Machine Learning (2026) 115:101
https://doi.org/10.1007/s10994-026-07040-y

On the Communication Complexity of Decentralized Stochastic Bilevel Optimization

Yihan Zhang[1] · **My T. Thai**[2] · **Jie Wu**[1] · **Hongchang Gao**[1]

Received: 11 October 2025 / Revised: 10 March 2026 / Accepted: 30 March 2026 /
Published online: 20 April 2026

Abstract
Stochastic bilevel optimization finds widespread applications in machine learning, including meta-learning, hyperparameter optimization, and neural architecture search. To extend stochastic bilevel optimization to distributed data, several decentralized stochastic bilevel optimization algorithms have been developed. However, existing methods often suffer from slow convergence rates and high communication costs in heterogeneous settings, limiting their applicability to real-world tasks. To address these issues, we propose two novel decentralized stochastic bilevel gradient descent algorithms based on simultaneous and alternating update strategies. Our algorithms can achieve faster convergence rates and lower communication costs than existing methods. Importantly, our convergence analyses do not rely on strong assumptions regarding heterogeneity. More importantly, our theoretical analyses clearly disclose how the computation and communication regarding the Hessian-inverse-vector product under the heterogeneous setting affects the convergence rate. To the best of our knowledge, this is the first time such favorable theoretical results have been achieved with mild assumptions in the heterogeneous setting. Furthermore, we demonstrate how to establish the convergence rate for the alternating update strategy when combined with the variance-reduced gradient. Finally, experimental results confirm the efficacy of our algorithms.

Keywords Decentralized optimization · Bilevel optimization · Communication complexity

1 Introduction

Bilevel optimization has been used in a wide range of machine learning models. For instance, the hyperparameter optimization Feurer and Hutter (2019); Franceschi et al. (2017), meta-learning (Franceschi et al., 2018; Rajeswaran et al., 2019), and neural architecture search (Liu et al., 2018) can be formulated as a bilevel optimization problem. Considering its

Editor: Haipeng Luo.

Extended author information available on the last page of the article

importance in machine learning, bilevel optimization has been attracting significant attention in recent years. Particularly, to handle the distributed data, parallel bilevel optimization has been actively studied in the past few years. In this paper, we are interested in an important class of parallel bilevel optimization: decentralized bilevel optimization, where all workers perform peer-to-peer communication to collaboratively optimize a bilevel optimization problem. Specifically, the loss function is defined as follows:

$$\min_{x\in\mathbb{R}^{d_x}} \frac{1}{K}\sum_{k=1}^{K} f^{(k)}(x, y^*(x)), \quad s.t. \quad y^*(x) = \arg\min_{y\in\mathbb{R}^{d_y}} \frac{1}{K}\sum_{k=1}^{K} g^{(k)}(x, y) \tag{1}$$

where k is the index of workers, $g^{(k)}(x,y) = \mathbb{E}_{\zeta\sim\mathcal{D}_g^{(k)}}[g^{(k)}(x,y;\zeta)]$ is the lower-level loss function of the k-th worker, $f^{(k)}(x,y) = \mathbb{E}_{\xi\sim\mathcal{D}_f^{(k)}}[f^{(k)}(x,y;\xi)]$ is the upper-level loss function of the k-th worker. Here, $\mathcal{D}_g^{(k)}$ and $\mathcal{D}_f^{(k)}$ denote the data distributions of the k-th worker. Throughout this paper, it is assumed that different workers have different data distributions.

In the past few years, a series of decentralized optimization algorithms have been proposed to solve Eq. (1). For instance, Gao et al. (2023) developed two decentralized stochastic bilevel gradient descent algorithms based on the momentum and variance reduction techniques: MDBO and VRDBO. Specifically, Gao et al. (2023) demonstrates that the variance-reduction based algorithm VRDBO is able to achieve the $O\left(\frac{1}{K\epsilon^{3/2}}\log\frac{1}{\epsilon}\right)$ convergence rate. More specifically, it is a double-loop algorithm, where there exists an inner loop to compute the Hessian inverse matrix for estimating stochastic hypergradient and the length of the inner loop is in the order of $O\left(\log\frac{1}{\epsilon}\right)$. Gao et al. (2023) shows that the inner loop does not require communication under the homogeneous setting. As such, its communication complexity[1] is in the order of $O\left(\frac{1}{K\epsilon^{3/2}}\right)$.

However, it is more challenging to solve Eq. (1) under the heterogeneous setting. Specifically, unlike the homogeneous setting where the local Jacobian and Hessian matrices are the same as the global ones, each worker under the heterogeneous setting has to pay extra efforts to estimate the global Jacobian and Hessian matrices to compute the hypergradient, which causes the following unique challenges for algorithmic design and theoretical analysis.

1.1 Heterogeneity Causes Challenges for Communication

Under the heterogeneous setting, each worker has to estimate the global Jacobian and Hessian matrices. This can incur a large communication cost, e.g., *a large number of communication rounds* and *a high communication cost per round*. For instance, existing methods (Chen et al., 2022a, b; Yang et al., 2022) suffer from a large number of communication rounds. Specifically, Chen et al. (2022a) developed a decentralized stochastic bilevel gradient descent algorithm: DSBO. It is a double-loop algorithm as VRDBO. However, DSBO requires communication in the inner loop. As such, the number of communication rounds is as large as its iteration complexity (i.e., convergence rate) $O\left(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$. Yang et al. (2022) proposed a decentralized stochastic bilevel gradient descent with momentum algorithm: Gossip-DSBO, which is also a double-loop algorithm and requires communication

[1] In the introduction, we ignore the spectral gap and the communication cost in each communication round to make it clear. The detailed communication complexity can be found in Table 1.

in the inner loop. The number of communication rounds and the iteration complexity can be slightly improved to $O\left(\frac{1}{K\epsilon^2}\log\frac{1}{\epsilon}\right)$. Chen et al. (2022b) proposed MA-DSBO, which is also a double-loop algorithm and shares the same number of communication rounds $O\left(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$ with DSBO. Moreover, most existing methods suffer from a high communication cost per communication round. Specifically, Gossip-DSBO and DSBO require to communicate the Hessian matrix or Jacobian matrix, which incurs a high communication cost $O(d_y^2)$ or $O(d_x d_y)$ in each round. As a result, the total communication complexity under the heterogeneous setting is much larger than the homogeneous setting.

It can be observed that all the aforementioned existing algorithms under the heterogeneous setting suffer from higher communication complexity compared to VRDBO under the homogeneous setting. This observation naturally leads to the first question: *Can we have a decentralized stochastic bilevel optimization algorithm under the heterogeneous setting to enjoy both a smaller number of communication rounds and a low communication cost per communication round?* Therefore, the first goal of this paper is to develop a communication-efficient decentralized stochastic bilevel optimization algorithm to tackle this challenge in algorithmic design.

1.2 Heterogeneity Causes Challenges for Convergence Analysis

When establishing the theoretical convergence rate, the aforementioned existing methods (Chen et al., 2022a; Yang et al., 2022; Chen et al., 2022b) require strong assumptions to bound the heterogeneity. Specifically, DSBO (Chen et al., 2022a) requires a Lipschitz continuous upper-level loss function regarding x, i.e., $\|\nabla_1 f^{(k)}(x,y)\| \leq c_{f_x}$, where $c_{f_x} > 0$ is a constant. Gossip-DSBO (Yang et al., 2022) also requires this assumption. Meanwhile, it requires that the lower-level loss function is Lipschitz continuous with respect to y, i.e., $\|\nabla_2 g^{(k)}(x,y)\| \leq c_{g_y}$, where $c_{g_y} > 0$ is a constant. With these strong assumptions, it is easy to bound the heterogeneity when establishing the convergence rate. However, it is worth noting that some of these strong assumptions might not hold. MA-DSBO (Chen et al., 2022b) does not require these strong assumptions at the cost of introducing the bounded heterogeneity assumption, i.e., $\|\nabla_2 g^{(k)}(x,y) - \nabla_2 g(x,y)\| \leq \delta$, where $\delta > 0$ is a constant. However, this assumption is also too strong to hold in practice (See Remark 3.6).

Moreover, communication under the heterogeneity setting introduces new challenges for convergence analysis. As discussed previously, additional communication is required to estimate the hypergradient compared to the homogeneous setting. Then, it is important to disclose how this additional communication operation affects the convergence rate. However, existing methods, including DSBO and MA-DSBO, fail to address this aspect, while the convergence rate of Gossip-DSBO is problematic due to contradictory assumptions (See Remark 3.5).

Then, the second natural question arises: *Can we establish the convergence rate of a communication-efficient decentralized stochastic bilevel optimization algorithm without strong assumptions and disclose how the additional communication operation affects the convergence rate under the heterogeneous setting?* Hence, our second goal is to establish the convergence rate under mild assumptions to address these theoretical analysis challenges.

1.3 Heterogeneity Causes Challenges for the Variable Update Strategy

Since there are two variables in stochastic bilevel optimization, there exist two strategies for updating variables: simultaneous update and alternating update. The former one updates x and y simultaneously, while the latter one updates x and y sequentially. The heterogeneity introduces more challenges for the alternating strategy. Specifically, assuming that the variables on different workers are the same in the t-th iteration, when updating x_t and y_t with the alternating strategy, y_t is updated to y_{t+1} first, and then x_t is updated with the gradient computed on y_{t+1}, rather than y_t as the simultaneous strategy. As a result, this gradient computed on y_{t+1} can be more heterogeneous than that computed on y_t in the simultaneous strategy, causing new challenges for convergence analysis. Although DSBO (Chen et al., 2022a) and MA-DSBO (Chen et al., 2022b) established the convergence rate for the alternating strategy, they restrict their focus on the unbiased stochastic gradient. As a result, it is unclear whether the alternating algorithm will converge when employing a biased variance-reduced gradient estimator (Cutkosky & Orabona, 2019). In particular, it can make the gradient across workers more heterogeneous due to the additional bias caused by this kind of gradient estimator. Specifically, the gradient estimation error regarding one variable can be affected by all the others, which is discussed in Section 5.2. All these issues make the convergence analysis more challenging for the alternating update strategy.

Then, the third natural question arises: *Can we establish the convergence rate of a communication-efficient decentralized bilevel optimization algorithm under mild conditions when employing the alternating update strategy and the variance-reduced gradient estimator?* Therefore, our third goal is to establish the convergence rate for the alternating update strategy to tackle these theoretical challenges.

In summary, to address the aforementioned challenges, this paper makes the following contributions.

- We have developed a novel decentralized stochastic bilevel gradient descent algorithm with a faster convergence rate based on the simultaneous update strategy and variance-reduced gradient estimators: DSVRBGD-S, which can reduce both the number of communication rounds and the cost in each communication round. In particular, the communication cost per round is only $O(d_x + d_y)$, and the number of communication rounds can be as small as $O(\frac{1}{K\epsilon^{3/2}})$, which can match the complexity of VRDBO under the homogeneous setting when ignoring other factors. To the best of our knowledge, this is the first method to achieve such a small communication complexity for decentralized stochastic bilevel optimization.

