INTERACT: Achieving Low Sample and Communication Complexities in Decentralized Bilevel Learning over Networks

Zhuqing Liu[1], Xin Zhang[2], Prashant Khanduri[1,3], Songtao Lu[4], and Jia Liu[1]

[1]Department of Electrical and Computer Engineering, The Ohio State University
[2]Department of Statistics, Iowa State University
[3]Department of Electrical and Computer Engineering, University of Minnesota
[4]IBM Research AI, IBM Thomas J. Watson Research Center

ABSTRACT

In recent years, decentralized bilevel optimization problems have received increasing attention in the networking and machine learning communities. However, for decentralized bilevel optimization over networks with limited computation and communication capabilities, how to achieve low sample and communication complexities are two fundamental challenges. In this paper, we make the first attempt to investigate the class of decentralized bilevel optimization problems with nonconvex and strongly-convex structure corresponding to the outer and inner subproblems, respectively. Our main contributions in this paper are two-fold: i) We first propose a deterministic algorithm called INTERACT (inner-gradient-descent-outer-tracked-gradient) that requires the sample complexity of $O(n\epsilon^{-1})$ and communication complexity of $O(\epsilon^{-1})$ to solve the bilevel optimization problem, where n and $\epsilon > 0$ are the number of samples at each agent and the desired stationarity gap, respectively. ii) To relax the need for full gradient evaluations in each iteration, we propose a stochastic variance-reduced version of INTERACT (SVR-INTERACT), which improves the sample complexity to $O(\sqrt{n}\epsilon^{-1})$ while achieving the same communication complexity as the deterministic algorithm. Our numerical experiments also corroborate our theoretical findings.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Networks → Network performance analysis**.

KEYWORDS

Bilevel learning, network consensus optimization

1 INTRODUCTION

In recent years, fueled by the rise of machine learning and artificial intelligence in edge networks, decentralized bilevel optimization problems have received increasing attention in the networking and machine learning communities. This is due to the versatility of decentralized bilevel optimization in supporting many decentralized learning paradigms over peer-to-peer networks, such as the *multi-agent versions* of meta learning [22, 34, 34], hyperparameter optimization problem[25, 30], area under curve (AUC) problems [19, 33], and reinforcement learning[9, 41]. To date, however, there remain many challenges and open problems in decentralized bilevel learning over peer-to-peer networks. Two of the most fundamental challenges in decentralized bilevel optimization are how to achieve low sample and communication complexities. The need for decentralized bilevel optimization with low sample- and communication-complexities is particularly compelling in peer-to-peer edge networks, where the nodes (mobile devices, sensors, UAVS, etc.) are typically limited in their computation and communication capabilities.

Mathematically, decentralized bilevel optimization problems share the same structure as their single-agent counterpart in that such problems contain an outer objective function that is in turn dependent on the optimal parameter values of an inner objective. Interestingly, the single-agent version of bilevel optimization is a class of challenging optimization problems in its own right due to the inherent non-convexity in these problems, and thus having received a significant amount of attention recently. To date, many algorithmic ideas have been proposed for single-agent bilevel optimization problems, such as double-loop iterative methods [5, 6, 10], single-level approach for reformulated bilevel problems [17, 28], and simultaneous upper-lower-level updates [3, 4, 8, 13, 14] (see Section 2 for more detailed discussions).

Compared to conventional single-agent bilevel optimization problems, designing effective and efficient algorithms for solving decentralized bilevel stochastic optimization problems remains underexplored. This is in large part due to the fact that using techniques designed for single-agent bilevel optimization faces several technical challenges when utilized for decentralized bilevel optimization. In fact, most of aforementioned algorithmic ideas for single-agent bilevel optimization are not applicable in decentralized multi-agent version of bilevel optimization. One of the fundamental reasons is that, instead of having only one optimization task in the inner subproblem, there are *multiple* inner tasks that need to be handled in the decentralized multi-agent setting, which renders techniques for single-agent bilevel optimization infeasible and often implies the challenges of efficiency and scalability.

