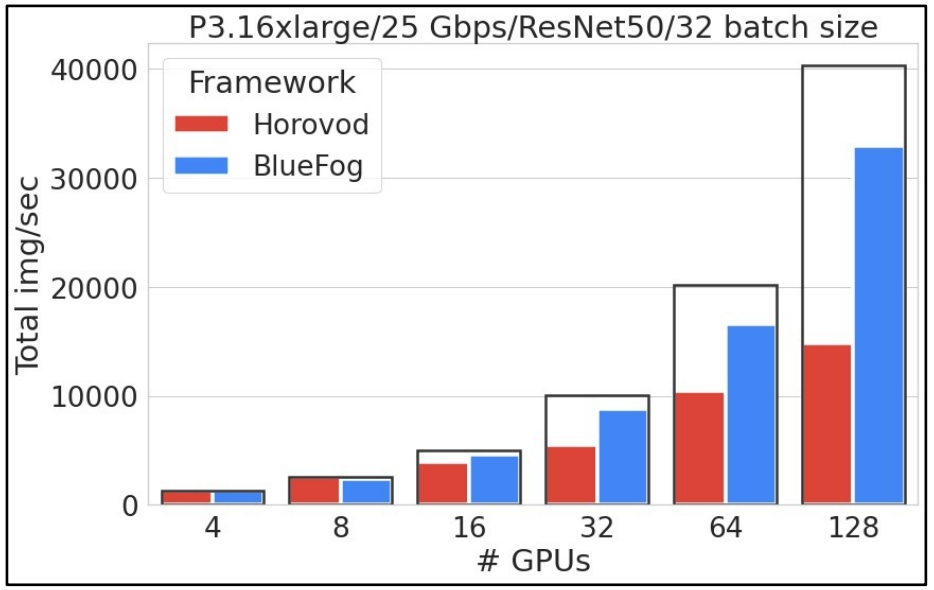
[1] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, ``Accelerating Gossip SGD with Periodic Global Averaging", ICML 2021

Decentralized Stochastic Bilevel Optimization

- DSGD (BlueFog) has **better scalability** than PSGD (Horovod) due to its small comm. overhead



Small comm.-to-compt. ratio



Large comm.-to-compt. ratio

- **This gives us strong motivation to develop decentralized bilevel algorithms**

B. Ying, K. Yuan, H. Hu, Y. Chen and W. Yin, "BlueFog: Make decentralized algorithms practical for optimization and deep learning", arXiv: 2111. 04287, 2021

Challenges to develop decentralized bilevel method

$$\min_{x\in\mathbb{R}^d} \quad \Phi(x) := f(x, y^*(x)) = \frac{1}{N}\sum_{i=1}^{N} f_i(x, y^*(x)) \qquad \text{s.t.} \quad y^*(x) := \arg\min_{y\in\mathbb{R}^p} \left\{ g(x,y) := \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \right\}$$

- The hyper-gradient $\nabla\Phi(x)$ can be computed by:

$$\nabla\Phi(x) = \nabla_1 f(x, y^*(x)) - \nabla_{12}^2 g(x, y^*(x)) \left[\nabla_{22}^2 g(x, y^*(x))\right]^{-1} \nabla_2 f(x, y^*(x))$$

- Main challenge: the Hessian inversion cannot be accessed easily

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

  - Matrix comm. is very expensive
  - Cannot be easily accessed through decentralized partial averaging due to the inversion operator

PART 03

D-SOBA: A single-loop DSBO algorithm

SOBA: Introduce an auxiliary variable

北京大学 PEKING UNIVERSITY

- **Stochastic One-loop Bilevel Algorithm (SOBA)** addresses Hessian-inverse estimation [1].

- In SBO hyper-gradient, we have:

$$\nabla\Phi(x) = \nabla_1 f(x, y^*(x)) - \nabla_{12}^2 g(x, y^*(x)) \left[\nabla_{22}^2 g(x, y^*(x))\right]^{-1} \nabla_2 f(x, y^*(x))$$

- **To remove Hessian inverse**, SOBA introduce an auxiliary variable z:

$$z^*(x) = \left[\nabla_{22}^2 g(x, y^*(x))\right]^{-1} \nabla_2 f(x, y^*(x))$$

which is the solution to the following problem:

$$\min_z \left[h(x, y^\star(x), z) = \frac{1}{2} z^\top \nabla_{22}^2 g(x, y^*(x)) z - z^\top \nabla_2 f(x, y^*(x)) \right]$$

[1] M. Dagréou, et. al., *A framework for bilevel optimization that enables stochastic and global variance reduction algorithms*. NeurIPS 2022.

SOBA: The algorithm framework

$$\min_{x\in\mathbb{R}^d} \quad \Phi(x) := f(x, y^*(x)) = \frac{1}{N}\sum_{i=1}^{N} f_i(x, y^*(x)) \qquad \text{s.t.} \quad y^*(x) := \arg\min_{y\in\mathbb{R}^p} \left\{ g(x,y) := \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \right\}$$

Exact version

$$y^\star(x) = \arg\min_y g(x,y)$$

$$z^\star(x) = \arg\min_y \; h(x, y^\star(x), z)$$

$$\nabla\Phi(x) = \nabla_1 f(x, y^*(x)) - \nabla^2_{12} g(x, y^*(x)) z^\star(x)$$

$$x^+ = x - \alpha\nabla\Phi(x)$$

No Hessian inversion

Gradient descent version

$$y^+ = y - \beta\nabla_2 g(x,y) \qquad (K\times)$$

$$z^+ = z - \gamma\nabla_3 h(x, y^+, z) \qquad (K\times)$$

$$\nabla\Phi(x) = \nabla_1 f(x, y^+) - \nabla^2_{12} g(x, y^+) z^+$$

$$x^+ = x - \alpha\nabla\Phi(x) \qquad (1\times)$$

No Hessian inversion

SOBA: Single-Loop!

$$y^+ = y - \beta\nabla_2 g(x,y) \qquad (1\times)$$

$$z^+ = z - \gamma\nabla_3 h(x, y^+, z) \qquad (1\times)$$

$$\nabla\Phi(x) = \nabla_1 f(x, y^+) - \nabla^2_{12} g(x, y^+) z^+$$

$$x^+ = x - \alpha\nabla\Phi(x) \qquad (1\times)$$

No Hessian inversion

M. Dagréou, et. al., *A framework for bilevel optimization that enables stochastic and global variance reduction algorithms*. NeurIPS 2022.

Centralized SOBA

$$\min_z \left[h(x, y^\star(x), z) = \frac{1}{2} z^\top \nabla_{22}^2 g(x, y^*(x)) z - z^\top \nabla_2 f(x, y^*(x)) \right]$$

北京大学 PEKING UNIVERSITY

- Recall that both the upper and lower level cost function are

$$f(x, y^*(x)) = \frac{1}{N} \sum_{i=1}^{N} f_i(x, y^*(x)) \qquad g(x, y) := \frac{1}{N} \sum_{i=1}^{N} g_i(x, y)$$

- Furthermore, the auxiliary cost function can be written into a finite-sum problem

$$h(x, y^*(x), z) = \frac{1}{N} \sum_{i=1}^{N} h_i(x, y^*(x), z)$$

$$\text{where} \quad h_i(x, y^*(x), z) := \frac{1}{2} z^\top \nabla_{22}^2 g_i(x, y^*(x)) z - z^\top \nabla_2 f_i(x, y^*(x))$$

B. Kong et. al., *Decentralized Bilevel Optimization over Graphs: Loopless Algorithmic Update and Transient Iteration Complexity*. arXiv 2024.