- We have established the theoretical convergence rate of our algorithm without relying on any strong heterogeneity assumptions. Furthermore, we have disclosed how the computation and communication regarding the Hessian-inverse-vector product affects the dependence of the convergence rate on the spectral gap of the communication topology. To the best of our knowledge, this is the first time such favorable theoretical results have been achieved. The detailed comparison between our algorithm and existing ones can be found in Table 1.
- We have developed a novel decentralized stochastic bilevel gradient descent algorithm, DSVRBGD-A, based on the alternating update strategy and *variance-reduced gradient estimators*. Similar to our first algorithm, this algorithm also enjoys a small commu-

Table 1 The comparison of the communication complexity between different algorithms under the homogeneous (IID) and heterogeneous (Non-IID) settings

Algorithms	R/IT	Cost/R	IT	Communication	Heterogeneity
MDBO (Gao et al., 2023)	$O(1)$	$O(d_x + d_y)$	$O\left(\frac{1}{\epsilon^2(1-\lambda)^2}\right)$	$O\left(\frac{d_x+d_y}{\epsilon^2(1-\lambda)^2}\right)$	i.i.d
VRDBO (Gao et al., 2023)	$O(1)$	$O(d_x + d_y)$	$O\left(\frac{1}{K\epsilon^{3/2}(1-\lambda)^2}\right)$	$O\left(\frac{d_x+d_y}{K\epsilon^{3/2}(1-\lambda)^2}\right)$	i.i.d
DSBO (Chen et al., 2022a)	$O\left(\log\frac{1}{\epsilon}\right)$	$O(d_x d_y)$	$O\left(\frac{1}{\epsilon^2(1-\lambda)^?}\right)$# a	$O\left(\frac{d_x d_y}{\epsilon^2(1-\lambda)^?}\log\frac{1}{\epsilon}\right)$	#c
Gossip-DSBO (Yang et al., 2022)	$O\left(\log\frac{1}{\epsilon}\right)$	$O(d_x d_y + d_y^2)$	$O\left(\frac{1}{K\epsilon^2}\right)$#b	$O\left(\frac{d_x d_y+d_y^2}{K\epsilon^2}\log\frac{1}{\epsilon}\right)$	#d
MA-DSBO (Chen et al., 2022b)	$O\left(\log\frac{1}{\epsilon}\right)$	$O(d_x + d_y)$	$O\left(\frac{1}{\epsilon^2(1-\lambda)^?}\right)$#a	$O\left(\frac{d_x+d_y}{\epsilon^2(1-\lambda)^?}\log\frac{1}{\epsilon}\right)$	#e
DSVRBGD-S (Ours)	$O(1)$	$O(d_x + d_y)$	$O\left(\frac{1}{K\epsilon^{3/2}(1-\lambda)^4}\right)$	$O\left(\frac{d_x+d_y}{K\epsilon^{3/2}(1-\lambda)^4}\right)$	-
DSVRBGD-A (Ours)	$O(1)$	$O(d_x + d_y)$	$O\left(\frac{1}{K\epsilon^{3/2}(1-\lambda)^4}\right)$	$O\left(\frac{d_x+d_y}{K\epsilon^{3/2}(1-\lambda)^4}\right)$	-

R/IT denotes the number of communication rounds in each iteration. Cost/R means the communication cost in each round. IT represents the iteration complexity. Communication denotes the communication complexity. #a: DSBO and MA-DSBO fail to provide the dependence on the spectral gap. #b: Gossip-DSBO assumes all gradients are upper bounded so that it eliminates the dependence on the spectral gap. #c: $\|\nabla_1 f^{(k)}\| \leq c_{f_x}$. #d: $\|\nabla_2 g^{(k)}\| \leq c_{g_y}$. #e: $\|\nabla_2 g^{(k)} - \nabla_2 g\| \leq \delta$. The limitations of the heterogeneity assumption of DSBO, Gossip-DSBO, and MA-DSBO are discussed in Remark 3.5 and Remark 3.6

nication complexity. Moreover, we have established its theoretical convergence rate. Remarkably, this marks the first time the convergence rate of the alternating variance-reduced gradient descent method for stochastic bilevel optimization has been established. Notably, even in the single-machine setting, such theoretical results do not exist. Therefore, our theoretical analysis strategy can also be extended to the single-machine setting, bridging an existing gap in the literature.

Finally, we conducted extensive experiments, and the experimental results confirm the efficacy of our proposed algorithms.

2 Related Works

2.1 Stochastic Bilevel Optimization

The main challenge in bilevel optimization lies in the computation of the hypergradient since it involves the Hessian inverse matrix. To address this issue, a commonly used approach is to leverage the Neumann series expansion technique to approximately compute Hessian

inverse. Based on the first approach, Ghadimi and Wang (2018) developed the bilevel stochastic approximation algorithm, where the lower-level problem is solved by stochastic gradient descent, and the upper-level problem is solved by stochastic hypergradient. As for the nonconvex-strongly-convex bilevel optimization problem, this algorithm achieves $O(\frac{1}{\epsilon^2})$ sample complexity (i.e., the gradient evaluation) for the upper-level problem and $O(\frac{1}{\epsilon^3})$ sample complexity for the lower-level problem. Later, Hong et al. (2020) developed a two-timescale stochastic approximation algorithm where different time scales are used for the upper-level and lower-level step sizes, whose sample complexities is $O(\frac{1}{\epsilon^{5/2}})$. Ji et al. (2021) proposed to employ the mini-batch stochastic gradient to improve both sample complexities to $O(\frac{1}{\epsilon^2})$. Chen et al. (2021) proposed an alternating stochastic bilevel gradient descent algorithm, which can also improve both sample complexities to $O(\frac{1}{\epsilon^2})$. Yang et al. (2021); Khanduri et al. (2021) leveraged the variance-reduced gradient estimators STORM (Cutkosky & Orabona, 2019) or SPIDER (Fang et al., 2018) to further improve the sample complexity to $O(\frac{1}{\epsilon^{3/2}})$. However, the Neumann series expansion based algorithm requires an inner loop to estimate Hessian inverse. As such, this class of algorithms suffers from a large Hessian-vector-product complexity.

Another commonly used approach for estimating Hessian inverse is to directly estimate the Hessian-inverse-vector product in the hypergradient. Specifically, it views the Hessian-inverse-vector product as the solution of a quadratic optimization problem and then employs the gradient descent algorithm to estimate it. For instance, under the finite-sum setting where the number of samples is finite, Dagréou et al. (2022) leveraged the variance-reduced gradient estimator SAGA (Defazio et al., 2014) to update the estimation of Hessian-inverse-vector product and two variables, providing the $O(\frac{(n+m)^{2/3}}{\epsilon})$ sample complexity where n and m are the number of samples in the upper-level and lower-level problems. Additionally, Dagréou et al. (2023) employs a SPIDER-like (Fang et al., 2018) gradient estimator to improve the sample complexity to $O(\frac{(n+m)^{1/2}}{\epsilon})$. Compared with the Neumann series expansion-based algorithm, this class of bilevel optimization algorithms does not need to use an inner loop to estimate Hessian inverse. Thus, they are more efficient in each iteration.

2.2 Decentralized Stochastic Bilevel Optimization

The decentralized bilevel optimization has been actively studied in the past few years. A series of algorithms have been proposed. For instance, under the homogeneous setting, Gao et al. (2023) developed a decentralized bilevel stochastic gradient descent with momentum algorithm, which can achieve $O(\frac{1}{\epsilon^2})$ communication complexity and can be improved to $O(\frac{1}{K\epsilon^2})$ when all gradients are upper bounded. Additionally, Gao et al. (2023) proposed a bilevel stochastic gradient descent algorithm based on the STORM (Cutkosky & Orabona, 2019) gradient estimator, which can achieve $O(\frac{1}{K\epsilon^{3/2}})$ communication complexity, even though not all gradients are upper bounded. Chen et al. (2022a) developed the decentralized bilevel full gradient descent and decentralized bilevel stochastic gradient descent algorithms under both homogeneous and heterogeneous settings. Yang et al. (2022) introduced the decentralized bilevel stochastic gradient descent with momentum algorithm under the heterogeneous setting, whose communication complexity can achieve linear speedup with

respect to the number of workers. All the aforementioned algorithms under the heterogeneous setting employ the Neumann series expansion approach to estimate Hessian inverse. As such, they suffer from a large communication complexity caused by the Neumann series expansion. Recently, Chen et al. (2022b) estimate the Hessian-inverse-vector product under the decentralized setting. However, it uses the standard stochastic gradient so that it needs to use an inner loop to estimate Hessian-inverse-vector product to reduce the estimation error. Thus, it still suffers from a large communication complexity and fails to achieve linear speedup.

Other than the decentralized bilevel optimization problem defined in Eq. (1), there exists another class of decentralized bilevel optimization problems, where $y^*(x)$ only depends on each local lower-level optimization problem rather than the global one. Without the global dependence, the hypergradient is much easier to estimate than that in Eq. (1). To address this class of decentralized bilevel optimization problems, Lu et al. (2022) developed a stochastic gradient-based algorithm, and Liu et al. (2022) leveraged the SPIDER (Fang et al., 2018) gradient estimator to update variables. Moreover, these also exist distributed bilevel optimization algorithms (Gao, 2022; Tarzanagh et al., 2022; Huang, 2022; Li et al., 2022) under the centralized setting, which are orthogonal to the decentralized setting. Recently, there appears a concurrent work (Dong et al., 2023), which focuses on the full gradient rather than the stochastic gradient.

3 Preliminaries

Assumption 3.1 For $\forall k$, $g^{(k)}(x,y)$ is μ-strongly convex with respect to y for fixed $x \in \mathbb{R}^{d_x}$ where $\mu > 0$ is a constant.

Assumption 3.2 For $\forall k$, $\forall (x,y) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, $\nabla_1 f^{(k)}(x,y;\xi)$ is ℓ_{f_x}-Lipschitz continuous where $\ell_{f_x} > 0$ is a constant, $\nabla_2 f^{(k)}(x,y;\xi)$ is ℓ_{f_y}-Lipschitz continuous where $\ell_{f_y} > 0$ is a constant, $\|\nabla_2 f^{(k)}(x,y;\xi)\| \leq c_{f_y}$ where $c_{f_y} > 0$ is a constant.