To address the efficiency and scalability for solving bilevel optimization problem with multiple inner subproblems, a natural idea is to leverage the decentralized network-consensus optimization approach [40, 42], where multiple agents distributively and collaboratively solve a global learning task. However, in developing network-consensus approaches for decentralized bilevel optimization, two fundamental challenges arise: First, the tightly coupled inner-outer mathematical structure, together with the decentralized nature and the non-convexity of the outer problem, makes the design and theoretical analysis of the algorithms far more challenging compared to solving conventional single-level minimization problems. Second, due to the potential low-speed network connections and the large variability of stochastic gradient-type information in geographically-dispersed edge networks, network-consensus approaches for decentralized bilevel optimization are very sensitive to communication and sample complexities. These challenges motivate us to design effective and efficient network-consensus-based algorithms for decentralized bilevel learning, which strike a good balance between sample and communication complexities.

The major contribution of this paper is that we propose a series of new algorithmic techniques to overcome the aforementioned challenges and achieve both low sample and communication complexities for decentralized bilevel optimization. Our main technical contributions are summarized below:

- We propose a decentralized bilevel optimization algorithm called INTERACT (inner-gradient-descent-outer-tracked-gradient). We show that, INTERACT enjoys a convergence rate of $O(1/T)$ to achieve an ϵ-stationary point, where T is the maximum number of iterations. As a consequence, INTERACT achieves $[O(n\epsilon^{-1}), O(\epsilon^{-1})]$ sample-communication complexity per agent, where n is the total data size at each agent and ϵ is the desired accuracy. To our knowledge, INTERACT is the first algorithm for decentralized bilevel optimization with such low sample and communication complexities (please see Table 1).

- To lower the sample complexity of INTERACT , we propose another enhanced version called SVR-INTERACT using stochastic vaiance reduction techniques, which further reduces the sample complexity of INTERACT while retaining the same communication complexity order. Specifically, we show that SVR-INTERACT retains the same $O(\epsilon^{-1})$ communication complexity as of INTERACT while achieving lower sample complexity of $O(\sqrt{n}\epsilon^{-1})$.

- To examine the performance of algorithm INTERACT and SVR-INTERACT and verify our theoretical results, we conduct experiments on meta-learning tasks with the MNIST [18] and CIFAR-10 [16]. Our results show that the proposed INTERACT and SVR-INTERACT algorithms outperform other baseline algorithms. Also, we numerically show that SVR-INTERACT enjoys low sample and communication complexities.

The rest of the paper is organized as follows. In Section 2, we review related literature. In Section 3, we provide the preliminaries of the decentralized bilevel optimization problems. In Section 4 and Section 5, we propose two algorithms, namely INTERACT and SVR-INTERACT . Also, the convergence rate, communication complexity, and sample complexity of these two algorithms are provided. Section 6 provides numerical results to verify our theoretical findings, and Section 7 concludes this paper.

Table 1: Comparisons among algorithms for bilevel optimization problems, where ϵ denotes the ϵ-stationary point defined in (2) and n is the size of dataset at each agent. Fos BSA, we denote sample complexities as (Outer, Inner) as to achieve the ϵ-staionary point the algorithm requires different number of outer and inner samples. Note that the rest of the algorithms require the same number of outer and inner samples to achieve ϵ-stationary point.

Algorithms	Sample Complex.	Decentralized	Multi LL tasks
BSA [6]	$(O(\epsilon^{-2}), O(\epsilon^{-3}))$	✗	✗
TTSA [9]	$O(\epsilon^{-5/2})$	✗	✗
stocBiO [10]	$O(\epsilon^{-2})$	✗	✗
MSTSA [13]	$O(\epsilon^{-2})$	✗	✗
STABLE [3]	$O(\epsilon^{-2})$	✗	✗
ALSET [4]	$O(\epsilon^{-2})$	✗	✗
SUSTAIN [14]	$O(\epsilon^{-1.5})$	✗	✗
VRBO [39]	$O(\epsilon^{-1.5})$	✗	✗
RSVRB [8]	$O(\epsilon^{-1.5})$	✗	✓
ITD-BiO/AID-BiO [10]	$O(n\epsilon^{-1})$	✗	✗
INTERACT (Ours)	$O(n\epsilon^{-1})$	✓	✓
SVR-INTERACT (Ours)	$O(\sqrt{n}K\epsilon^{-1} + n)$	✓	✓

2 RELATED WORK

1) Bilevel Optimization Approaches: Classical approaches to solve bilevel optimization problems include: i) reformulating the bilevel problem to a single-level problem and replacing the inner-level problem by its KKT conditions [36], or ii) utilizing penalty based approaches [27]. Motivated by various machine learning applications, gradient-based techniques have recently become the most popular strategies for solving bilevel optimization problems. A variety of explicit gradient-based methods have been investigated (e.g., AID-based methods [10, 34] and ITD-based methods [10, 31]). Due to large memory requirements and high sample complexities, these methods face challenges when implemented for machine learning applications, especially if the training data sizes are large.