Centralized SOBA

北京大学 PEKING UNIVERSITY

SOBA: Single-Loop!

$$y^+ = y - \beta \nabla_2 g(x, y) \qquad (1\times)$$

$$z^+ = z - \gamma \nabla_3 h(x, y^+, z) \qquad (1\times)$$

$$\nabla \Phi(x) = \nabla_1 f(x, y^+) - \nabla^2_{12} g(x, y^+) z^+$$

$$x^+ = x - \alpha \nabla \Phi(x) \qquad (1\times)$$

No Hessian inversion

Centralized SOBA

$$x^{(t+1)} = x^{(t)} - \frac{\alpha_t}{N} \sum_{i=1}^{N} D^{(t)}_{x,i}, \qquad (1\times)$$

$$y^{(t+1)} = y^{(t)} - \frac{\beta_t}{N} \sum_{i=1}^{N} D^{(t)}_{y,i}, \qquad (1\times)$$

$$z^{(t+1)} = z^{(t)} - \frac{\gamma_t}{N} \sum_{i=1}^{N} D^{(t)}_{z,i}. \qquad (1\times)$$

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

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

Decentralized SOBA

北京大学 PEKING UNIVERSITY

Centralized SOBA

$$x^{(t+1)} = x^{(t)} - \frac{\alpha_t}{N}\sum_{i=1}^{N} D_{x,i}^{(t)}, \quad (1\times)$$

$$y^{(t+1)} = y^{(t)} - \frac{\beta_t}{N}\sum_{i=1}^{N} D_{y,i}^{(t)}, \quad (1\times)$$

$$z^{(t+1)} = z^{(t)} - \frac{\gamma_t}{N}\sum_{i=1}^{N} D_{z,i}^{(t)}. \quad (1\times)$$

Decentralized SOBA (D-SOBA)

$$x_i^{(t+1)} = \sum_{j=1}^{N} w_{ij}\left(x_j^{(t)} - \alpha_t D_{x,i}^{(t)}\right), \quad (1\times)$$

$$y_i^{(t+1)} = \sum_{j=1}^{N} w_{ij}\left(y_j^{(t)} - \alpha_t D_{y,i}^{(t)}\right), \quad (1\times)$$

$$z_i^{(t+1)} = \sum_{j=1}^{N} w_{ij}\left(z_j^{(t)} - \alpha_t D_{z,i}^{(t)}\right). \quad (1\times)$$

$$D_{x,i}^{(t)}(x,y,z) = \nabla_1 f_i(x^{(t)},y^{(t)}) - \nabla_{12}^2 g_i(x^{(t)},y^{(t)})z^{(t)},$$

$$D_{y,i}^{(t)}(x,y,z) = \nabla_2 g_i(x^{(t)},y^{(t)}), \quad D_{y,i}^{(t)}(x,y,z) = \nabla_{22}^2 g_i(x^{(t)},y^{(t)})z^{(t)} - \nabla_2 f_i(x^{(t)},y^{(t)}).$$

B. Kong et. al., *Decentralized Bilevel Optimization over Graphs: Loopless Algorithmic Update and Transient Iteration Complexity*. arXiv 2024.

D-SOBA (the stochastic version)

- By **introducing the decentralized** communication to centralized SOBA, we present the following **D-SOBA algorithm**:

Algorithm 1: D-SOBA

Initialize $x^{(0)} = y^{(0)} = z^{(0)} = h^{(0)} = 0$, the values of $\alpha_t, \beta_t, \gamma_t$ are set properly

for $t = 0, 1, \cdots, T-1$ **do**

  for *each node* $i = 1, 2, \cdots, N$ *in parallel* **do**

$$
\begin{aligned}
x_i^{(t+1)} &:= \textstyle\sum_{j\in\mathcal{N}_i} w_{ij}(x_j^{(t)} - \alpha_t h_j^{(t)});\\
y_i^{(t+1)} &:= \textstyle\sum_{j\in\mathcal{N}_i} w_{ij}(y_j^{(t)} - \beta_t v_j^{(t)});\\
\hat{z}_i^{(t+1)} &:= z_i^{(t)} - \gamma_t(H_i^{(t)} z_i^{(t)} - u_{i,y}^{(t)});\\
z_i^{(t+1)} &:= \textstyle\sum_{j\in\mathcal{N}_i} w_{ij}\hat{z}_j^{(t+1)};\\
\omega_i^{(t+1)} &:= u_{i,x}^{(t)} - J_i^{(t)} z_i^{(t)};\\
h_i^{(t+1)} &:= (1-\theta_t)h_i^{(t)} + \theta_t \omega_i^{(t+1)}.
\end{aligned}
$$

$$
\begin{aligned}
&\xi_i^{(t)} \sim \mathcal{D}_{f_i},\ \zeta_i^{(t)} \sim \mathcal{D}_{g_i},\\
&u_{i,x}^{(t)} := \nabla_1 F(x_i^{(t)}, y_i^{(t)}; \xi_i^{(t)}),\\
&u_{i,y}^{(t)} := \nabla_2 F(x_i^{(t)}, y_i^{(t)}; \xi_i^{(t)}),\\
&v_i^{(t)} := \nabla_2 G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)}),\\
&J_i^{(t)} := \nabla_{12}^2 G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)}),\\
&H_i^{(t)} := \nabla_{22}^2 G(x_i^{(t)}, y_i^{(t)}; \zeta_i^{(t)}).
\end{aligned}
$$

D-SOBA Convergence: Assumptions

北京大学 PEKING UNIVERSITY

Assumption 1 (Smoothness)

Each function $\nabla f_i, \nabla g_i, \nabla^2 g_i$ are Lipschitz continuous, $g_i(x,\cdot)$ is μ_g-strongly convex for any $x \in \mathbb{R}^d$, and $\left\|\nabla_2 f_i(x, y^2(x))\right\| \leq L_f < \infty$ for all $x \in \mathbb{R}^d$.

Assumption 2 (Bounded Gradient Dissimilarity)

There exists $b > 0$ such that for any $(x,y) \in \mathbb{R}^d \times \mathbb{R}^p$:

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla_1 f_i(x,y) - \nabla_1 f(x,y)\|^2 \leq b^2, \quad \frac{1}{N}\sum_{i=1}^{N}\|\nabla_2 g_i(x,y) - \nabla_2 g(x,y)\|^2 \leq b^2, \quad \frac{1}{N}\sum_{i=1}^{N}\|\nabla_{22} g_i(x,y) - \nabla_{22} g(x,y)\|^2 \leq b^2.$$

Assumption 3 (Stochasticity)

There exists $\sigma > 0$ such that for any $(x,y) \in \mathbb{R}^d \times \mathbb{R}^p$:

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_{f_i}}[\nabla_1 F(x,y;\xi_i)] = \nabla_1 f_i(x,y) \quad \mathbb{E}_{\xi_i \sim \mathcal{D}_{f_i}}\|\nabla_1 F(x,y;\xi_i) - \nabla_1 f_i(x,y)\|^2 \leq \sigma^2 \quad \mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}[\nabla^2_{12} G(x,y;\zeta_i)] = \nabla^2_{12} g_i(x,y) \quad \mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}\|\nabla^2_{12} G(x,y;\zeta_i) - \nabla^2_{12} g_i(x,y)\|^2 \leq \sigma^2$$

$$\mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}[\nabla_1 G(x,y;\xi_i)] = \nabla_2 g_i(x,y) \quad \mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}\|\nabla_2 G(x,y;\zeta_i) - \nabla_2 g_i(x,y)\|^2 \leq \sigma^2 \quad \mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}[\nabla^2_{22} G(x,y;\zeta_i)] = \nabla^2_{22} g_i(x,y) \quad \mathbb{E}_{\zeta_i \sim \mathcal{D}_{g_i}}\|\nabla^2_{22} G(x,y;\zeta_i) - \nabla^2_{22} g_i(x,y)\|^2 \leq \sigma^2$$

D-SOBA Convergence

北京大学 PEKING UNIVERSITY

Theorem 1

Suppose Assumption 1, 2, and 3 hold, and the mixing matrix W is doubly-stochastic. Then there exist $c_1, c_2, c_3 > 0, \alpha_t \equiv \Theta\left((N/T)^{1/2}\right)$, and $\beta_t \equiv c_1\alpha_t, \gamma_t \equiv c_2\alpha_t, \theta_t \equiv c_3\alpha_t$, such that the iteration of $\bar{x}^{(t)}$ in D-SOBA satisfies:

Clarify the influence of data heterogeneity: large hetero. hurts the rate

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

Clarify the influence of network topology: $\rho \to 1$ hurts the rate

Decentralized single-level stochastic gradient descent:

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

Bilevel structure does not deteriorate the order of the convergence rate!