Assumption 3.3 For $\forall k$, $\forall (x,y) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, $\nabla_2 g^{(k)}(x,y;\zeta)$ is ℓ_{g_y}-Lipschitz continuous where $\ell_{g_y} > 0$ is a constant, $\nabla^2_{12} g^{(k)}(x,y;\zeta)$ is $\ell_{g_{xy}}$-Lipschitz continuous where $\ell_{g_{xy}} > 0$ is a constant, $\nabla^2_{22} g^{(k)}(x,y;\zeta)$ is $\ell_{g_{yy}}$-Lipschitz continuous where $\ell_{g_{yy}} > 0$ is a constant, $\|\nabla^2_{12} g^{(k)}(x,y;\zeta)\| \leq c_{g_{xy}}$ where $c_{g_{xy}} > 0$ is a constant, and $\mu \mathrm{I} \preceq \nabla^2_{22} g^{(k)}(x,y;\zeta) \preceq \ell_{g_y} \mathrm{I}$.

Assumption 3.4 All stochastic gradients have bounded variance σ^2 where $\sigma > 0$ is a constant.

Remark 3.5 Our assumptions regarding the gradient are milder than existing heterogeneous decentralized bilevel optimization algorithms (Chen et al., 2022a; Yang et al., 2022). In particular, they have additional assumptions: $\|\nabla_1 f^{(k)}(x,y)\| \leq c_{f_x}$ and $\|\nabla_2 g^{(k)}(x,y)\| \leq c_{g_y}$. The latter on does not hold for a strongly convex function as discussed in C.1 of Chen et al. (2022b).

Remark 3.6 Chen et al. (2022b) introduces the explicit heterogeneity assumption: $\|\nabla_2 g^{(k)}(x,y) - \nabla_2 g(x,y)\| \leq \delta$ where $\delta > 0$ is a constant. In fact, a quadratic function $g^{(k)}(x,y) = \frac{1}{2} y^T A_k y$ does not satisfy this assumption when $A_k \neq A_{k'}$ for $k \neq k'$ because $\|(A_k - \frac{1}{K}\sum_{k'=1}^K A_{k'})y\|^2$ is unbounded for $y \in \mathbb{R}^{d_y}$.

Assumption 3.7 The adjacency matrix E of the communication graph is symmetric and doubly stochastic, whose eigenvalues satisfy $|\lambda_n| \leq \cdots \leq |\lambda_2| < |\lambda_1| = 1$.

In this paper, we denote $\lambda \triangleq |\lambda_2|$ so that the spectral gap of the communication graph can be represented as $1 - \lambda$. Additionally, Assumptions 3.2 and 3.3 also hold for the full gradient. Throughout this paper, we use $a_t^{(k)}$ to denote the variable a on the k-th worker in the t-th iteration and use $\bar{a}_t = \frac{1}{K}\sum_{k=1}^K a_t^{(k)}$ to denote the averaged variables. Moreover, for a function $f(\cdot,\cdot)$, we use $\nabla_i f(\cdot,\cdot)$ to denote the gradient with respect to the i-th argument where $i \in \{1,2\}$.

4 Decentralized Stochastic Bilevel Gradient Descent

4.1 Estimation of Stochastic Hypergradient

In this paper, we denote $F^{(k)}(x) \triangleq f^{(k)}(x, y^*(x))$, $F(x) \triangleq \frac{1}{K}\sum_{k=1}^K F^{(k)}(x)$, and $f(x, y^*(x)) \triangleq \frac{1}{K}\sum_{k=1}^K f^{(k)}(x, y^*(x))$. Then, the global hypergradient is defined as follows:

$$\nabla F(x) = \nabla_1 f(x, y^*(x)) - \left[\frac{1}{K}\sum_{k=1}^K \nabla_{12}^2 g^{(k)}(x, y^*(x))\right]\left[\frac{1}{K}\sum_{k=1}^K \nabla_{22}^2 g^{(k)}(x, y^*(x))\right]^{-1} \nabla_2 f(x, y^*(x)).$$

Here, following (Li et al., 2022), the Hessian-inverse-vector product $\left[\frac{1}{K}\sum_{k=1}^K \nabla_{22}^2 g^{(k)}(x, y^*(x))\right]^{-1} \nabla_2 f(x, y^*(x))$ can be viewed as the optimal solution of the following constrained strongly-convex quadratic optimization problem:

$$\min_{z \in \mathbb{R}^{d_y}} h(x,z) \triangleq \frac{1}{K}\sum_{k=1}^K h^{(k)}(x,z) \quad s.t. \quad \|z\| \leq \frac{c_{f_y}}{\mu} \tag{2}$$

where $h^{(k)}(x,z) = \frac{1}{2} z^T \nabla_{22}^2 g^{(k)}(x, y^*(x)) z - z^T \nabla_2 f^{(k)}(x, y^*(x))$ and $z^*(x) = \left[\frac{1}{K}\sum_{k=1}^K \nabla_{22}^2 g^{(k)}(x, y^*(x))\right]^{-1} \frac{1}{K}\sum_{k=1}^K \nabla_2 f^{(k)}(x, y^*(x))$. It is easy to know that $\|z^*(x)\| \leq \frac{c_{f_y}}{\mu}$ based on the aforementioned assumptions. Therefore, we have the

constraint $\|z\| \leq \frac{c_{fy}}{\mu}$ in Eq. (2). Otherwise, the solution of Eq. (2) may not be a good approximation for $z^*(x)$. Moreover, a favorable property of Eq. (2) is that the gradient $\nabla_z h(x, z^*(x)) = 0$ even though there exists a constraint.

In terms of $z^*(x)$, the global hypergradient can be represented as $\nabla F(x) = \nabla_1 f(x, y^*(x)) - \left[\frac{1}{K}\sum_{k=1}^{K} \nabla_{12}^2 g^{(k)}(x, y^*(x))\right] z^*(x)$. With this reformulation, we can use the approximated solution of Eq. (2) to approximate the Hessian-inverse-vector product without computing Hessian inverse explicitly.

On the other hand, the hypergradient of the *k*-th worker is defined as follows:

$$\nabla F^{(k)}(x) = \nabla_1 f^{(k)}(x, y^*(x)) - \nabla_{12}^2 g(x, y^*(x)) \left[\nabla_{22}^2 g(x, y^*(x))\right]^{-1} \nabla_2 f^{(k)}(x, y^*(x)). \tag{3}$$

Obviously, it depends on the global Jacobian matrix $\nabla_{12}^2 g(x, y^*(x))$ and Hessian matrix $\nabla_{22}^2 g(x, y^*(x))$, which are expensive to obtain in each iteration. Moreover, $y^*(x)$ and $z^*(x)$ are also expensive to obtain in each iteration of the stochastic gradient based algorithm. Therefore, we proposed the following *biased* gradient estimators to approximate $\nabla F^{(k)}(x)$ and $\nabla_2 h^{(k)}(x, z)$ on the *k*-th worker for updating *y* and *z*:

$$\begin{aligned} \hat{\mathcal{G}}_F^{(k)}(x, y, z) &\triangleq \nabla_1 f^{(k)}(x, y) - \nabla_{12}^2 g^{(k)}(x, y) z^{(k)}, \\ \hat{\mathcal{G}}_h^{(k)}(x, y, z) &\triangleq \nabla_{22}^2 g^{(k)}(x, y) z^{(k)} - \nabla_2 f^{(k)}(x, y) \end{aligned} \tag{4}$$

where $z^{(k)}$ is the approximated solution of the optimization problem: $\min_{z:\|z\| \leq \frac{c_{fy}}{\mu}} h^{(k)}(x, z)$, and *y* is an approximation of $y^*(x)$. In other words, we can leverage $\hat{\mathcal{G}}_h^{(k)}(x, y, z)$ to update $z^{(k)}$, which will be used to construct the approximated hypergradient $\hat{\mathcal{G}}_F^{(k)}(x, y, z)$. Correspondingly, we can define the stochastic gradient as follows:

$$\begin{aligned} \hat{\mathcal{G}}_F^{(k)}(x, y, z; \hat{\xi}) &\triangleq \nabla_1 f^{(k)}(x, y; \xi) - \nabla_{12}^2 g^{(k)}(x, y; \zeta) z^{(k)} \\ \hat{\mathcal{G}}_h^{(k)}(x, y, z; \hat{\xi}) &\triangleq \nabla_{22}^2 g^{(k)}(x, y; \zeta) z^{(k)} - \nabla_2 f^{(k)}(x, y; \xi) \end{aligned} \tag{5}$$

where $\hat{\xi} \triangleq \{\xi, \zeta\}$.

To present our algorithms, we introduce the following terminologies:

$$\begin{aligned} &X_t = [x_t^{(1)}, x_t^{(2)}, \cdots, x_t^{(K)}]\, Y_t = [y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(K)}]\, Z_t = [z_t^{(1)}, z_t^{(2)}, \cdots, z_t^{(K)}] \\ &\delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_t, Z_t; \hat{\xi}_t) \triangleq [\hat{\mathcal{G}}_F^{(1)}(x_t^{(1)}, y_t^{(1)}, z_t^{(1)}; \hat{\xi}_t^{(1)}), \cdots, \hat{\mathcal{G}}_F^{(K)}(x_t^{(K)}, y_t^{(K)}, z_t^{(K)}; \hat{\xi}_t^{(K)})] \\ &\delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t; \hat{\xi}_t) \triangleq [\hat{\mathcal{G}}_h^{(1)}(x_t^{(1)}, y_t^{(1)}, z_t^{(1)}; \hat{\xi}_t^{(1)}), \cdots, \hat{\mathcal{G}}_h^{(K)}(x_t^{(K)}, y_t^{(K)}, z_t^{(K)}; \hat{\xi}_t^{(K)})] \\ &\delta_t^{g}(X_t, Y_t; \zeta_t) \triangleq [\nabla_y g^{(1)}(x_t^{(1)}, y_t^{(1)}; \zeta_t^{(1)}), \cdots, \nabla_y g^{(K)}(x_t^{(K)}, y_t^{(K)}; \zeta_t^{(K)})]. \end{aligned} \tag{6}$$

Then, we use $\delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_t, Z_t)$, $\delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t)$, and $\delta_t^{g}(X_t, Y_t)$ to denote the corresponding full gradient. Moreover, we use $\bar{A}_t = [\bar{a}_t, \cdots, \bar{a}_t]$ to denote the matrix which is composed of *K* mean variables $\bar{a}_t$, where *a* can be any variable in this paper.

Require: $x_0^{(k)} = x_0$, $y_0^{(k)} = y_0$, $z_0^{(k)} = z_0$, $\eta > 0$, $\alpha_1 > 0$, $\alpha_2 > 0$, $\alpha_3 > 0$, $\beta_1 > 0$, $\beta_2 > 0$, $\beta_3 > 0$, $\alpha_1\eta^2 < 1$, $\alpha_2\eta^2 < 1$, $\alpha_3\eta^2 < 1$. $P_{-1} = 0$, $Q_{-1} = 0$, $R_{-1} = 0$, $U_{-1} = 0$, $V_{-1} = 0$, $W_{-1} = 0$.