To overcome this challenge, a stochastic gradient descent (SGD) based double-loop algorithm called BSA [6] was recently proposed for bilevel optimization. BSA provided the first finite-time convergence guarantees for solving non-convex-strongly-convex bilevel optimization problem. The convergence performance of BSA was improved by stocBiO [10], however at the cost of evaluating very large batch-size gradients in each iteration for both outer and inner subproblems (see Table 1 for a detailed comparison). Recently, motivated by sequential games, many single-loop algorithms have been proposed for solving the bilevel problems, where the outer and inner level problem's iterates are updated simultaneously [3, 4, 8, 9, 13, 14, 39]. Initially, it was shown in [9] that TTSA, a vanilla SGD-based algorithm that updates inner and outer level problem's iterates simultaneously, requires $O(\epsilon^{-5/2})$ samples of both inner and outer-level functions to achieve ϵ-stationarity. Later, this rate was improved by MSTSA [13] and STABLE [3] to $O(\epsilon^{-2})$ by utilizing momentum-based gradient estimators for the outer objective. Recently, [4] improved the analysis of TTSA and showed that vanilla SGD-based updates for outer- and inner subproblems

can in fact achieve $O(\epsilon^{-2})$ rate. This performance was further improved by SUSTAIN [14] and RSVRB [8] by utilizing momentum-based gradient estimators for both outer- and inner-level updates. It was shown that SUSTAIN and RSVRB match the best complexity bounds of $O(\epsilon^{-1.5})$ as the optimal SGD algorithm for solving single-level optimization problems. Another approach to accelerate traditional SGD based methods is the family of "variance-reduced" (VR) methods, which are referred to as VRBO [39]. VRBO adopts recursive gradient estimators to achieve the best known sample complexity of $O(\epsilon^{-1.5})$.

Orthogonal to the aforementioned approaches, which focus developing bilevel optimization algorithms only for a single agent (centralized), in this paper, we focus on a multi-agent (decentralized) setting to solve a bilevel optimization problem over a network of agents. To this end, we propose two novel decentralized bilevel algorithms named INTERACT and SVR-INTERACT using gradient descent and variance reduction techniques. We show that INTERACT achieves the sample complexity of $O(n\epsilon^{-1})$, while SVR-INTERACT achieves the sample complexity of $O(\sqrt{n}\epsilon^{-1})$. We summarize and compare the complexity results of the above state-of-the-art algorithms in Table 1.

2) Decentralized Learning over Networks: From a mathematical point of view, conducting decentralized learning over a network amounts to a group of agents collectively solving an optimization problem. Most of the existing literature of decentralized learning are focused on the standard loss minimization formulation[40, 42], i.e., $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$, where $f(\cdot)$ is the objective loss function and $\mathbf{x}$ denotes the global model parameters to be learned, and d is the model dimension. In the literature, a wide range of machine learning applications can be modeled by the standard decentralized loss minimization formulation (e.g., robotic network [15, 32], network resource allocation [12, 35], power networks [2, 7]). Some recent works, [20, 21, 26, 43] studied decentralized min-max optimization problems, i.e., $\min_{\mathbf{x}\in\mathbb{R}^{d_1}} \max_{\mathbf{y}\in\mathbb{R}^{d_2}} f(\mathbf{x}, \mathbf{y})$, which are a special case (with same outer and inner level objective) of bilevel optimization problems. Unfortunately, studies on solving general decentralized bilevel optimization problems are still limited. Existing works in [1, 38] applied the fuzzy goal programming method for linearization of functions based on Jacobian matrix for solving multiobjective bilevel problem. However, none of these work tackles decentralized bilevel optimization via the network consensus approach.

3 SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we will describe the system model and the problem formulation first. Then, we will provide several application examples to further motivate the decentralized nonconvex-strongly-convex bilevel optimization problems studied in this paper.