The first result to show that decentralized bilevel method can match with single-level method

Comparison with existing algorithms

北京大学 PEKING UNIVERSITY

State-of-the-art complexity

Algorithm	Single loop	Asymptotic rate$^{\diamond}$	Asymptotic grad. complexity†	Transient complexity‡	Assumption$^{\triangleleft}$
DSBO[11]	✗	$\frac{1}{\sqrt{T}}$	$\frac{1}{\epsilon^3}$	N. A.	LC f_i
MA-DSBO[12]	✗	$\frac{1}{\sqrt{T}}$	$\frac{1}{\epsilon^2}\log(\frac{1}{\epsilon})$	N. A.	LC f_i
SLAM[40]	✗	$\frac{1}{\sqrt{NT}}$	$\frac{1}{N\epsilon^2}\log(\frac{1}{\epsilon})$	N. A.	LC f_i
Gossip DSBO[62]	✗	$\frac{1}{\sqrt{NT}}$	$\frac{1}{N\epsilon^2}\log(\frac{1}{\epsilon})$	$\frac{N^3G^4}{(1-\rho)^4}$ $^{\triangleright}$	BG ∇f_i
MDBO[22, Thm 1]*	✗	$\frac{1}{(1-\rho)\sqrt{T}}$	$\frac{1}{(1-\rho)^2\epsilon^2}\log(\frac{1}{\epsilon})$	N. A.	BG ∇f_i
MDBO[22, Thm 2]*	✗	$\frac{1}{\sqrt{NT}}$	$\frac{1}{N\epsilon^2}\log(\frac{1}{\epsilon})$	$\frac{N^3}{(1-\rho)^8}$	BG $\nabla f_i, \nabla g_i$
D-SOBA (ours)	✔	$\boldsymbol{\frac{1}{\sqrt{NT}}}$	$\boldsymbol{\frac{1}{N\epsilon^2}}$	$\boldsymbol{\max\left\{\frac{\rho^4N^3}{(1-\rho)^2}, \frac{\rho^4N^3b^2}{(1-\rho)^4}\right\}}$	**BGD** $\nabla f_i, \nabla g_i$
Single-level DSGD [14]	✔	$\frac{1}{\sqrt{NT}}$	$\frac{1}{N\epsilon^2}$	$\max\left\{\frac{\rho^4N^3}{(1-\rho)^2}, \frac{\rho^4N^3b^2}{(1-\rho)^4}\right\}$	BGD ∇f_i

D-SOBA Experiment 1: Hyper-parameter tuning in logistic regression

北京大学 PEKING UNIVERSITY

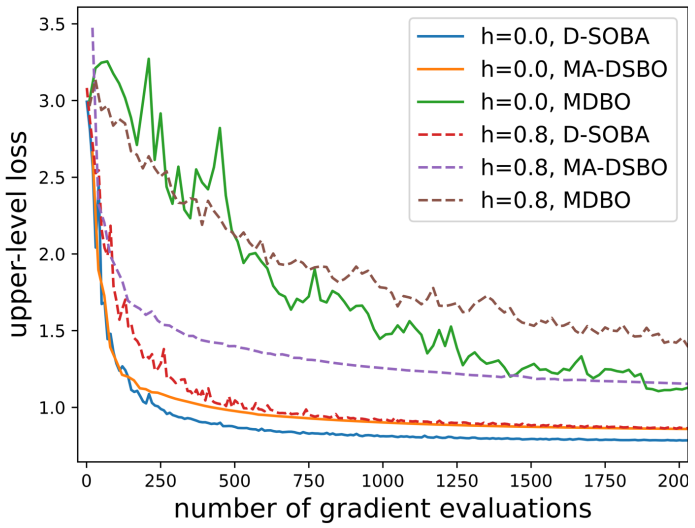
- Loss function on the i-th node:

$$f_i(x,y) = \frac{1}{|\mathcal{D}_{val}^{(i)}|} \sum_{(\xi,\zeta)\in\mathcal{D}_{val}^{(i)}} L(\xi^\top y, \zeta),$$

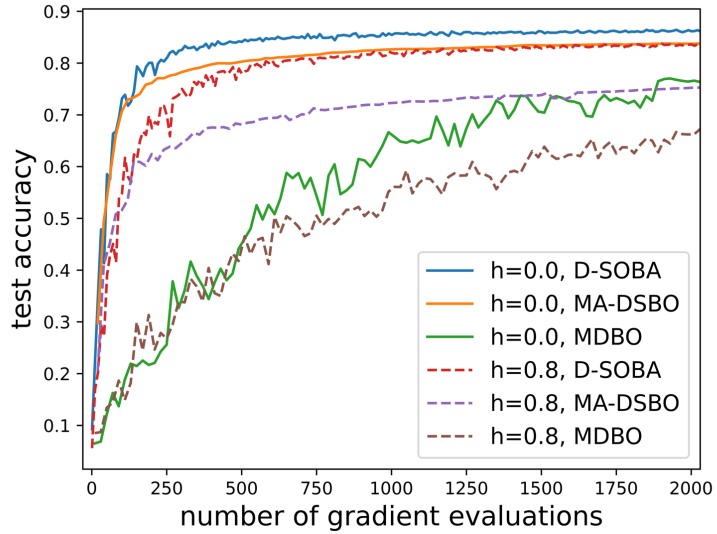
$$g_i(x,y) = \frac{1}{|\mathcal{D}_{tr}^{(i)}|} \sum_{(\xi,\zeta)\in\mathcal{D}_{tr}^{(i)}} L(\xi^\top y, \zeta) + \frac{1}{cp}\sum_{i=1}^{c}\sum_{j=1}^{p} e^{x_j} y_{ij}^2$$

- N=20, two different heterogeneity.
- Compare with MA-DSBO and MDBO.

upper-level loss



test accuracy



D-SOBA achieves a **faster convergence** and **higher test-accuracy** than the other two algorithms.

D-SOBA Experiment 2: Data hyper-cleaning for FashionMNIST dataset

北京大学 PEKING UNIVERSITY

- Loss function on the i-th node:

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

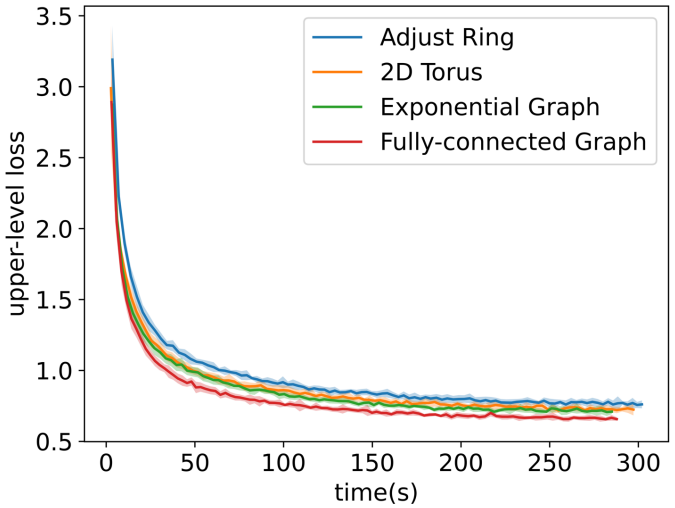
$$g_i(x,y) = \frac{1}{\left|\mathcal{D}_{tr}^{(i)}\right|} \sum_{(\xi_e,\zeta_e)\in \mathcal{D}_{tr}^{(i)}} \sigma(x_e) L(\phi(\xi_e;y),\zeta_e) + C\|y\|^2$$

  - ϕ: MLP with a 300-dim hidden layer and ReLU activation.