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

2: **Option-I & II: Compute variance-reduced gradient V_t for simultaneous/alternating update**

$$V_t = \begin{cases} \delta_t^g(X_t, Y_t; \zeta_t), & t = 0 \\ (1-\alpha_2\eta^2)(V_{t-1} - \delta_t^g(X_{t-1}, Y_{t-1}; \zeta_t)) + \delta_t^g(X_t, Y_t; \zeta_t), & t > 0 \end{cases},$$

Update Y:

$$Q_t = Q_{t-1}E + V_t - V_{t-1}\ ,\ Y_{t+\frac{1}{2}} = Y_tE - \beta_2 Q_t,\ \ Y_{t+1} = Y_t + \eta(Y_{t+\frac{1}{2}} - Y_t),$$

3: **Option-I: Compute variance-reduced gradient W_t for simultaneous update**

$$W_t = \begin{cases} \delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t; \hat{\xi}_t), & t = 0 \\ (1-\alpha_3\eta^2)(W_{t-1} - \delta_t^{\hat{\mathcal{G}}_h}(X_{t-1}, Y_{t-1}, Z_{t-1}; \hat{\xi}_t)) + \delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t; \hat{\xi}_t), & t > 0 \end{cases},$$

Option-II: Compute variance-reduced gradient W_t for alternating update

$$W_t = \begin{cases} \delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t; \hat{\xi}_t), & t = 0 \\ (1-\alpha_3\eta^2)(W_{t-1} - \delta_t^{\hat{\mathcal{G}}_h}(X_{t-1}, Y_t, Z_{t-1}; \hat{\xi}_t)) + \delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t; \hat{\xi}_t), & t > 0 \end{cases},$$

Update Z:

$$R_t = R_{t-1}E + W_t - W_{t-1}\ ,\ Z_{t+\frac{1}{2}} = \mathcal{P}(Z_tE - \beta_3 R_t),\ \ Z_{t+1} = Z_t + \eta(Z_{t+\frac{1}{2}} - Z_t),$$

4: **Option-I: Compute variance-reduced gradient U_t for simultaneous update**

$$U_t = \begin{cases} \delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_t, Z_t; \hat{\xi}_t), & t = 0 \\ (1-\alpha_1\eta^2)(U_{t-1} - \delta_t^{\hat{\mathcal{G}}_F}(X_{t-1}, Y_{t-1}, Z_{t-1}; \hat{\xi}_t)) + \delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_t, Z_t; \hat{\xi}_t), & t > 0 \end{cases},$$

Option-II: Compute variance-reduced gradient U_t for alternating update

$$U_t = \begin{cases} \delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_{t+1}, Z_{t+1}; \hat{\xi}_t), & t = 0 \\ (1-\alpha_1\eta^2)(U_{t-1} - \delta_t^{\hat{\mathcal{G}}_F}(X_{t-1}, Y_t, Z_t; \hat{\xi}_t)) + \delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_{t+1}, Z_{t+1}; \hat{\xi}_t), & t > 0 \end{cases},$$

Update X:

$$P_t = P_{t-1}E + U_t - U_{t-1}\ ,\ X_{t+\frac{1}{2}} = X_tE - \beta_1 P_t,\ \ X_{t+1} = X_t + \eta(X_{t+\frac{1}{2}} - X_t),$$

5: **end for**

Algorithm 1 Decentralized Stochastic Variance-Reduced Bilevel Gradient Descent Algorithm with Simultaneous (corresponds to Option-I) and Alternating (corresponds to Option-II) update.

4.2 Decentralized Stochastic Variance-Reduced Bilevel Gradient Descent Algorithm

In Algorithm 1, we present two novel decentralized stochastic variance-reduced bilevel gradient descent algorithms based on the simultaneous update (**DSVRBGD-S**) and alternating update (**DSVRBGD-A**) strategies, respectively. Generally speaking, for *the computation on each worker*, we use the variance-reduced gradient estimator Cutkosky and Orabona (2019), which is a biased gradient estimator, to solve the lower-level optimization problem and the upper-level optimization problem in Eq. (1), as well as the additional constrained quadratic optimization problem in Eq. (2). For *the communication across workers*, we leverage the gradient tracking communication strategy to communicate three variables and the corresponding gradient estimators between neighboring workers.

DSVRBGD-S Algorithm. DSVRBGD-S employs the **simultaneous** update strategy. Specifically, as shown in Option-I of each step in Algorithm 1, DSVRBGD-S constructs the variance-reduced gradient estimator with the stochastic gradient[2]: $\delta_t^g(X_t, Y_t; \zeta_t)$, $\delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t; \hat{\xi}_t)$, and $\delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_t, Z_t; \hat{\xi}_t)$, which are computed on the variable in the t-th iteration: $\{X_t, Y_t, Z_t\}$. These three stochastic gradients can be computed simultaneously, and the update of three variables can also be completed simultaneously.

[2] Throughout this paper, we ignore the discussion of the stochastic gradient computed on the variable in the $(t-1)$-th iteration for simplicity.

DSVRBGD-A Algorithm. DSVRBGD-A leverages the **alternating** update strategy, which is to update three variables sequentially. Specifically, as shown in Step 2 of Algorithm 1, DSVRBGD-A first computes the variance-reduced gradient estimator V_t based on the variable $\{X_t, Y_t, Z_t\}$, with which the variable Y_t is updated to Y_{t+1}. After that, as shown in Option-II of Step 3 in Algorithm 1, DSVRBGD-A computes the variance-reduced gradient estimator W_t based on the variable $\{X_t, Y_{t+1}, Z_t\}$ as:

$$W_t = (1-\alpha_3\eta^2)(W_{t-1} - \delta_t^{\hat{\mathcal{G}}_h}(X_{t-1}, Y_t, Z_{t-1}; \hat{\xi}_t)) + \delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t; \hat{\xi}_t) \tag{7}$$

where $\alpha_3 > 0$, $\eta > 0$ and $\alpha_3\eta^2 < 1$. It can be observed the new update Y_{t+1} is used for computing $\delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t; \hat{\xi}_t)$, rather than using the prior update Y_t to compute $\delta_t^{\hat{\mathcal{G}}_h}(X_t, Y_t, Z_t; \hat{\xi}_t)$ as Option-I of DSVRBGD-S. Then, DSVRBGD-A employs the following gradient-tracking approach to update and communicate Z_t:

$$R_t = R_{t-1}E + W_t - W_{t-1}, Z_{t+\frac{1}{2}} = \mathcal{P}(Z_tE - \beta_3 R_t)\, Z_{t+1} = Z_t + \eta(Z_{t+\frac{1}{2}} - Z_t) \tag{8}$$

where $\beta_3 > 0$, R_t can be viewed as the estimation of the global $\bar{W}_t$, $R_{t-1}E$ and Z_tE denote the peer-to-peer communication to communicate R and Z in terms of the adjacency matrix E. Since Eq. (2) is a constrained optimization problem, we apply the projection operator $\mathcal{P}(\cdot)$ to the intermediate variable $Z_{t+\frac{1}{2}}$ such that the intermediate variable on all workers always satisfies that constraint. It is worth noting that Z_{t+1} also satisfies that constraint because it is a convex combination between Z_t and $Z_{t+\frac{1}{2}}$. After obtaining Z_{t+1}, DSVRBGD-A constructs the stochastic hypergradient $\delta_t^{\hat{\mathcal{G}}_F}(X_t, Y_{t+1}, Z_{t+1}; \hat{\xi}_t)$ based on $\{X_t, Y_{t+1}, Z_{t+1}\}$ in Option-II of Step 4 in Algorithm 1. Then, the variance-reduced gradient estimator is computed for local update and the gradient tracking communication strategy is used for communication to obtain the new update X_{t+1}.

Both the computation overhead and communication cost for estimating the Hessian-inverse-vector product in our two algorithms are small. In particular, estimating this product on each worker only requires a gradient descent update, whose cost is in the order of $O(d_y)$ in each iteration. As for the communication, our algorithms only need to communicate the auxiliary optimization variable and its gradient estimator, with a cost of $O(d_y)$ in each iteration. Therefore, the computation and communication costs are small compared to those of the double-loop baselines shown in Table 1.

Key Features. Our two algorithms have the following favorable features.

- The communication cost of our two algorithms in each communication round is just $O(d_x + d_y)$ because only $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_x}$, $z \in \mathbb{R}^{d_y}$, and their gradient estimators are communicated. It is smaller than $O(d_y^2 + d_x d_y)$ of Gossip-DSBO (Yang et al., 2022) and $O(d_x d_y)$ of DSBO (Chen et al., 2022a).
- There is only one loop in our two algorithms. On the contrary, the existing methods, including DSBO (Chen et al., 2022a), Gossip-DSBO (Yang et al., 2022), and MA-DSBO (Yang et al., 2022), are double-loop algorithms. As a result, the number of communication rounds of our two algorithms in each iteration is just $O(1)$, which is smaller than $O(\log \frac{1}{\epsilon})$ of those three existing methods.
- We have developed algorithms based on both the simultaneous and alternating update strategies. Notably, this is the first time applying the alternating update strategy to the

variance-reduced gradient for bilevel optimization.

In summary, our two algorithms are communication-efficient due to the low communication cost in each round and the smaller number of communication rounds.

5 Convergence Analysis

In this section, we present the theoretical convergence rate of our algorithms and discuss the unique challenges and novelties in the convergence analysis. The proof sketch can be found in AppendixB, and the detailed proof can be found in Appendix C and Appendix D.

5.1 Convergence Rate

Theorem 5.1 *Under Assumptions 3.1-3.7, by letting η, β_1, β_2, and β_3 satisfy Eq. (C98), and setting $\alpha_1 = O(\frac{1}{K})$, $\alpha_2 = O(\frac{1}{K})$, and $\alpha_3 = O(\frac{1}{K})$, for a sufficiently large $T \geq \frac{K^2}{(1-\lambda)^4}$, DSVRBGD-S has the following convergence rate:*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{x}_t)\|^2] \leq & O\left(\frac{1}{\eta T}\right) + O\left(\frac{1}{\eta T B_0}\right) + O\left(\frac{1}{\beta_1 \eta T}\right) + O\left(\frac{1}{\beta_2 \eta T}\right) + O\left(\frac{1}{\beta_3 \eta T}\right) \\ & + O\left(\frac{1}{\alpha_1 \eta^2 T K B_0}\right) + O\left(\frac{1}{\alpha_2 \eta^2 T K B_0}\right) + O\left(\frac{1}{\alpha_3 \eta^2 T K B_0}\right) + O\left(\frac{\alpha_1 \eta^2}{K}\right) + O\left(\frac{\alpha_2 \eta^2}{K}\right) \\ & + O\left(\frac{\alpha_3 \eta^2}{K}\right) + O\left(\alpha_1^2 \eta^3\right) + O\left(\alpha_2^2 \eta^3\right) + O\left(\alpha_3^2 \eta^3\right),\end{aligned} \tag{9}$$

where B_0 is the batch size in the first iteration.