1) Network Consensus Formulation: We represent the underlying peer-to-peer communication network with m agents by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the set of agents and edges, respectively, with $|\mathcal{N}| = m$. Each agent can share information with its neighbors. For agent $i \in [m]$, we denote its set of neighbors by $\mathcal{N}_i$, i.e., $\mathcal{N}_i \triangleq \{j \in \mathcal{N}, : (i, j) \in \mathcal{L}\}$. The decentralized bilevel

optimization (DBO) problem can be written as:

$$\min_{\mathbf{x}_i\in\mathbb{R}^{d_1}} \ell(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \ell_i(\mathbf{x}_i) = \frac{1}{m}\sum_{i=1}^m f_i\left(\mathbf{x}_i, \mathbf{y}_i^*(\mathbf{x}_i)\right)$$

$$= \frac{1}{mn}\sum_{i=1}^m\sum_{j=1}^n f_i\left(\mathbf{x}_i, \mathbf{y}_i^*(\mathbf{x}_i); \xi_{ij}\right),$$

$$\text{s.t. } \mathbf{y}_i^*(\mathbf{x}_i) = \arg\min_{\mathbf{y}_i} g_i(\mathbf{x}_i, \mathbf{y}_i) = \arg\min_{\mathbf{y}_i} \frac{1}{n}\sum_{j=1}^n g_i(\mathbf{x}_i, \mathbf{y}_i; \xi_{ij}),$$

$$\mathbf{x}_i = \mathbf{x}_j, \text{ if } i, j \in \mathcal{L}, \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the local copies of the outer and inner-level variables at agent $i \in [m]$. Each agent has access to a local dataset of size n. The local loss is defined as: $f_i(\mathbf{x}_i, \mathbf{y}_i^*(\mathbf{x}_i)) \coloneqq \frac{1}{n}\sum_{j=1}^n f_i(\mathbf{x}_i, \mathbf{y}_i; \xi_{ij})$. The equality constraint in (1) ensures a "consensus" among the outer level variables of each individual agents. Our goal is to solve Problem (1) in a *decentralized* manner, where $\ell(x)$ is non-convex while the inner-level funtions $g_i(\mathbf{x}, \mathbf{y})$ are strongly-convex with respect to $\mathbf{y}$ for all $i \in [m]$. In the rest of this paper, we will refer to (1) as a decentralized nonconvex-strong-convex bilevel problem.

Next, we define the notion of ϵ-stationary point. We define $\{\mathbf{x}_i, \mathbf{y}_i, \forall i \in [m]\}$ as an ϵ-stationary point if it satisfies:

$$\underbrace{\frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2}_{\text{Consensus Error}} + \underbrace{\|\mathbf{y}^* - \mathbf{y}\|^2}_{\substack{\text{Lower-Level}\\\text{Error}}} + \underbrace{\|\nabla\ell(\bar{\mathbf{x}})\|^2}_{\substack{\text{Stationarity}\\\text{Error}}} \leq \epsilon, \tag{2}$$

where $\bar{\mathbf{x}} \triangleq \frac{1}{m}\sum_{i=1}^m \mathbf{x}_i$, $\mathbf{y} \triangleq [\mathbf{y}_1^\top, ... \mathbf{y}_m^\top]^\top$, and $\mathbf{y}^* \triangleq [\mathbf{y}_1^{*\top}, ... \mathbf{y}_m^{*\top}]^\top$. Next, we define the notions of sample and communication complexities [37] of a decentralized algorithm to achieve an ϵ-stationary point defined in (2).

DEFINITION 1 (SAMPLE COMPLEXITY). *The sample complexity is defined as the total number of incremental first-order oracle (IFO) calls required per node for an algorithm to reach an ϵ-stationary point. Note that we define one IFO call as the evaluation of the (stochastic) gradient of upper and lower level problems at node $i \in [m]$.*

DEFINITION 2 (COMMUNICATION COMPLEXITY). *We define a communication round as the sharing and receiving of local parameters by each node from all its neighboring nodes. Then, the communication complexity is defined as the total rounds of communications required by an algorithm to achieve an ϵ-stationary point.*

2) Motivating Application Examples: Problem (1) can be applied to a number of interesting real-world decentralized machine learning problems. Here, we provide two examples to further motivate its practical relevance:

- *Multi-agent meta-learning [34]:* Meta-learning (or learning to learn) is a powerful tool for quickly learning new tasks by using the prior experience from related tasks. Consider a meta-learning task with m lower level problems. There are m agents who collectively solve this meta-learning problem over a network. This problem can be formulated as:

$$\min_{\mathbf{x}\in\mathbb{R}^{d_1}} \sum_{i=1}^m f_i\left(\mathbf{x}, \mathbf{y}_i^*(\mathbf{x})\right)$$

$$\mathbf{y}_i^*(\mathbf{x}) \in \arg\min_{\mathbf{y}_i} g_i\left(\mathbf{x}, \mathbf{y}_i\right), i = 1, \ldots, m, \tag{3}$$

where agent $i \in [m]$ has a local dataset with n samples, and f_i corresponds to the loss function for the task of agent i. Parameter $\mathbf{x} \in \mathbb{R}^{d_1}$ is the model parameters shared among all agents, and $\mathbf{y}_i \in \mathbb{R}^{d_2}$ denotes task-specific parameters solved by each agent.

- *Multi-agent AUPRC optimization [8]:* Another application of (1) is the multi-agent optimization of area under precision-recall curve (AUPRC) [33]. Multi-agent AUPRC optimization problem can be formulated as a bilevel optimization problem with many lower-level problems as:

$$\min_{\mathbf{x} \in \mathbb{R}^{d_1}} \sum_{i=1}^{m} f_i\left(\mathbf{y}_i^*(\mathbf{x})\right)$$

$$\mathbf{y}_i^*(\mathbf{x}) = \arg\min_{\mathbf{y}_i} -\mathbf{y}_i^\top g_i\left(\mathbf{x}\right) + \frac{1}{2}\|\mathbf{y}_i\|^2, i = 1, \dots, m,$$

where each $i \in [n]$ corresponds to a binary classification task and $\mathbf{x}$ denotes the model parameter.

With the system model and the problem formulation in (1), we are in a position to present our algorithm design in the next section.

4 THE INTERACT ALGORITHM

In this section, we present our INTERACT algorithm for solving the decentralized bilevel optimization problems and provide its theoretical convergence guarantees.

4.1 Algorithm Description

To solve Problem (1), we adopt the network consensus approach [29]. In each iteration, every agent shares and receives information from its neighboring nodes. Each agent aggregates the information received from its neighbours through a consensus weight matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$. We denote by $[\mathbf{M}]_{ij}$ the element in the i-th row and the j-th column in $\mathbf{M}$. We assume that $\mathbf{M}$ satisfies the following:

(a) *Doubly stochastic:* $\sum_{i=1}^{m} [\mathbf{M}]_{ij} = \sum_{j=1}^{m} [\mathbf{M}]_{ij} = 1$;
(b) *Symmetric:* $[\mathbf{M}]_{ij} = [\mathbf{M}]_{ji}, \forall i, j \in \mathcal{N}$;
(c) *Network-Defined Sparsity:* $[\mathbf{M}]_{ij} > 0$ if $(i, j) \in \mathcal{L}$; otherwise $[\mathbf{M}]_{ij} = 0, \forall i, j \in \mathcal{N}$.

The above properties imply that the eigenvalues of $\mathbf{M}$ are real and lie in the interval $(-1, 1]$. We sort the eigen values of $\mathbf{M}$ as: $-1 < \lambda_m(\mathbf{M}) \leq \cdots \leq \lambda_2(\mathbf{M}) < \lambda_1(\mathbf{M}) = 1$. We denote the second-largest eigenvalue in magnitude of $\mathbf{M}$ as $\lambda \triangleq \max\{|\lambda_2(\mathbf{M})|, |\lambda_m(\mathbf{M})|\}$. We note that λ plays an important role in the step-size selection for our proposed algorithms (cf. Theorems 1 and 3). Next, using the implicit function theorem, we evaluate the hyper-gradient of $l_i(\mathbf{x}_i) := f_i(\mathbf{x}_i, \mathbf{y}_i^*(\mathbf{x}_i))$ at $\mathbf{x}_{i,t}$ and $\mathbf{y}_i^*(\mathbf{x}_{i,t})$ as:

$$\nabla \ell_i(\mathbf{x}_{i,t}) = \nabla_{\mathbf{x}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_i^*(\mathbf{x}_{i,t})) - \nabla_{\mathbf{xy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_i^*(\mathbf{x}_{i,t})) \times$$
$$[\nabla_{\mathbf{yy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_i^*(\mathbf{x}_{i,t}))]^{-1} \nabla_{\mathbf{y}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_i^*(\mathbf{x}_{i,t})). \quad (4)$$