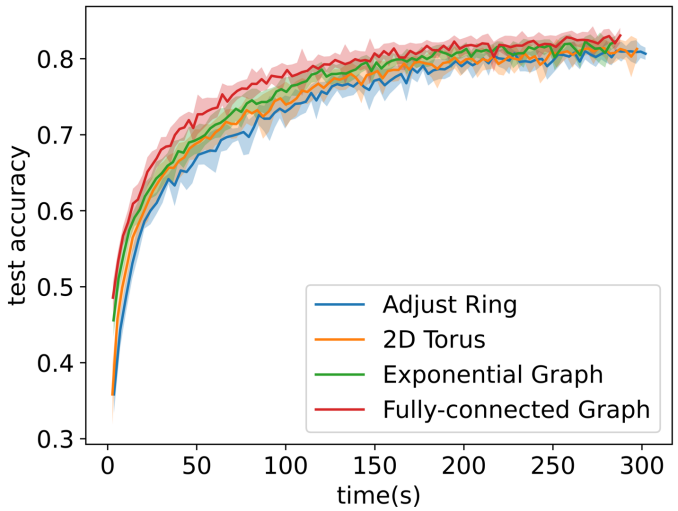
- N=10, non-i.i.d FashionMNIST dataset.

$$\rho_{\text{fully-connected}} < \rho_{\text{exp}} < \rho_{\text{torus}} < \rho_{\text{adjusted-ring}}$$

upper-level loss



test accuracy



Sparser topology leads to worse convergence performance.

北京大学
PEKING UNIVERSITY

The end of the story? Not Yet !

D-SOBA suffers from several drawbacks

北京大学 PEKING UNIVERSITY

- **Stringent assumption**

Assumption 2 (Bounded Gradient Dissimilarity)

There exists $b > 0$ such that for any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^p$:

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla_1 f_i(x,y) - \nabla_1 f(x,y)\|^2 \le b^2, \quad \frac{1}{N}\sum_{i=1}^{N}\|\nabla_2 g_i(x,y) - \nabla_2 g(x,y)\|^2 \le b^2, \quad \frac{1}{N}\sum_{i=1}^{N}\|\nabla_{22} g_i(x,y) - \nabla_{22} g(x,y)\|^2 \le b^2.$$

It does not hold even for simple quadratic functions $f_i(x,y) = x^\top A_i x + y^\top A_i y$

- Gradient dissimilarity/data heterogeneity **amplifies the influence of network topology**

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

Amplifies the influence of ρ

Open questions

北京大学 PEKING UNIVERSITY

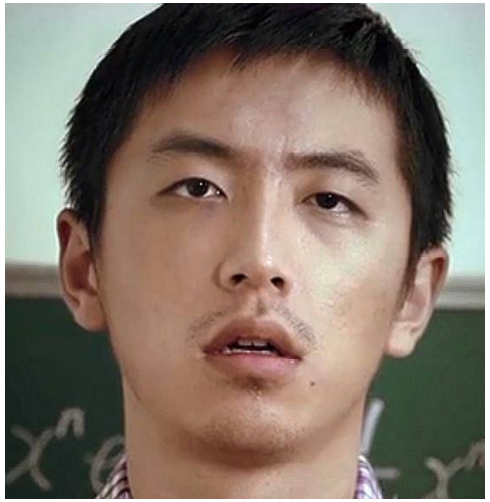
- Can we remove the influence of data heterogeneity?
- Can heterogeneity-correction techniques, originally designed for single-level optimization, be effectively applied to bilevel optimization problems? Which technique yields the best results?
- Can we use different decentralized algorithms across different optimization levels? What combination of the algorithms lead to best performance?
- Can we use different network topologies across different optimization levels? How to choose the network topology for each level?

SPARKLE: A unified framework

To address the above open questions,

- Should each heterogeneity-correction mechanism be explored one-by-one?
- Should we consider combining any two of these techniques to update the upper and lower-level problems, respectively?

This would involve an unbearable amount of effort !



SPARKLE: A unified framework

To address the above open questions,

- Should each heterogeneity-correction mechanism be explored one-by-one?
- Should we consider combining any two of these techniques to update the upper and lower-level problems, respectively?

This would involve an unbearable amount of effort !

Our solution: A **unified framework** that supports all existing mechanisms and their combinations!

SPARKLE: A Unified Single-Loop Primal-Dual Framework for DSBO

$$\min_{x_i \in \mathbb{R}^d} f(x_1, ..., x_n) = \frac{1}{n} \sum_{i=1}^{n} f_i(x_i), \quad \text{s.t.} \quad x_1 = ... = x_n.$$

- Warm up: A unified framework for single-level decentralized optimization [1]:

$$\mathbf{x}^{k+1} = \mathbf{C}\mathbf{x}^k - \alpha \mathbf{A}\mathbf{g}^k - \mathbf{B}\mathbf{d}^k, \quad \mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{B}\mathbf{d}^{k+1}.$$

Algorithms	**A**	**B**	**C**	The specific update rule at the k-th iteration for variable $\mathbf{s}$
ED	$\mathbf{W}$	$(\mathbf{I}-\mathbf{W})^{\frac{1}{2}}$	$\mathbf{W}^2$	$\mathbf{s}^{k+2} = \mathbf{W}\left(2\mathbf{s}^{k+1} - \mathbf{s}^k - \alpha\left(\mathbf{g}(\mathbf{s}^{k+1}) - \mathbf{g}(\mathbf{s}^k)\right)\right)$
EXTRA	$\mathbf{I}$	$(\mathbf{I}-\mathbf{W})^{\frac{1}{2}}$	$\mathbf{W}$	$\mathbf{s}^{k+2} = \mathbf{W}\left(2\mathbf{s}^{k+1} - \mathbf{s}^k\right) - \alpha\left(\mathbf{g}(\mathbf{s}^{k+1}) - \mathbf{g}(\mathbf{s}^k)\right)$
ATC-GT	$\mathbf{W}^2$	$\mathbf{I}-\mathbf{W}$	$\mathbf{W}^2$	$\mathbf{s}^{k+1} = \mathbf{W}\left(\mathbf{s}^k - \alpha\mathbf{h}^k\right)$, $\mathbf{h}^{k+1} = \mathbf{W}\left(\mathbf{h}^k + \mathbf{g}(\mathbf{s}^{k+1}) - \mathbf{g}(\mathbf{s}^k)\right)$
Semi-ATC-GT	$\mathbf{W}$	$\mathbf{I}-\mathbf{W}$	$\mathbf{W}^2$	$\mathbf{s}^{k+1} = \mathbf{W}\mathbf{s}^k - \alpha\mathbf{h}^k$, $\mathbf{h}^{k+1} = \mathbf{W}\left(\mathbf{h}^k + \mathbf{g}(\mathbf{s}^{k+1}) - \mathbf{g}(\mathbf{s}^k)\right)$
Non-ATC-GT	$\mathbf{I}$	$\mathbf{I}-\mathbf{W}$	$\mathbf{W}^2$	$\mathbf{s}^{k+1} = \mathbf{W}\mathbf{s}^k - \alpha\mathbf{h}^k$, $\mathbf{h}^{k+1} = \mathbf{W}\mathbf{h}^k + \mathbf{g}(\mathbf{s}^{k+1}) - \mathbf{g}(\mathbf{s}^k)$

S. Alghunaim, et.al., *Decentralized proximal gradient algorithms with linear convergence rates,* IEEE TAC 2020

SPARKLE: A Unified Single-Loop Primal-Dual Framework for DSBO

$$\min_{x\in\mathbb{R}^d} \quad \Phi(x) := f(x, y^*(x)) = \frac{1}{N}\sum_{i=1}^{N} f_i(x, y^*(x)) \qquad \text{s.t.} \quad y^*(x) := \arg\min_{y\in\mathbb{R}^p} \left\{ g(x,y) := \frac{1}{N}\sum_{i=1}^{N} g_i(x,y) \right\}$$

- The above bilevel problem can be decomposed into three pillar subproblems:

$$x^\star = \arg\min_{x\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} f_i(x, y^\star(x)), \qquad \text{upper level}$$

$$y^\star(x) = \arg\min_{y\in\mathbb{R}^q} \frac{1}{n}\sum_{i=1}^{n} g_i(x, y), \qquad \text{lower level}$$

$$z^\star(x) = \arg\min_{z\in\mathbb{R}^q} \frac{1}{n}\sum_{i=1}^{n} h_i(x, y^\star(x), z), \qquad \text{auxiliary level}$$

where $h_i(x, y^\star(x), z) := \frac{1}{2} z^\top \nabla^2_{22} g_i(x, y^\star(x)) z - z^\top \nabla_2 f_i(x, y^\star(x))$.

Use single-level decentralized algorithms to solve each level problem.