Corollary 5.2 *Under Assumptions 3.1-3.7, by setting $\alpha_1 = O(1/K)$, $\alpha_2 = O(1/K)$, $\alpha_3 = O(1/K)$, $\beta_1 = O((1-\lambda)^4)$, $\beta_2 = O((1-\lambda)^2)$, $\beta_3 = O((1-\lambda)^4)$, $\eta = O(K\epsilon^{1/2})$, the batch size in the first iteration as $B_0 = O(1/\epsilon^{1/2})$, the batch size in other iterations as O(1), and $T = O\left(\frac{1}{K(1-\lambda)^4\epsilon^{3/2}}\right)$, DSVRBGD-S can achieve the ϵ-accuracy solution: $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{x}_t)\|^2] \leq \epsilon$.*

Remark 5.3 1) Our convergence rate does not depend on any assumptions regarding the heterogeneity. On the contrary, existing methods (Chen et al., 2022a; Yang et al., 2022; Chen et al., 2022b) require strong assumptions, which are shown in Table 1. 2) Our algorithm is more communication-efficient than them (Chen et al., 2022a; Yang et al., 2022; Chen et al., 2022b) because DSVRBGD-S has a smaller number of communication rounds and the low cost in each round. 3) DSVRBGD-S has a worse dependence on the spectral gap than the homogeneous method VRDBO (Gao et al., 2023), i.e., $1/(1-\lambda)^4$ versus $1/(1-\lambda)^2$. This is caused by the projection operation for the estimator of the Hessian-inverse-vector product as discussed in Section 5.2.

Theorem 5.4 *Under Assumptions 3.1-3.7, by letting η, β_1, β_2, and β_3 satisfy Eq. (D205), and setting $\alpha_1 = O(\frac{1}{K})$, $\alpha_2 = O(\frac{1}{K})$, and $\alpha_3 = O(\frac{1}{K})$, for a sufficiently large $T \geq \frac{K^2}{(1-\lambda)^4}$, DSVRBGD-S has the following convergence rate:*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{x}_t)\|^2] \leq\ & O\left(\frac{1}{\eta T}\right)+O\left(\frac{1}{\eta T B_0}\right)+O\left(\frac{1}{\beta_1\eta T}\right)+O\left(\frac{1}{\beta_2\eta T}\right)+O\left(\frac{1}{\beta_3\eta T}\right)\\ &+O\left(\frac{1}{\alpha_1\eta^2 TKB_0}\right)+O\left(\frac{1}{\alpha_2\eta^2 TKB_0}\right)+O\left(\frac{\alpha_1\eta^2}{K}\right)+O\left(\frac{\alpha_2\eta^2}{K}\right)+O\left(\frac{\alpha_3\eta^2}{K}\right)\\ &+O\left(\alpha_3^2\eta^3\right)+O\left(\alpha_1^2\eta^3\right)+O\left(\alpha_2^2\eta^3\right)+O\left(\alpha_2^2\eta^4\right)+O\left(\frac{\eta\beta_2^2}{T}\right)+O\left(\frac{\eta\beta_3^2}{T}\right)+O\left(\frac{\eta^3\beta_2^2\beta_3^2}{T}\right)\\ &+O\left(\frac{\beta_2^2}{\alpha_1 T}\right)+O\left(\frac{\beta_3^2}{\alpha_1 T}\right)+O\left(\frac{\eta^2\beta_2^2\beta_3^2}{\alpha_1 T}\right)+O\left(\frac{\eta\beta_2^2}{TB_0}\right)+O\left(\frac{\eta^3\beta_2^2\beta_3^2}{TB_0}\right)+O\left(\frac{\beta_2^2}{\alpha_1 TB_0}\right)\\ &+O\left(\frac{1}{\alpha_3\eta^2 TKB_0}\right)+O\left(\frac{\eta^2\beta_2^2\beta_3^2}{\alpha_1 TB_0}\right)\end{aligned} \tag{10}$$

where B_0 is the batch size in the first iteration.

Corollary 5.5 *Under Assumptions 3.1-3.7, by setting $\alpha_1 = O(1/K)$, $\alpha_2 = O(1/K)$, $\alpha_3 = O(1/K)$, $\beta_1 = O((1-\lambda)^4)$, $\beta_2 = O((1-\lambda)^2)$, $\beta_3 = O((1-\lambda)^4)$, $\eta = O(K\epsilon^{1/2})$, the batch size in the first iteration as $B_0 = O(1/\epsilon^{1/2})$, the batch size in other iterations as O(1), and $T = O\left(\frac{1}{K(1-\lambda)^4\epsilon^{3/2}}\right)$, DSVRBGD-A can achieve the ϵ-accuracy solution: $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\bar{x}_t)\|^2] \leq \epsilon$.*

Remark 5.6 According to Corollary 5.2 and Corollary 5.5, the convergence rate of DSVRBGD-A is in the same order as that of DSVRBGD-S. From Theorem 5.1 and Theorem 5.4, it can be observed that DSVRBGD-A has some additional terms. These additional terms are only related to the higher order of ϵ, which can be found in Eq. (D222).

Remark 5.7 The communication complexity of our two algorithms can improved from two perspectives. On the one hand, we can improve the dependence on ϵ by using the SPIDER gradient estimator Fang et al. (2018). In particular, by using SPIDER, we can use a large batch size in iteration, which leads to a smaller number of iterations. As a result, the communication complexity can be reduced. In fact, in the single-machine setting, Corollary 2 in Yang et al. (2021) has demonstrated the iteration complexity is $O(\epsilon^{-1})$ when using the SPIDER gradient estimator. On the other hand, we can improve the dependence on the spectral gap $1-\lambda$ by using other communication strategies. For example, we can use the multi-round communication, i.e., performing multiple rounds of communication in each iteration, as in Xin et al. (2021) to improve the dependence on the spectral gap.

5.2 Key Challenges and Novelties

The projection operation on the communicated variable z introduces new challenges for convergence analysis. In Algorithm 1, to solve the constrained quadratic optimization problem in Eq. (2), we employ the projection operation to ensure the new update $z_t^{(k)}$ to satisfy the constraint $\|z_t^{(k)}\| \leq \frac{c_{f_y}}{\mu}$, which incurs unique challenges for decentralized optimization. Specifically, since the projection operator is NOT a linear operator, it leads to $\sum_{k=1}^K \mathcal{P}(z_t^{(k)}) \neq \mathcal{P}(\sum_{k=1}^K z_t^{(k)})$, which makes it challenging to bound the optimization error $\mathbb{E}[\|\bar{z}_t - z^*(\bar{x}_t)\|^2]$.

First, due to the non-linear projection projection, we cannot bound $\mathbb{E}[\|\bar{z}_t - z^*(\bar{x}_t)\|^2]$ as $\mathbb{E}[\|\bar{y}_t - y^*(\bar{x}_t)\|^2]$. In particular, when there is no projection, the averaged variable $\bar{y}_t$ behaves like the variable under the single-machine setting. Therefore, it is easy to bound $\mathbb{E}[\|\bar{y}_t - y^*(\bar{x}_t)\|^2]$ by following Lemma 28 in Huang et al. (2008) under the single-machine setting. However, due to the non-linear projection operation, Eq. (83) in Huang et al. (2008) only holds for the variable on each worker $z_t^{(k)}$, rather than the averaged variable $\bar{z}_t$. Therefore, the approach (Huang et al., 2008) designed for the single-machine setting cannot be applied to our algorithms.

Second, although there exists an alternative approach, i.e., Lemma 2 in Zhang et al. (2024), to bound $\mathbb{E}[\|\bar{y}_t - y^*(\bar{x}_t)\|^2]$ under the decentralized setting, it cannot be directly applied to $\mathbb{E}[\|\bar{z}_t - z^*(\bar{x}_t)\|^2]$ due to the projection operation. In particular, just as shown in Eq. (C27), we cannot obtain $\mathbb{E}[\|\bar{z}_{t+1} - z^*(\bar{x}_t)\|^2] = \mathbb{E}[\|\bar{z}_t - \eta\beta_3\bar{r}_t - z^*(\bar{x}_t)\|^2]$ as Eq. (23) in Zhang et al. (2024) due to $\sum_{k=1}^K \mathcal{P}(z_t^{(k)}) \neq \mathcal{P}(\sum_{k=1}^K z_t^{(k)})$. Therefore, we establish a new upper bound for $\mathbb{E}[\|\bar{z}_{t+1} - z^*(\bar{x}_t)\|^2]$, which has two additional terms as shown in Eq. (C27). Especially, our theoretical analysis shows that the additional term $\mathbb{E}[\|Z_t - \bar{Z}_t\|_F^2]$ in Eq. (C27) makes the convergence rate have a worse dependence on the spectral gap. Specifically, it makes the coefficient c_4 of $\mathbb{E}[\|Z_t - \bar{Z}_t\|_F^2]$ in the potential function $\mathcal{L}_t$ in Eq. (B10) have an additional term compared with c_3 of $\mathbb{E}[\|Y_t - \bar{Y}_t\|_F^2]$, i.e., $c_4 = \frac{2\beta_1}{(1-\lambda^2)}\tilde{c}_4 + \frac{5\beta_1}{\beta_3\eta(1-\lambda^2)}\tilde{c}_1$ (See Eq. (C79)), where the second term has a factor $\frac{1}{\beta_3}$. Such a factor makes β_3 have to be as small as $O((1-\lambda)^4)$. Specifically, as shown in Eq. (C87), that factor leads to the second inequality, which further leads to Eq. (C89), resulting in a worse dependence $O((1-\lambda)^4)$. Obviously, if there were no such a factor in c_4, the upper bound of β_3 could be improved to $O((1-\lambda)^2)$, which will lead to a better convergence rate.

In summary, the additional communication required for estimating hypergradient makes the convergence analysis more challenging due to the involved projection operation. Our theoretical analysis clearly discloses how it affects the convergence rate.