Note that computing the local gradient $\nabla l_i(\mathbf{x}_{i,t}), i \in [m]$ requires knowledge of the lower level problem's optimal solution, $\mathbf{y}_i^*(\mathbf{x}_i)$. However, obtaining $\mathbf{y}_i^*(\mathbf{x}_i)$ in closed form is usually not feasible. Therefore, we replace $\mathbf{y}_i^*(\mathbf{x}_i)$ by an approximate solution $\mathbf{y}_{i,t}$, thereby, defining the following approximate gradient estimate [6]:

$$\bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) = \nabla_{\mathbf{x}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \nabla_{\mathbf{xy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) \times$$
$$[\nabla_{\mathbf{yy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})]^{-1} \nabla_{\mathbf{y}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}). \quad (5)$$

Algorithm 1 The INTERACT Algorithm.

Set parameter pair $\forall i \in [m], (\mathbf{x}_{i,0}, \mathbf{y}_{i,0}) = (\mathbf{x}^0, \mathbf{y}^0)$.
At each agent $i \in [m]$, compute the local gradients:

$$\mathbf{u}_{i,0} = \bar{\nabla} f_i(\mathbf{x}_{i,0}, \mathbf{y}_{i,0}); \ \mathbf{v}_{i,0} = \nabla_{\mathbf{y}} g_i(\mathbf{x}_{i,0}, \mathbf{y}_{i,0});$$

for $t = 1, \cdots, T$ **do**
    Update local parameters using (6) and (7);
    Compute local gradients using (8) and (9);
    Track the gradients using (10);
end for

The proposed INTERACT algorithm for each agent $i \in [m]$ is illustrated in Algorithm 1. Note that, to achieve state-of-the-art convergence guarantees, INTERACT relies on consensus updates along with gradient tracking for solving Problem (1) as follows:

- *Step 1 (Consensus update with gradient descent):* The local parameters of each agent $i \in [m]$ are updated using (6). Note that the consensus is established by only sharing the information of the outer model parameters $\mathbf{x}_i$, while the inner-model parameters $\mathbf{y}_i$ are updated only locally and the local gradient is computed by just evaluating the local gradient of the lower level objective:

$$\mathbf{x}_{i,t} = \sum_{j \in \mathcal{N}_i} [\mathbf{M}]_{ij} \mathbf{x}_{j,t-1} - \alpha \mathbf{u}_{i,t-1}, \quad (6)$$

$$\mathbf{y}_{i,t} = \mathbf{y}_{i,t-1} - \beta \mathbf{v}_{i,t-1}, \quad (7)$$

where α and β are the step-sizes for updating $\mathbf{x}$- and $\mathbf{y}$-variables, respectively, $\mathbf{v}_{i,t}$ is the local gradient, and $\mathbf{u}_{i,t}$ is an auxiliary vector for gradient tracking purposes and will be defined shortly.

- *Step 2 (Local gradient estimate):* Each agent $i \in [m]$ evaluates its (full) local gradient $\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ and $\mathbf{d}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ using as:

$$\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) = \bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}), \quad (8)$$

$$\mathbf{v}_{i,t} = \mathbf{d}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) = \nabla_{\mathbf{y}} g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}). \quad (9)$$

- *Step 3 (Local gradient tracking):* Each agent $i \in [m]$ updates the global gradient estimate $\mathbf{u}_{i,t}$ by averaging over its neighboring global gradient estimates:

$$\mathbf{u}_{i,t} = \sum_{j \in \mathcal{N}_i} [\mathbf{M}]_{ij} \mathbf{u}_{j,t-1} + \mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbf{p}_i(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1}). \quad (10)$$

Note that estimating $\mathbf{v}_{i,t}$ does not require gradient tracking as the inner problem can be solved locally at each agent (cf. Problem (1)).