SPARKLE: A Unified Single-Loop Primal-Dual Framework for DSBO

北京大学 PEKING UNIVERSITY

- Apply the single-level framework to sub-problems at three optimization levels:

Algorithm 1 SPARKLE: A unified framework for decentralized stochastic bilevel optimization

Require: Initialize $\mathbf{x}^0 = \mathbf{y}^0 = \mathbf{z}^0 = \mathbf{r}^0 = \mathbf{0}$, $\mathbf{d}_x^0 = \mathbf{d}_y^0 = \mathbf{d}_z^0 = \mathbf{0}$, learning rate $\alpha_k, \beta_k, \gamma_k, \theta_k$.

for $k = 0, 1, \cdots, K-1$ **do**

$$\mathbf{y}^{k+1} = \mathbf{C}_y\mathbf{y}^k - \beta_k\mathbf{A}_y\mathbf{v}^k - \mathbf{B}_y\mathbf{d}_y^k, \quad \mathbf{d}_y^{k+1} = \mathbf{d}_y^k + \mathbf{B}_y\mathbf{y}^{k+1}; \quad \triangleright \text{ lower-level update}$$

$$\mathbf{z}^{k+1} = \mathbf{C}_z\mathbf{z}^k - \gamma_k\mathbf{A}_z\mathbf{p}^k - \mathbf{B}_z\mathbf{d}_z^k, \quad \mathbf{d}_z^{k+1} = \mathbf{d}_z^k + \mathbf{B}_z\mathbf{z}^{k+1}; \quad \triangleright \text{ auxiliary-level update}$$

$$\mathbf{r}^{k+1} = (1-\theta_k)\mathbf{r}^k + \theta_k\mathbf{u}^k; \quad \triangleright \text{ momentum update}$$

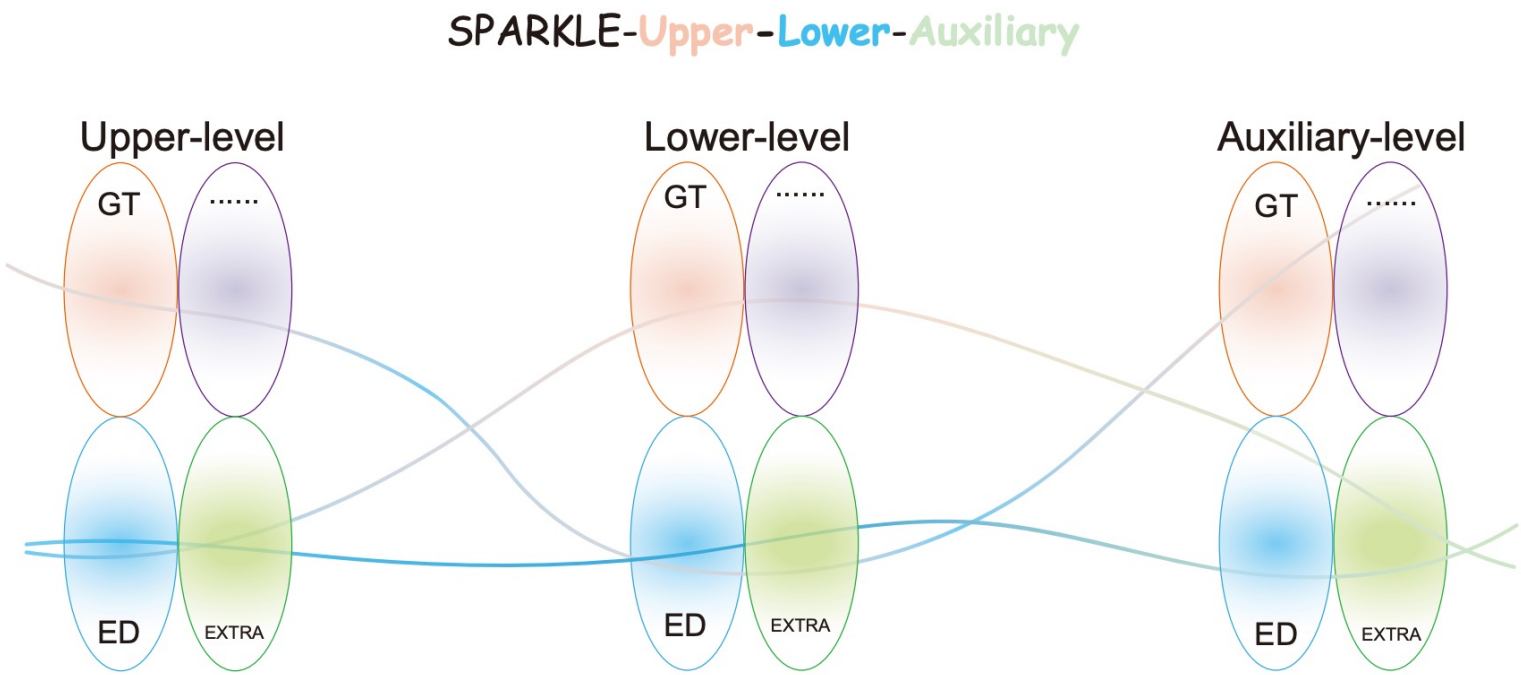
$$\mathbf{x}^{k+1} = \mathbf{C}_x\mathbf{x}^k - \alpha_k\mathbf{A}_x\mathbf{r}^{k+1} - \mathbf{B}_x\mathbf{d}_x^k, \quad \mathbf{d}_x^{k+1} = \mathbf{d}_x^k + \mathbf{B}_x\mathbf{x}^{k+1}; \quad \triangleright \text{ upper-level update}$$

end for

- Cover various DSBO algorithms by substituting different specific *A, B, C.*

- **SPARKLE-L-U**: SPARKLE using the decentralized mechanism **L** for the lower-level and auxiliary variables, and **U** for the upper-level. Abbreviated as **SPARKLE-L** if **L=U.**

SPARKLE: A Unified Single-Loop Primal-Dual Framework for DSBO



- For example, SPARKLE can utilize **GT** to update the upper-level variable x while using **ED** to update the auxiliary- and lower-level variables y and z, referred to as **SPARKLE-ED-GT.**

Unified convergence analysis: Assumptions

- **No influence from data heterogeneity**.

Theorem 1. *Under Assumptions 1 – 4, there exist proper constant step-sizes α, β, γ and momentum coefficient θ, such that the* SPARKLE *framework listed in Algorithm 1 will converge as follow:*

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}[\|\nabla\Phi(\bar{x}^k)\|^2] \lesssim \frac{\kappa^5\sigma}{\sqrt{nK}} + \kappa^{\frac{16}{3}}\left(\delta_{y,1}+\delta_{z,1}\right)\frac{\sigma^{\frac{2}{3}}}{K^{\frac{2}{3}}} + \kappa^{\frac{7}{2}}\delta_{x,1}\frac{\sigma^{\frac{1}{2}}}{K^{\frac{3}{4}}}$$
$$+\left(\kappa^{\frac{26}{5}}\delta_{y,2}+\kappa^6\delta_{z,2}\right)\frac{\sigma^{\frac{2}{5}}}{K^{\frac{4}{5}}} + \left(\kappa^{\frac{16}{3}}\delta_{y,3}+\kappa^{\frac{14}{3}}\delta_{z,3}+\kappa^{\frac{8}{3}}\delta_{x,3}\right)\frac{1}{K} + \left(\kappa C_\alpha+\kappa^4 C_\theta\right)\frac{1}{K},$$

where $\sigma \triangleq \max\{\sigma_{f,1},\sigma_{g,1},\sigma_{g,2}\}$, $\{\delta_{s,i}\}_{i=1}^3$ *are constants depending only on* $\mathbf{W}_s,\mathbf{A}_s,\mathbf{B}_s,\mathbf{C}_s$ *for* $s\in\{x,y,z\}$, *and* C_α, C_θ *are constants independent of* K.

One analysis **immediately applies to all SPARKLE variants** with distinct heterogeneity-correction technique. No need to analyze each variant one by one.