The alternating update introduces more challenges for convergence analysis. Specifically, considering the t-th iteration, the alternating update introduces new challenges when bounding the gradient estimation error: $\mathbb{E}[\|\delta^{\hat{\mathcal{G}}_F}(X_t, Y_{t+1}, Z_{t+1}) - U_t\|_F^2]$ and $\mathbb{E}[\|\delta^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t) - W_t\|_F^2]$. Taking the former one as an example, as shown in Option-II of Step 4 in Algorithm 1, U_t should use the new update Y_{t+1} and Z_{t+1}. As a result, its upper bound depends on $\mathbb{E}[\|Y_{t+1} - Y_t\|_F^2]$ and $\mathbb{E}[\|Z_{t+1} - Z_t\|_F^2]$ as shown in Lemma D.5. Furthermore, as shown in Lemma D.14 and Lemma D.15, those two terms depend on the gradient estimation error $\mathbb{E}[\|V_{t-1} - \delta^g(X_{t-1}, Y_{t-1})\|_F^2]$ and $\mathbb{E}[\|W_{t-1} - \delta^{\hat{\mathcal{G}}_h}(X_{t-1}, Y_t, Z_{t-1})\|_F^2]$, respectively. As a result, the upper bound of $\mathbb{E}[\|\delta^{\hat{\mathcal{G}}_F}(X_t, Y_{t+1}, Z_{t+1}) - U_{t+1}\|_F^2]$ depends on all three gradient estimation errors in the $(t-1)$-th iteration, i.e., $\mathbb{E}[\|\delta^{\hat{\mathcal{G}}_F}(X_{t-1}, Y_t, Z_t) - U_{t-1}\|_F^2]$, $\mathbb{E}[\|V_{t-1} - \delta^g(X_{t-1}, Y_{t-1})\|_F^2]$, and $\mathbb{E}[\|W_{t-1} - \delta^{\hat{\mathcal{G}}_h}(X_{t-1}, Y_t, Z_{t-1})\|_F^2]$, which can be found in Eq. (B15). On the contrary, for the simultaneous update, as shown in Lemma C.5, it only depends on its own estimation error in the t-th iteration. Therefore, it is much more challenging to establish the convergence rate for DSVRBGD-A compared with DSVRBGD-S. To address this challenge, we design a new potential function to handle the challenging dependence between those gradient estimation errors. In particular, as shown in Eq. (D187), we introduce additional terms for the coefficient c_{11} and c_{13} of $\mathbb{E}[\|V_t - \delta^g(X_t, Y_t)\|_F^2]$ and $\mathbb{E}[\|W_t - \delta^{\hat{\mathcal{G}}_h}(X_t, Y_{t+1}, Z_t)\|_F^2]$ compared with those in Eq. (C99) of the simultaneous update.

6 Experiment

In our experiments, we focus on the hyperparameter optimization problem and the hyper-representation learning problem.

6.1 Hyperparameter Optimization

In our experiments, we focus on the following hyperparameter optimization problem:

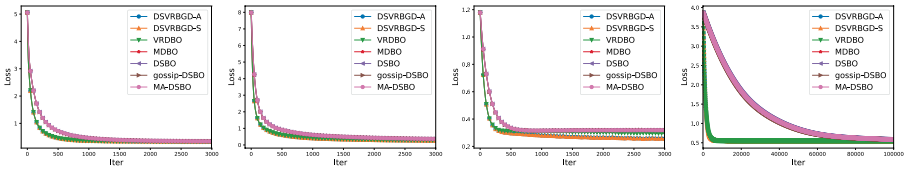
$$\begin{aligned} &\min_{x\in\mathbb{R}^d} \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n^{(k)}}\sum_{i=1}^{n^{(k)}}\ell(y^*(x)^T a_{v,i}^{(k)}, b_{v,i}^{(k)}) \\ &s.t.\ y^*(x) = \arg\min_{y\in\mathbb{R}^{d\times c}} \frac{1}{K}\sum_{k=1}^{K}\frac{1}{m^{(k)}}\sum_{i=1}^{m^{(k)}}\ell(y^T a_{t,i}^{(k)}, b_{t,i}^{(k)}) + \frac{1}{cd}\sum_{p=1}^{c}\sum_{q=1}^{d}\exp(x_q)y_{pq}^2 \end{aligned} \tag{11}$$

where the lower-level optimization problem optimizes the classifier's parameter $y \in \mathbb{R}^{d\times c}$ based on the training set $\{(a_{t,i}^{(k)}, b_{t,i}^{(k)})\}_{i=1}^{m^{(k)}}$, the upper-level optimization problem optimizes the hyperparameter $x \in \mathbb{R}^d$ based on the validation set $\{(a_{v,i}^{(k)}, b_{v,i}^{(k)})\}_{i=1}^{n^{(k)}}$, the loss function is the cross-entropy loss function.

To verify the performance of our algorithm, we use four real-world LIBSVM datasets[3]: w8a, a9a, ijcnn1, covtype. For each dataset, we randomly select 10% of samples as the test set, 70% of the remaining samples as the training set, and the others as the validation set. Then, to demonstrate the performance of our algorithms under the heterogeneous setting, we construct a heterogeneous variant for each training set. In detail, we use eight workers in our experiments. We set the imbalance ratio on these workers, i.e., the ratio between the number of samples in the positive class and the total number of samples, as $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$ by randomly dropping samples from the positive or negative class.

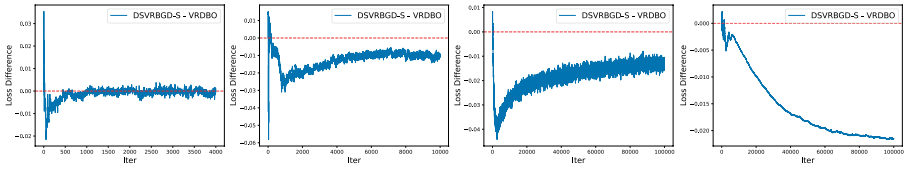
We compare our algorithm with five baseline methods: including MDBO (Gao et al., 2023), DSBO (Chen et al., 2022a), Gossip-DSBO (Yang et al., 2022), MA-DSBO (Chen et al., 2022b), and VRDBO (Gao et al., 2023). As for the hyperparameter, we set the solution accuracy ϵ to 0.05 so that the learning rates of the first four methods are set to $\eta = \epsilon^2$ according to their theoretical results, e.g., Theorem 3.3 in (Chen et al., 2022b), and the learning rate of VRDBO and our algorithm is set to $\eta = \epsilon$ in terms of the corresponding theoretical results. Moreover, α_i is set such that $\alpha_i\eta^2 = 0.9$ and β_i is set to 1 for both VRDBO and our algorithms. As for the double-loop basline methods: DSBO,Gossip-DSBO, and MA-DSBO, the number of iterations for the lower-level update and the Hessian-inverse-vector product update is set to 10. For the single-loop baseline method MDBO, we use a learning rate for the lower-level variable that is ten times larger than those double-loop methods to mitigate the effect of the double loop. Additionally, the batch size of all algorithms is set to 100, and the number of iterations is set to 2, 000. As for the communication topology, we consider three classes: ring graph, random graph, and torus graph. In particular, the probability for generating the random graph is set to 0.4 in our experiments.

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

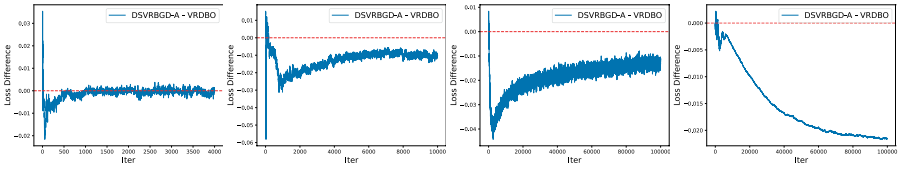


(a) a9a (b) w8a (c) ijcnn1 (d) covtype

Fig. 1 The upper-level loss function value versus the number of iterations on a ring graph



(a) a9a (b) w8a (c) ijcnn1 (d) covtype

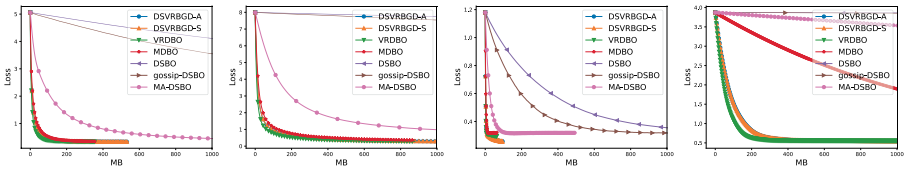


(e) a9a (f) w8a (g) ijcnn1 (h) covtype

Fig. 2 The first row shows the loss function value difference: $\Delta = \mathrm{LOSS}_{\mathrm{DSVRBGD\text{-}S}} - \mathrm{LOSS}_{\mathrm{VRDBO}}$. The second row shows the loss function value difference: $\Delta = \mathrm{LOSS}_{\mathrm{DSVRBGD\text{-}A}} - \mathrm{LOSS}_{\mathrm{VRDBO}}$. The negative difference value denotes our methods converge faster than VRDBO, as they reduce the loss function value more quickly

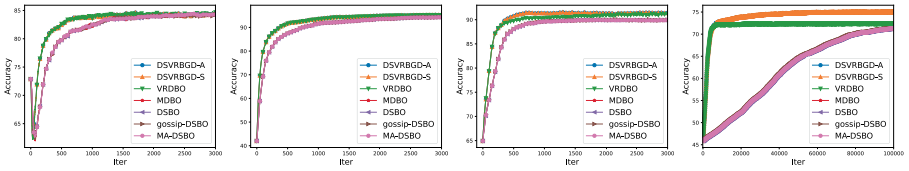
In Figure 1, we plot the upper-level loss function value versus the number of iterations. It can be observed that our two methods converge faster than the method that does not use variance-reduced gradient, including Gossip-DSBO, DSBO, MA-DSBO, and MDBO, and also enjoys a faster convergence speed than the method using variance reduction, i.e., VRDBO. In addition, to better show the improvement of our two methods over VRDBO, we plot the difference between the loss function value obtained from our methods and that obtained from VRDBO in Figure 2. In particular, we plot $\Delta = \mathrm{LOSS}_{\mathrm{DSVRBGD\text{-}S}} - \mathrm{LOSS}_{\mathrm{VRDBO}}$ in the first row and $\Delta = \mathrm{LOSS}_{\mathrm{DSVRBGD\text{-}A}} - \mathrm{LOSS}_{\mathrm{VRDBO}}$ in the second row. When $\Delta < 0$, DSVRBGD-S(DSVRBGD-A) decreases the loss function value faster than VRDBO. Otherwise, VRDBO converges faster. In Figure 2, it can be seen that the value of the loss function decreases more rapidly with our two methods than with VRDBO, confirming that our two methods outperform VRDBO.