4.2 Convergence Results of INTERACT

In this subsection, we focus on the theoretical guarantees associated with INTERACT (Algorithm 1). Providing convergence guarantees of INTERACT for solving the decentralized bilevel optimization problem in (1) presents two major challenges: *a) Inner-loop error:* Recall that, in constructing the true gradient estimate (4), we utilize a crude approximation $\mathbf{y}_{i,t} \neq \mathbf{y}_{i,t}^*(\mathbf{x}_{i,t})$. This approximation induces an error in the estimated gradient of (cf. (4)) that presents a major challenge in the analysis of INTERACT ; *b) Decentralized topology:* Since INTERACT applies decentralized consensus in outer-loop iterations, the analysis needs to capture the effect of the errors induced by the decentralized topology. These two issues make

the convergence analysis of INTERACT difficult. In spite of these challenges, we will show that INTERACT provides state-of-the-art convergence guarantees for solving Problem (1).

1) Convergence metric: We propose the following *new* metric for decentralized bilevel problems, which is the key in determining all convergence results in the following sections:

$$\mathfrak{M}_t \triangleq \mathbb{E}[\|\nabla \ell(\bar{\mathbf{x}}_t)\|^2 + \|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 + \|\mathbf{y}_t^* - \mathbf{y}_t\|^2], \quad (11)$$

where $\mathbf{y}_t^*$ is defined in (2). Note that the first term in (11) quantifies the convergence of the $\bar{\mathbf{x}}_t$ to a stationary point of the global objective. The second term in (11) measures the consensus error among local copies of the outer variable, while the third term in (11) quantifies the (aggregated) error in the inner problem's iterates across all agents. Thus, if the iterates generated by an algorithm that achieves $\mathfrak{M}_t \to 0$, it implies that the algorithm achieves three goals simultaneously: 1) consensus of outer variables, 2) stationary point of Problem (1), and 3) solution to the inner problem.

2) Technical Assumptions: Next, before presenting the main convergence result of INTERACT , we first state the technical assumptions on the outer and inner objective functions in Problem (1):

ASSUMPTION 1. *The inner function g satisfies:*

a) *For any $i \in [m]$, and $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $g_i(\mathbf{x}_i, \mathbf{y}_i)$ is μ_g -strongly convex with respect to the variable $\mathbf{y}_i$, i.e., $\mu_g I \preceq \nabla_y^2 g(\mathbf{x}_i, \mathbf{y}_i)$.*

b) *For any $i \in [m]$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$, $\nabla g_i(\mathbf{x}_i, \mathbf{y}_i)$ is L_g-Lipschitz continuous with $L_g > 0$.*

c) *For any $i \in [m]$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$, we have $\left\|\nabla_{xy}^2 g_i(\mathbf{x}_i, \mathbf{y}_i)\right\|^2 \le C_{g_{xy}}$ for some $C_{g_{xy}} > 0$.*

d) *For any $i \in [m]$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$, we have $\nabla_{x,y}^2 g_i(\mathbf{x}_i, \mathbf{y}_i)$ and $\nabla_{yy}^2 g_i(\mathbf{x}_i, \mathbf{y}_i)$ are Lipschitz continuous with constants $L_{g_{xy}} > 0$ and $L_{g_{yy}} > 0$, respectively.*

ASSUMPTION 2. *The outer function f and $\ell(\mathbf{x})$ satisfy:*

a) *For any $i \in [m]$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\nabla_{\mathbf{x}} f_i(\mathbf{x}_i, \mathbf{y}_i)$, $\nabla_{\mathbf{y}} f_i(\mathbf{x}_i, \mathbf{y}_i)$ are Lipschitz continuous with constant $L_{f_x} \ge 0$, $L_{f_y} \ge 0$.*

b) *For $i \in [m]$, $\mathbf{x}_i \in \mathbb{R}^{d_1}$, $\mathbf{y}_i \in \mathbb{R}^{d_2}$, we have $\left\|\nabla_y f(\mathbf{x}_i, \mathbf{y}_i)\right\| \le C_{f_y}$ for some $C_{f_y} \ge 0$.*

c) *There exists a finite lower bound $\ell^* = \inf_{\mathbf{x}} \ell(\mathbf{x}) > -\infty$.*

Assumptions 1 and 2 are standard in the literature of bilevel optimization (e.g., [14][6]). Many problems of practical interest in meta-learning [34][22] and optimization of area under precision-recall curve(AUPRC) [33] can be shown to satisfy these assumptions.