SPARKLE with the same decentralized mechanism

Corollary 1. *Under the same assumptions as in Theorem 1, the transient iteration complexity of* SPARKLE*—with the influence of κ and σ^2 omitted for brevity—is on the order of*

$$\max\left\{n^2\delta_x, n^3\delta_y, n^3\delta_z, n\hat{\delta}_x, n\hat{\delta}_y, n\hat{\delta}_z\right\}, \tag{8}$$

where $\delta_s, \hat{\delta}_s$ only depend $\mathbf{W}_s, \mathbf{A}_s, \mathbf{B}_s, \mathbf{C}_s$ for $s \in \{x, y, z\}$. Their values are in Lemma 18.

Corollary 2. *For* SPARKLE-ED *and* SPARKLE-EXTRA*, if we choose $\mathbf{W}_y = \mathbf{W}_z$, it holds that*

$$\delta_x = \mathcal{O}\left((1-\rho(\mathbf{W}_x))^{-2}\right), \qquad \delta_y = \delta_z = \mathcal{O}\left((1-\rho(\mathbf{W}_y))^{-2}\right),$$
$$\hat{\delta}_x = \mathcal{O}\left((1-\rho(\mathbf{W}_x))^{-\frac{3}{2}}\right), \qquad \hat{\delta}_y = \hat{\delta}_z = \mathcal{O}\left((1-\rho(\mathbf{W}_y))^{-2}\right).$$

Furthermore, if we choose $\mathbf{W}_x = \mathbf{W}_y = \mathbf{W}_z$ and denote $\rho \triangleq \rho(\mathbf{W}_x)$, the transient iteration complexity can be simplified as $n^3/(1-\rho)^2$.

Corollary 3. *For* SPARKLE-GT *and its variants with semi/non-ATC-GT, if we let $\mathbf{W}_y = \mathbf{W}_z$,*

$$\delta_x = \mathcal{O}\left((1-\rho(\mathbf{W}_x))^{-2}\right), \qquad \delta_y = \delta_z = \mathcal{O}\left((1-\rho(\mathbf{W}_y))^{-2}\right),$$
$$\hat{\delta}_x = \mathcal{O}\left((1-\rho(\mathbf{W}_x))^{-2}\right), \qquad \hat{\delta}_y = \hat{\delta}_z = \mathcal{O}\left((1-\rho(\mathbf{W}_y))^{-\frac{8}{3}}\right).$$

Furthermore, if we let $\mathbf{W}_x = \mathbf{W}_y = \mathbf{W}_z$ and denote $\rho \triangleq \rho(\mathbf{W}_x)$, the transient iteration complexity derived in (8) can be simplified as $\max\{n^3/(1-\rho)^2, n/(1-\rho)^{8/3}\}$.

SPARKLE with the different mechanisms across levels

北京大学 PEKING UNIVERSITY

Corollary 4. *For* SPARKLE-ED-GT *which uses ED to update* y *and* z *and GT to update* x*, if* $\mathbf{W}_x = \mathbf{W}_y = \mathbf{W}_z$ *and we denote* $\rho = \rho(\mathbf{W}_x)$*, it then holds that*

$$\delta_x = \delta_y = \delta_z = \mathcal{O}\left((1-\rho)^{-2}\right), \qquad \hat{\delta}_x = \hat{\delta}_y = \hat{\delta}_z = \mathcal{O}\left((1-\rho)^{-2}\right),$$

which implies that the transient iteration complexity in (8) *can be simplified as* $n^3/(1-\rho)^2$.

- Different **heterogeneity-correction strategies** across optimization levels:

lower \ upper	ED	EXTRA	GT
ED	$\frac{n^3}{(1-\rho)^2}$	$\frac{n^3}{(1-\rho)^2}$	$\frac{n^3}{(1-\rho)^2}$
EXTRA	$\frac{n^3}{(1-\rho)^2}$	$\frac{n^3}{(1-\rho)^2}$	$\frac{n^3}{(1-\rho)^2}$
GT	$\max\left\{\frac{n^3}{(1-\rho)^2}, \frac{n}{(1-\rho)^{8/3}}\right\}$	$\max\left\{\frac{n^3}{(1-\rho)^2}, \frac{n}{(1-\rho)^{8/3}}\right\}$	$\max\left\{\frac{n^3}{(1-\rho)^2}, \frac{n}{(1-\rho)^{8/3}}\right\}$

Mixed strategies outperform employing GT only!

SPARKLE with the different topologies across levels

北京大学 PEKING UNIVERSITY

- Different **topologies** across optimization levels:

Theoretical gap: *x* has **less impact** on transient iteration complexity than *y and z.*

↓

Enables a sparser communication graph for *x* while maintaining the same transient iteration complexity.

- An example: For SPARKLE-ED, the transient iteration complexity is

$$\max\{n^2(1-\rho(\mathbf{W}_x))^{-2}, n^3(1-\rho(\mathbf{W}_y))^{-2}\}$$

It retains $n^3(1-\rho(\mathbf{W}_y))^{-2}$ if $(1-\rho(\mathbf{W}_x))^{-1} \lesssim \sqrt{n}(1-\rho(\mathbf{W}_y))^{-1}$.

Unified convergence analysis: Comparison

Algorithms	Assumption▷	A. Rate. ◊	A. Comp. †	A. Comm. ‡	Tran. Iter.◁	Loopless
DSBO [9]	LC	$\frac{1}{\sqrt{K}}$	$\frac{1}{\varepsilon^3}$	$\left(pq\log(\frac{1}{\varepsilon})+\frac{q}{\varepsilon}\right)\frac{1}{\varepsilon^2}$	N.A.	No
MA-DSBO [10]	LC	$\frac{1}{\sqrt{K}}$	$\frac{1}{\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{q}{\varepsilon^2}\log(\frac{1}{\varepsilon})+\frac{p}{\varepsilon^2}$	N.A.	No
SLAM [33]	LC	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{p+q}{n\varepsilon^2}$	N.A.	No
MDBO [21]	BG	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{p+q}{n\varepsilon^2}$	$\frac{n^3}{(1-\rho)^8}$	No
Gossip DSBO [52]	BG	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}\log(\frac{1}{\varepsilon})$	$\frac{q^2}{n\varepsilon^2}\log(\frac{1}{\varepsilon})+\frac{pq}{n\varepsilon^2}$	$\frac{n^3}{(1-\rho)^4}$	No
LoPA [40]*	BGD	$\frac{1}{\sqrt{K}}$	$\frac{1}{\varepsilon^2}$	$\frac{p+q}{\varepsilon^2}$	N.A.	Yes
D-SOBA [29]	BGD	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}$	$\frac{p+q}{n\varepsilon^2}$	$\max\left\{\frac{n^3}{(1-\rho)^2},\frac{n^3b^2}{(1-\rho)^4}\right\}$	Yes
SPARKLE-GT (ours)	**None**	$\boldsymbol{\frac{1}{\sqrt{nK}}}$	$\boldsymbol{\frac{1}{n\varepsilon^2}}$	$\boldsymbol{\frac{ap+q}{n\varepsilon^2}}^{\sharp}$	$\boldsymbol{\max\left\{\frac{n^3}{(1-\rho)^2},\frac{n}{(1-\rho)^{8/3}}\right\}}$	**Yes**
SPARKLE-EXTRA (ours)	**None**	$\boldsymbol{\frac{1}{\sqrt{nK}}}$	$\boldsymbol{\frac{1}{n\varepsilon^2}}$	$\boldsymbol{\frac{ap+q}{n\varepsilon^2}}^{\sharp}$	$\boldsymbol{\frac{n^3}{(1-\rho)^2}}$	**Yes**
SPARKLE-ED (ours)	**None**	$\boldsymbol{\frac{1}{\sqrt{nK}}}$	$\boldsymbol{\frac{1}{n\varepsilon^2}}$	$\boldsymbol{\frac{ap+q}{n\varepsilon^2}}^{\sharp}$	$\boldsymbol{\frac{n^3}{(1-\rho)^2}}$	**Yes**
Single-level GT [2, 28]	None	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}$	$\frac{p}{n\varepsilon^2}$	$\max\left\{\frac{n^3}{(1-\rho)^2},\frac{n}{(1-\rho)^{8/3}}\right\}$	Yes
Single-level EXTRA [2]	None	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}$	$\frac{p}{n\varepsilon^2}$	$\frac{n^3}{(1-\rho)^2}$	Yes
Single-level ED [2]	None	$\frac{1}{\sqrt{nK}}$	$\frac{1}{n\varepsilon^2}$	$\frac{p}{n\varepsilon^2}$	$\frac{n^3}{(1-\rho)^2}$	Yes

$^{\sharp}$ $a>0$ measures the relative sparsity of the mixing weights $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{W}_z$, which can be very small in certain cases. Here $1-\rho$ in **Tran. Iter.** denotes the smallest spectral gap of $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{W}_z$.