In Figure 3, we plot the upper-level loss function value versus the number of communicated megabytes (MB), when using the ring graph. Note that we only show the first 100MB to make the comparison clearer. Compared with the homogeneous baseline methods, we can find that VRDBO and MDBO have a smaller communication cost. The reason is that they are designed for homogeneous setting so that they do not need to communicate any information regarding Hessian or Jacobian. On the contrary, our methods, DSVRBGD-S and



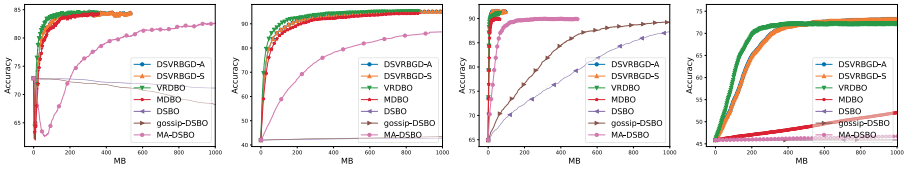
(a) a9a (b) w8a (c) ijcnn1 (d) covtype

Fig. 3 The upper-level loss function value versus the communication cost (MB) on a ring graph



(a) a9a (b) w8a (c) ijcnn1 (d) covtype

Fig. 4 The test accuracy versus the number of iterations on a ring graph



(a) a9a (b) w8a (c) ijcnn1 (d) covtype

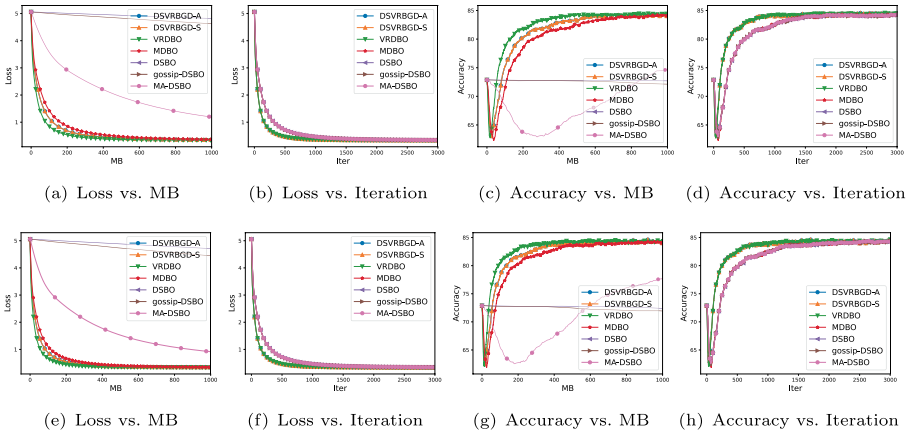
Fig. 5 The test accuracy versus the communication cost (MB) on a ring graph

DSVRBGD-A, should communicate an additional Hessian-inverse-vector product z compared to these two baseline methods. Therefore, our communication cost is a little larger than them. Compared with the heterogeneous baseline methods, we can find that our two algorithms are much more communication-efficient than all of them, because these baseline methods should communicate the high-dimensional Hessian or Jacobian matrices. In contrast, our methods only need to communicate the low-dimensional Hessian-inverse-vector product z, saving communication costs significantly.

Moreover, in Figure 4 and Figure 5, we plot the test accuracy versus the number of iterations and the communication cost, respectively. We can still find that our methods enjoy small communication costs and converge faster than all baselines that are designed for the heterogeneous setting.

Topology. In Figure 6, we plot the loss function value and test accuracy with respect to the communication cost and the number of iterations on the random communication graph and the torus communication graph. Under these two settings, we can still find that our methods outperform all baseline methods designed for the heterogeneous setting.

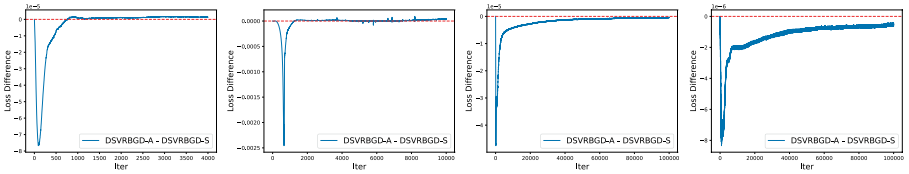
Simultaneous Update vs Alternating Update. In Figure 7, we show the difference between the upper-level loss function value obtained from the simultaneous update (DSVRBGD-S) and the alternating update (DSVRBGD-A). In particular, we plot



(a) Loss vs. MB (b) Loss vs. Iteration (c) Accuracy vs. MB (d) Accuracy vs. Iteration

(e) Loss vs. MB (f) Loss vs. Iteration (g) Accuracy vs. MB (h) Accuracy vs. Iteration

Fig. 6 The first row uses a random communication graph. The second row uses a torus communication graph. The datasets is a9a

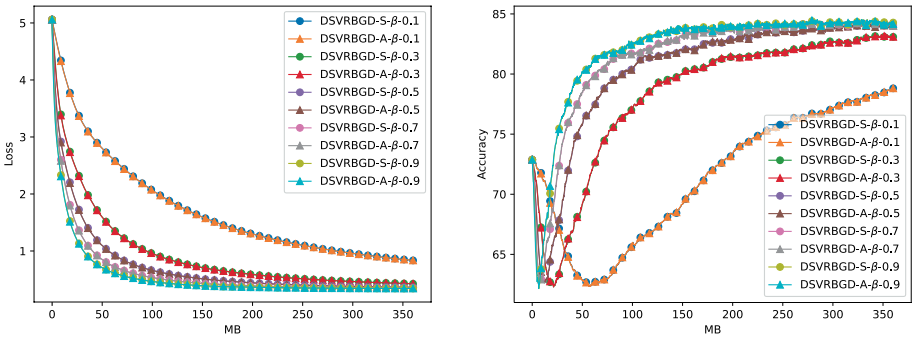


(a) a9a (b) w8a (c) ijcnn1 (d) covtype

Fig. 7 The loss function value difference: $\Delta = \text{LOSS}_{\text{DSVRBGD-A}} - \text{LOSS}_{\text{DSVRBGD-S}}$. The negative difference value denotes DSVRBGD-A converges faster than DSVRBGD-S, as it reduces the loss function value more quickly

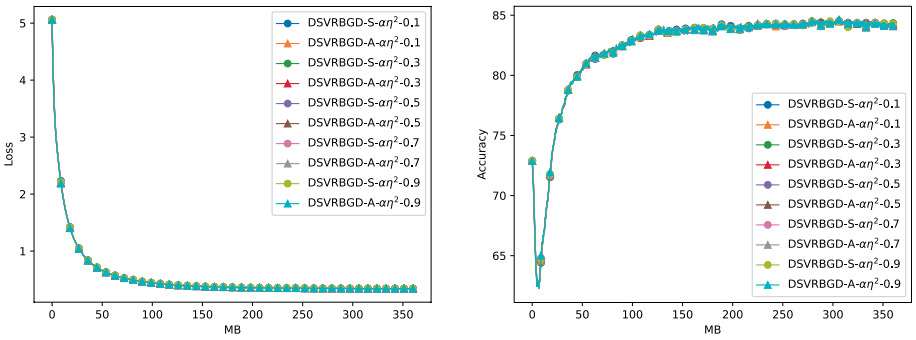
$\Delta = \text{Loss}_{\text{DSVRBGD-A}} - \text{Loss}_{\text{DSVRBGD-S}}$. When $\Delta < 0$, DSVRBGD-A decreases the loss function value faster than DSVRBGD-S. Otherwise, DSVRBGD-S converges faster. From Figure 7, we can find that the alternating update (DSVRBGD-A) converges a faster than the simultaneous update (DSVRBGD-S) because $\Delta < 0$.

Ablation Studies for Hyperparameters. We conduct experiments to show the performance of our methods with different hyperparameters, where the dataset is a9a. In Figure 8, we show the effect of β_i (where $i \in \{1, 2, 3\}$) on the convergence performance of our DSVRBGD-S and DSVRBGD-A. Specifically, we set $\beta_i = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ while fixing $\eta = 0.05$. Since the actual learning rate is $\beta_i\eta$ from the global view as $\bar{X}_{t+1} = \bar{X}_t - \beta_1\eta\bar{P}_t$, a smaller β_i results in a smaller learning rate. Therefore, a smaller β_i leads to a slower convergence rate as shown in Figure 8. In Figure 9, we show the effect of α_i (where $i \in \{1, 2, 3\}$) on the convergence performance of our DSVRBGD-S and DSVRBGD-A. In this experiment, we fix $\eta = 0.05$ and then vary α_i such that $\alpha_i\eta^2 = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. It can be observed that it does not affect the convergence rate significantly.



(a) Upper-level loss versus MB (b) Test Accuracy versus MB

Fig. 8 The convergence performance of our DSVRBGD-S and DSVRBGD-A for different values of β_i. The dataset is a9a



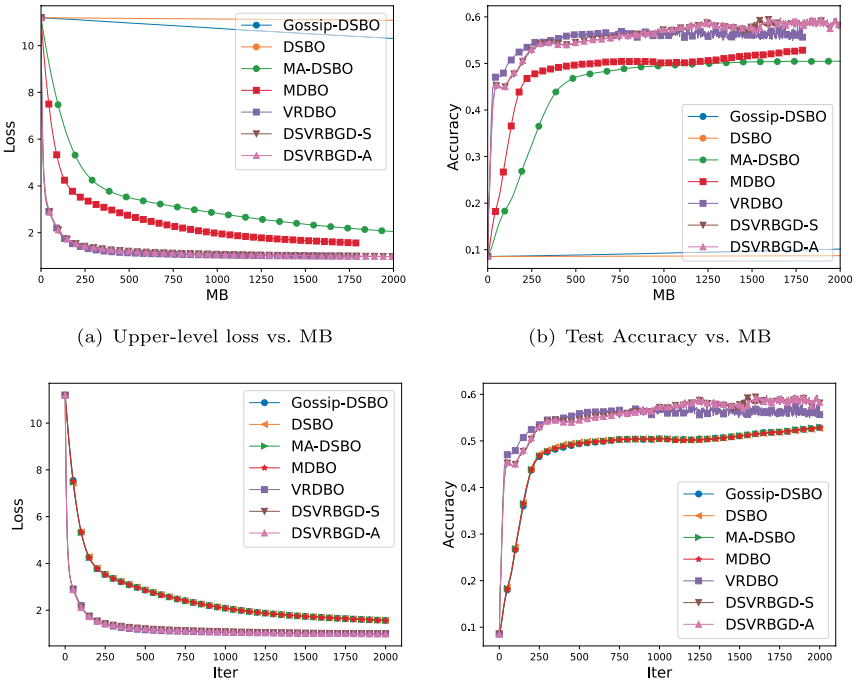
(a) Upper-level loss versus MB (b) Test Accuracy versus MB

Fig. 9 The convergence performance of our DSVRBGD-S and DSVRBGD-A for different values of α_i. The dataset is a9a

6.2 Hyper-representation Learning

In this additional experiment, by following FedNEST Tarzanagh et al. (2022), we focus on the hyper-representation optimization problem. Specifically, the upper-level optimization problem is to optimize the backbone neural network's weight to obtain better features, while the lower-level optimization problem is to optimize the classifier's weight given the backbone neural network. In our experiment, we use a two-layer fully-connected neural network, whose hidden layer has 30 neurons. In this way, the upper-level optimization problem learns the weight of the first layer, which is nonconvex optimization problem. The lower-level optimization problem learns the weight of the last layer, which is actually a logistic regression problem when fixing the first layer. It is a strongly convex problem when adding a regularization term. In this experiment, we consider the multi-class classification dataset from LIBSVM[4]: covtype, which has 7 classes, 581,012 samples, and 54 features. In this experiment, we still randomly select 10% of samples as the test set, 70% of the remaining

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html



(a) Upper-level loss vs. MB

(b) Test Accuracy vs. MB

(c) Upper-level loss vs. iteration

(d) Test Accuracy vs. iteration

Fig. 10 The comparison of different methods for hyper-representation learning using the (multi-class) covtype dataset and a ring graph

samples as the training set, and the others as the validation set. In addition, the solution accuracy ϵ is set to 0.02. The other settings are set in the same way as our first experiment.

In Figure 10, we plot the upper-level loss function value and test accuracy versus the communication cost and the number of iterations, respectively. It can be observed that our two methods converge much faster than all state-of-the-art heterogeneous methods in terms of both communication costs and iterations, which confirms the effectiveness of our proposed methods. Note that the homogeneous methods, VRDBO and MDBO, have a smaller communication cost because they do not communicate Hessian or Jacobian matrices.

7 Future Work

In our algorithms, we use the Gradient Tracking (GT) communication strategy. In fact, there are communication strategies other than GT. For example, SPARKLE Zhu et al. (2024) studies the EXTRA and ED communication strategies for bilevel optimization, showing that they enjoy better transient iteration complexity than GT. Specifically, as shown in Table 1 of that paper, when the spectral gap satisfies $1-\rho < \frac{1}{n^3}$, i.e., when the communication graph is very sparse, EXTRA and ED perform better than GT in terms of transient complex-

ity. However, applying these two communication strategies to our algorithms introduces significant challenges, and it is unclear whether they would lead to similar conclusions, as discussed below.

SPARKLE Zhu et al. (2024) algorithm uses stochastic gradient for updating y and z and uses momentum to update x. As such, the dominating term $\frac{1}{\sqrt{nT}}$ (or $\frac{1}{n\epsilon^2}$ for achieving ϵ -accuracy solution) in its convergence upper bound does not rely on the spectral gap $1-\rho$. This leads to the concept of transient iteration complexity, which is defined as: *those iterations before an algorithm reaches its linear-speedup stage*Chen et al. (2021). However, our algorithms use the variance-reduced gradient estimator, STORM, to update all variables. As a result, the dominating term in the convergence upper bound depends on the spectral gap, i.e., $O\left(\frac{1}{K(1-\lambda)^4\epsilon^{3/2}}\right)$. Therefore, according to Chen et al. (2021), the transient iteration complexity is not well defined when there exists a spectral gap in the dominating term. Consequently, the challenge for studying the influence of different communication topology is to establish a convergence rate whose dominating term is independent of the spectral gap. In fact, this is still an open problem even for the traditional single-level optimization algorithm. For example, as shown in Table 1 of Xin et al. (2021), which uses the STORM gradient estimator for decentralized minimization problems, the iteration complexity relies on the spectral gap under the worst case. In summary, the advantage of EXTRA and ED over GT is reflected by the transient iteration complexity. However, this notion is not well defined for algorithms that rely on variance-reduced gradient estimators, since the dominating term in their convergence upper bounds depends on the spectral gap.

Since comparing different communication strategies for variance-reduced gradient-based methods involves an open problem and is nontrivial to address, we leave this for future work.

8 Conclusion

In this paper, we analyzed the convergence rate of decentralized stochastic bilevel optimization algorithms under the heterogeneous setting. In particular, to reduce the communication rounds in each iteration, we proposed to employ the variance-reduced gradient descent on each worker to estimate the Hessian-inverse-vector product. As a result, our algorithm can achieve a small number of iterations. Meanwhile, it can reduce the communication rounds in each iteration and also reduce the cost in each round. In addition, we develop a new algorithm based on the alternating update strategy, which also enjoy these nice empirical and theoretical properties. The extensive experimental results confirm the effectiveness of our two algorithms.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-026-07040-y.

Author Contributions Y. Z. and H.G. contributed to the algorithm design, theoretical analysis, and empirical evaluation. Y. Z., M. T., J.W., and H.G. wrote, reviewed, and revised the manuscript.

Funding This work was partially supported by U.S. NSF CAREER 2339545, NSF IIS 2416607, NSF CNS 2107014.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication All authors have reviewed this paper and agreed to publish this paper.

Materials Availability Not applicable.

Code Availability The code is based on the standard Pytorch and MPI.

References

Chen, X., Huang, M., & Ma, S. (2022). Decentralized bilevel optimization. arXiv preprint arXiv:2206.05670

Chen, X., Huang, M., Ma, S., & Balasubramanian, K. (2022). Decentralized stochastic bilevel optimization with improved per-iteration complexity. arXiv preprint arXiv:2210.12839

Chen, T., Sun, Y., & Yin, W. (2021). Tighter analysis of alternating stochastic gradient method for stochastic nested problems. arXiv preprint arXiv:2106.13781

Chen, Y., Yuan, K., Zhang, Y., Pan, P., Xu, Y., & Yin, W. (2021). Accelerating gossip sgd with periodic global averaging. In: International Conference on Machine Learning, 1791–1802. PMLR

Cutkosky, A., & Orabona, F. (2019). Momentum-based variance reduction in non-convex sgd. Advances in neural information processing systems **32**

Dagréou, M., Ablin, P., Vaiter, S., & Moreau, T. (2022). A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. arXiv preprint arXiv:2201.13409

Dagréou, M., Moreau, T., Vaiter, S., & Ablin, P. (2023). A lower bound and a near-optimal algorithm for bilevel empirical risk minimization. arXiv e-prints, 2302.

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. Advances in neural information processing systems **27**

Dong, Y., Ma, S., Yang, J., & Yin, C. (2023). A single-loop algorithm for decentralized bilevel optimization. CoRR **abs/2311.08945** arXiv:2311.08945

Fang, C., Li, C.J., Lin, Z., & Zhang, T. (2018). Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. Advances in Neural Information Processing Systems **31**.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In: Automated Machine Learning, 3–33. Springer International Publishing Cham

Franceschi, L., Donini, M., Frasconi, P., & Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In: International Conference on Machine Learning, 1165–1173. PMLR

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., & Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In: International Conference on Machine Learning, 1568–1577. PMLR

Gao, H. (2022). On the convergence of momentum-based algorithms for federated stochastic bilevel optimization problems. arXiv preprint arXiv:2204.13299

Gao, H., Gu, B., & Thai, M.T. (2023). On the convergence of distributed stochastic bilevel optimization algorithms over a network. In: International Conference on Artificial Intelligence and Statistics, 9238–9281. PMLR

Ghadimi, S., & Wang, M. (2018). Approximation methods for bilevel programming. arXiv preprint arXiv:1802.02246

Hong, M., Wai, H.-T., Wang, Z., & Yang, Z. (2020). A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. arXiv preprint arXiv:2007.05170

Huang, F. (2022). Fast adaptive federated bilevel optimization. arXiv preprint arXiv:2211.01122

Huang, F., Gao, S., Pei, J., & Huang, H. (2020). Accelerated zeroth-order momentum methods from mini to minimax optimization. arXiv e-prints, 2008

Ji, K., Yang, J., & Liang, Y. (2021). Bilevel optimization: Convergence analysis and enhanced design. In: International Conference on Machine Learning, 4882–4892. PMLR

Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., & Yang, Z. (2021). A near-optimal algorithm for stochastic bilevel optimization via double-momentum. Advances in Neural Information Processing Systems **34**

Li, J., Huang, F., & Huang, H. (2022). Local stochastic bilevel optimization with momentum-based variance reduction. arXiv preprint arXiv:2205.01608

Li, J., Gu, B., & Huang, H. (2022). A fully single loop algorithm for bilevel optimization without hessian inverse. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*, 7426–7434.

Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055

Liu, Z., Zhang, X., Khanduri, P., Lu, S., & Liu, J. (2022). Interact: achieving low sample and communication complexities in decentralized bilevel learning over networks. In: Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 61–70.

Lu, S., Cui, X., Squillante, M.S., Kingsbury, B., & Horesh, L. (2022). Decentralized bilevel optimization for personalized client learning. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5543–5547. IEEE

Niu, Y., Xu, J., Sun, Y., Huang, Y., & Chai, L. (2025). Distributed stochastic bilevel optimization: Improved complexity and heterogeneity analysis. *Journal of Machine Learning Research, 26*(76), 1–58.

Rajeswaran, A., Finn, C., Kakade, S.M., & Levine, S. (2019). Meta-learning with implicit gradients. Advances in neural information processing systems **32**

Spiridonoff, A., Olshevsky, A., & Paschalidis, Y. (2021). Communication-efficient sgd: From local sgd to one-shot averaging. *Advances in Neural Information Processing Systems, 34*, 24313–24326.

Tarzanagh, D.A., Li, M., Thrampoulidis, C., & Oymak, S. (2022). Fednest: Federated bilevel, minimax, and compositional optimization. In: International Conference on Machine Learning, 21146–21179. PMLR

Xin, R., Das, S., Khan, U.A., & Kar, S. (2021). A stochastic proximal gradient framework for decentralized non-convex composite optimization: Topology-independent sample complexity and communication efficiency. arXiv preprint arXiv:2110.01594

Xin, R., Khan, U., & Kar, S. (2021). A hybrid variance-reduced method for decentralized stochastic non-convex optimization. In: International Conference on Machine Learning, 11459–11469. PMLR

Yang, J., Ji, K., & Liang, Y. (2021). Provably faster algorithms for bilevel optimization. Advances in Neural Information Processing Systems **34**

Yang, S., Zhang, X., & Wang, M. (2022). Decentralized gossip-based stochastic bilevel optimization over communication networks. arXiv preprint arXiv:2206.10870

Zhang, X., Mancino-Ball, G., Aybat, N. S., & Xu, Y. (2024). Jointly improving the sample and communication complexities in decentralized stochastic minimax optimization. *Proceedings of the AAAI Conference on Artificial Intelligence, 38*, 20865–20873.

Zhu, S., Kong, B., Lu, S., Huang, X., & Yuan, K. (2024). Sparkle: a unified single-loop primal-dual framework for decentralized bilevel optimization. *Advances in Neural Information Processing Systems, 37*, 62912–62987.

Authors and Affiliations

Yihan Zhang[1] · My T. Thai[2] · Jie Wu[1] · Hongchang Gao[1]

✉ Hongchang Gao
hongchang.gao@temple.edu

Yihan Zhang
yihan.zhang0002@temple.edu

My T. Thai
mythai@cise.ufl.edu

Jie Wu
jiewu@temple.edu

[1] Department of Computer and Information Sciences, Temple University, 1925 N. 12th Street, Philadelphia 19122, PA, USA

[2] Department of Computer and Information Science and Engineering, University of Florida, 1889 Museum Road, Gainesville 32611, FL, USA