Experiment 1: Data hyper-cleaning for FashionMNIST dataset

北京大学 PEKING UNIVERSITY

- Goal: Train a classifier from a corrupted dataset, where the label of each training data is replaced by a random class number with a probability *p* (the corruption rate)

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

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

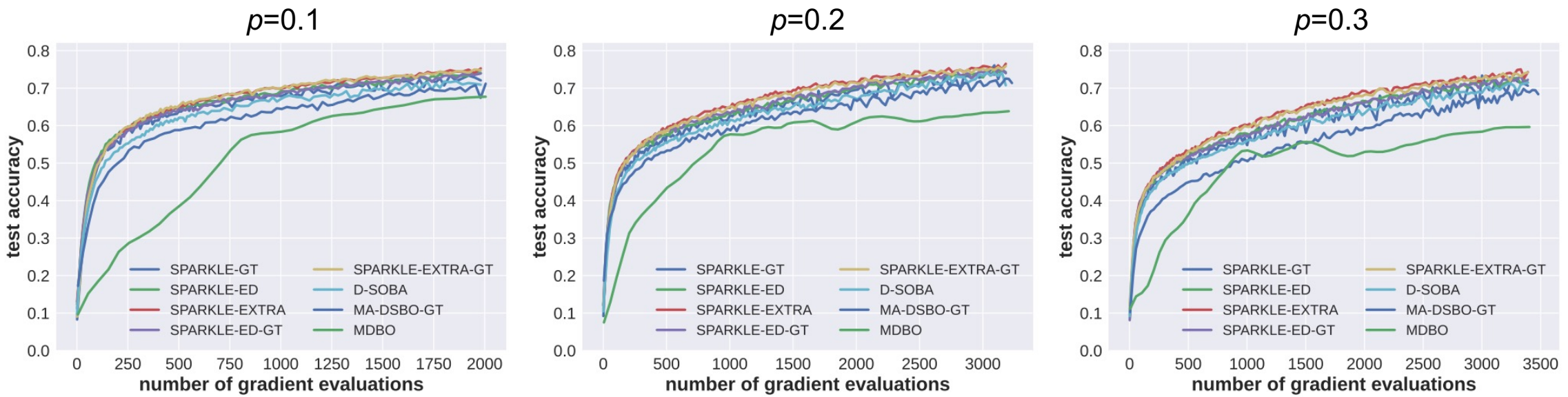
- ϕ : 2 layer MLP with parameters *y*.
- $\mathcal{D}_{tr}^{(i)}$: the training dataset of the *i*-th agent.
- $\mathcal{D}_{val}^{(i)}$: the validation dataset of the *i*-th agent.
- σ : the sigmoid function.

Higher *p* implies more severe data heterogeneity.

Experiment 1: Data hyper-cleaning for FashionMNIST dataset

北京大学 PEKING UNIVERSITY

- Comparing SPARKLE variants with other baseline methods across different data heterogeneity:

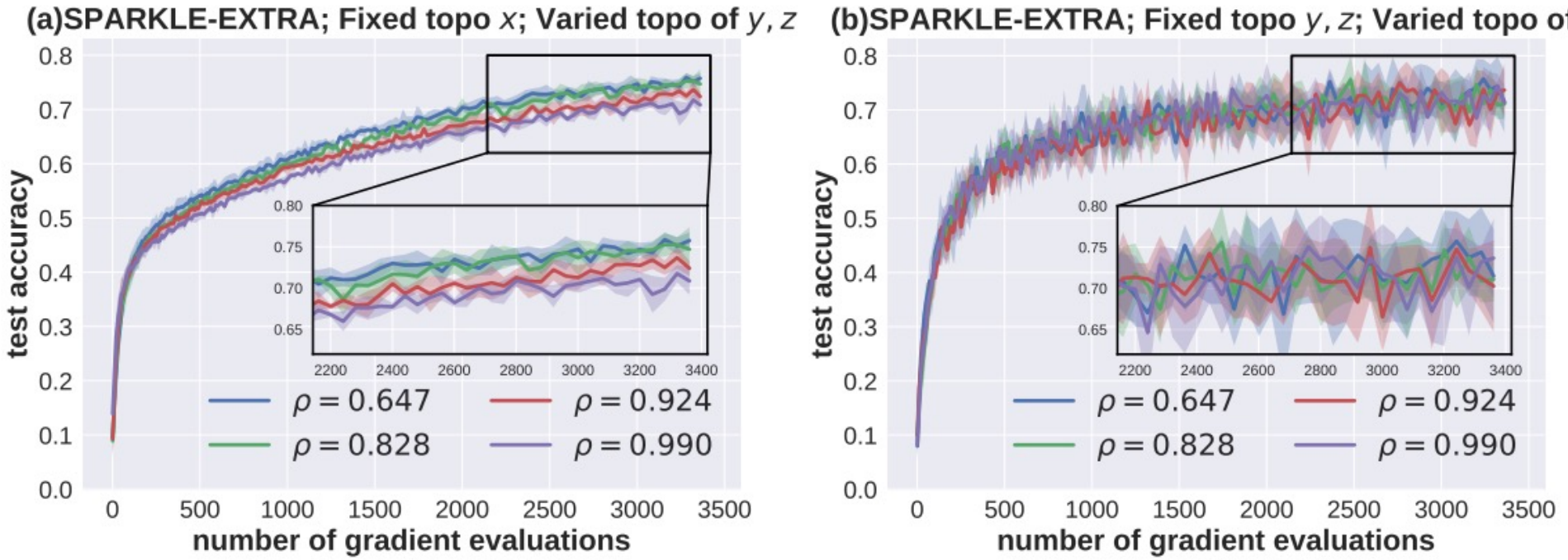


- SPARKLE with **ED/EXTRA** communication can achieve higher test accuracy than **GT**.

Experiment 1: Data hyper-cleaning for FashionMNIST dataset

北京大学 PEKING UNIVERSITY

- The influence of using different topologies across optimization levels:



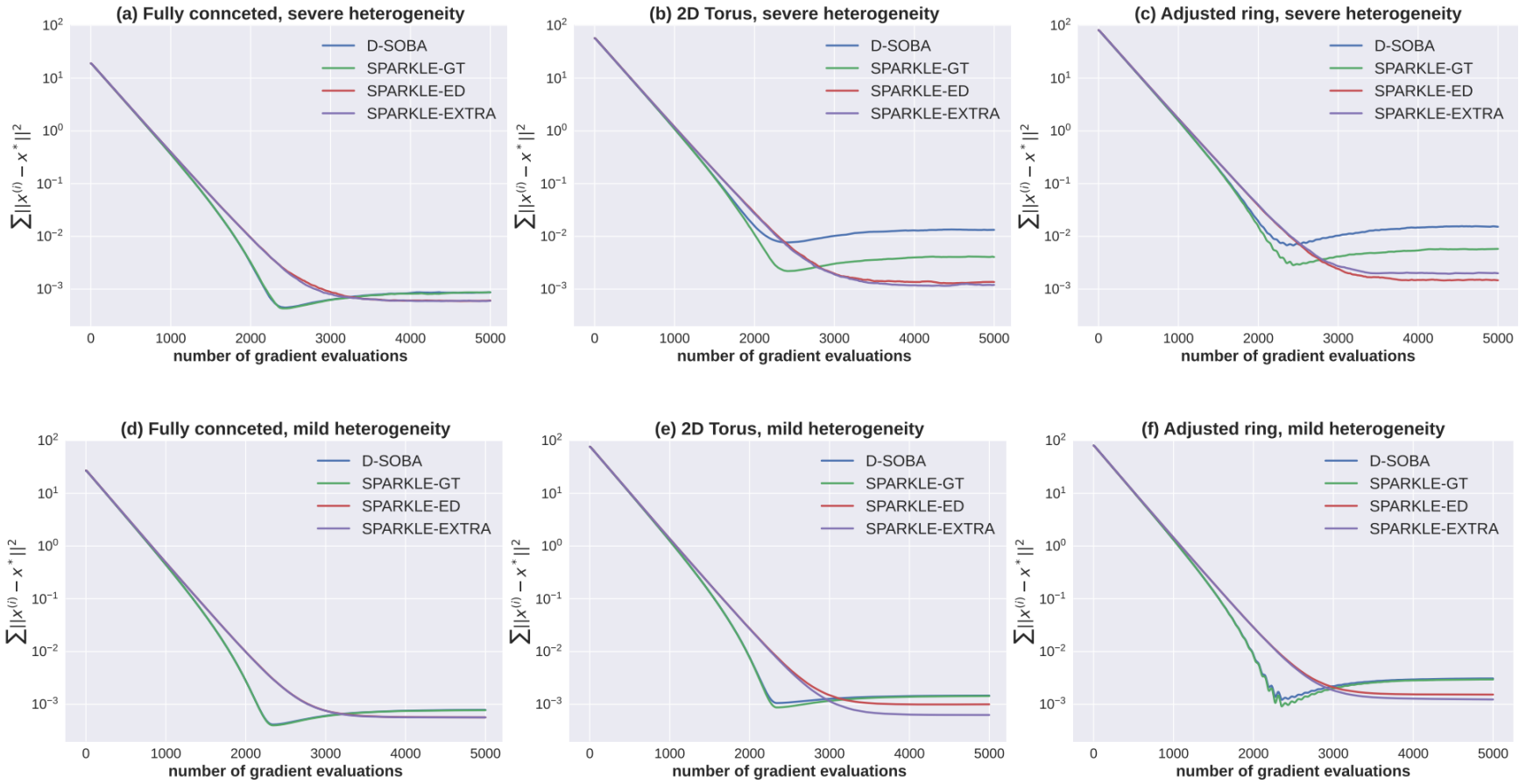
- The sparsity of communication topologies of *y, z* has more influence on the convergence than that of *x.*

Experiment 2: Synthetic linear regression

- Robustness to **data heterogeneity** across different topologies:

$$f_i(x,y) = \mathbb{E}_{A_i,b_i}\left[\|A_i y - b_i\|^2\right],$$
$$g_i(x,y) = \mathbb{E}_{A_i,b_i}\left[\|A_i y - x\|^2 + C_r\|y\|^2\right],$$

$$\rho_{\text{fully-connected}} < \rho_{\text{torus}} < \rho_{\text{ring}}$$



Experiment 3: Distributed policy evaluation in reinforcement learning

北京大学 PEKING UNIVERSITY

- Bellman minimization problem:

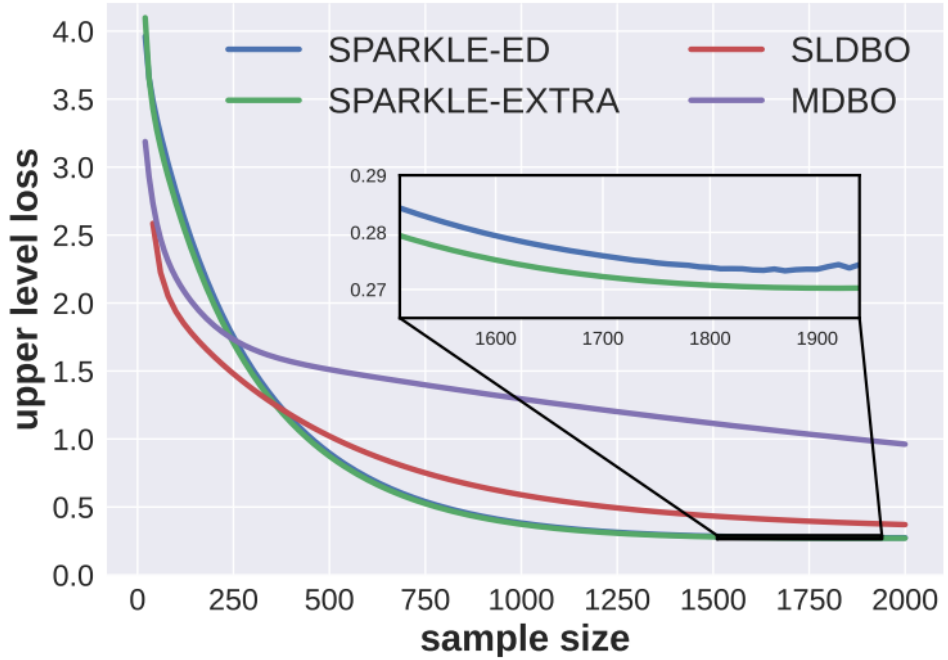
$$\min_{x\in\mathbb{R}^m} \quad F(x)=\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2|\mathcal{S}|}\sum_{s\in\mathcal{S}}\left(\phi_s^\top x-\mathbb{E}_{s'}\left[r^i(s,s')+\gamma\boxed{\phi_{s'}^\top x}\middle|s\right]\right)^2\right]$$

value function

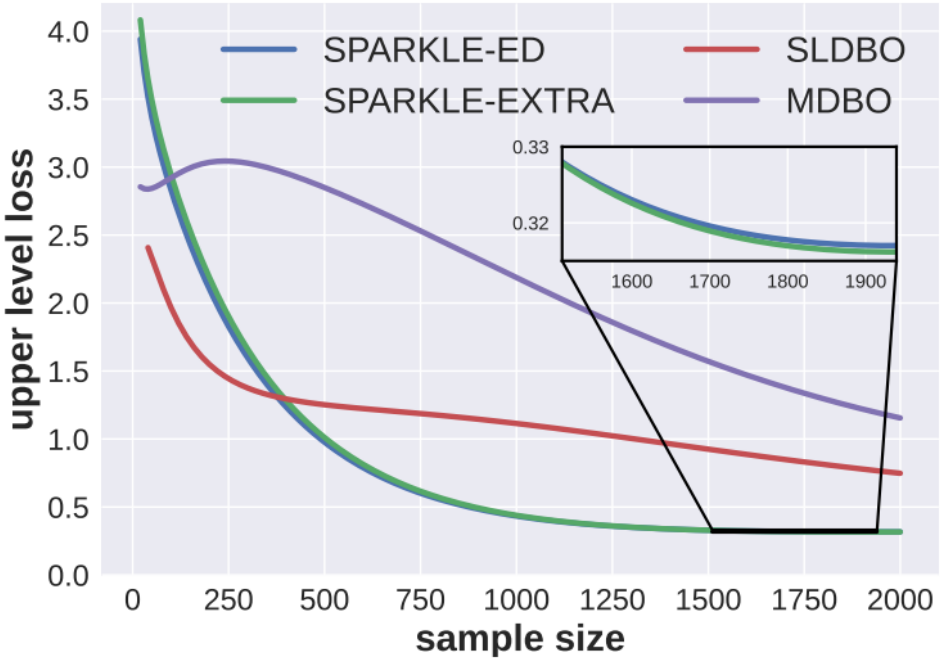
$$f_i(x,y)=\frac{1}{2|\mathcal{S}|}\sum_{s\in\mathcal{S}}(\phi_s^\top x-y_s)^2,$$

$$g_i(x,y)=\sum_{s\in\mathcal{S}}\left(y_s-\mathbb{E}_{s'}\left[r^i(s,s')+\gamma\phi_{s'}^\top x\middle|s\right]\right)^2,$$

n=10



n=20



Summary

We propose SPARKLE: a unified single-loop primal-dual framework for DSBO.

- Provides a unified and sharp convergence analysis, achieving SOTA **transient iteration complexity** and **communication efficiency** under mild assumptions.
- Encompasses a **broad range** of DSBO algorithms, including various heterogeneity-correction schemes, with **versatility** in decentralized strategies and adaptability across optimization levels.
- Demonstrates that the **lower-level** is **more influential** than the upper level in determining convergence performance, allowing for reduced communication costs at the upper level.
- **Recovers single-level** decentralized optimization while maintaining optimal theoretical results.
- Numerical results **verify** our theoretical findings.

北京大学
PEKING UNIVERSITY

Thank you!