3) Supporting Lemmas: Next, we present several lemmas that characterizes the Lipschitz properties of the hypergradient in (4), the approximate gradient in (5), and the optimal solution of the inner problem under Assumptions 1 and 2. These lemmas will be useful in proving our main convergence result (proofs directly follow from [6] due to structural similarities).

LEMMA 1. *Under Assumptions 1–2, we have*

$$\|\bar{\nabla} f_i(\mathbf{x}, \mathbf{y}) - \nabla \ell_i(\mathbf{x})\|^2 \le L_f \left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}\right\|^2,$$

$$\left\|\mathbf{y}_i^*(\mathbf{x}_1) - \mathbf{y}_i^*(\mathbf{x}_2)\right\|^2 \le L_y \|\mathbf{x}_1 - \mathbf{x}_2\|^2,$$

$$\|\nabla \ell_i(\mathbf{x}_1) - \nabla \ell_i(\mathbf{x}_2)\|^2 \le L_\ell \|\mathbf{x}_1 - \mathbf{x}_2\|^2,$$

for all $i \in [m]$ and $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d_1}$ and $\mathbf{y} \in \mathbb{R}^{d_2}$. The above Lipschitz constants are defined as: $L_f = \left(L_{f_x} + \frac{L_{f_y} C_{g_{xy}}}{\mu_g} + C_{f_y}\left(\frac{L_{g_{xy}}}{\mu_g} + \frac{L_{g_{yy}} C_{g_{xy}}}{\mu_g^2}\right)\right)^2$, $L_\ell = \left(L_f + \frac{L_f C_{g_{xy}}}{\mu_g}\right)^2$, $L_y = \left(\frac{C_{g_{xy}}}{\mu_g}\right)^2$.

LEMMA 2. *Under Assumption 1-2, we have*

$$\|\nabla f_i(\mathbf{x}_1, \mathbf{y}_1) - \nabla f_i(\mathbf{x}_2, \mathbf{y}_2)\|^2 \le L_K^2 [\|\mathbf{x}_1 - \mathbf{x}_2\|^2$$

$$+ \|\mathbf{y}_1 - \mathbf{y}_2\|^2], \forall \, \mathbf{x}_1, \, \mathbf{x}_2 \in \mathbb{R}^{d_1}, \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^{d_2}. \quad (12)$$

In the above expressions, $\exists L_K \ge L_{K_d}$ and L_{K_d} is defined as:

$$L_{K_d}^2 := 2L_{f_x}^2 + 6C_{gy}^2 \frac{1}{\mu_g^2} L_{f_y}^2 + 6C_{f_y}^2 \frac{1}{\mu_g^2} L_{g_{xy}}^2 + 6C_{g_{xy}}^2 C_{f_y}^2 L_{g_{yy}}^2 \frac{1}{\mu_g^4}. \quad (13)$$

4) Convergence Results of INTERACT : We state the main convergence rate result of INTERACT in the following theorem:

THEOREM 1. *(Convergence of INTERACT) Under Assumptions 1-2, if the step-sizes satisfy $\beta \le \min\{\frac{3(\mu_g+L_g)}{\mu_g L_g}, \frac{1}{\mu_g+L_g}\}$, $\alpha \le \min\{\frac{1}{4L_\ell}$, $\frac{1}{4L_K}\sqrt{\frac{1-\lambda}{2m}}, \frac{1}{m(1-\lambda)}, \frac{(1-\lambda)^2}{32L_K^2}, \frac{m(1-\lambda)}{4L_\ell}, \frac{9r^2 m(1-\lambda)}{32L_y^2(1+1/r)L_f^2}, \frac{(1-r)(1+r)r(1-\lambda)^2}{32L_y^2(\mu_g+L_g)L_K^2\beta}$, $\frac{1-\lambda}{4L_K}, 1\}$, $r = \frac{1}{3}\beta\frac{\mu_g L_g}{\mu_g+L_g}$, then $\{\mathbf{x}_t, \mathbf{y}_t\}$ generated by INTERACT satisfy*

$$\frac{1}{T+1}\sum_{t=0}^T \mathfrak{M}_t \le \frac{\mathfrak{B}_0 - \ell^*}{(T+1)\min\{\frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2}\}} = O\left(\frac{1}{T}\right),$$

where $\mathfrak{B}_t = \ell(\bar{\mathbf{x}}_t) + \|\mathbf{y}_t - \mathbf{y}_t^*\|^2 + \|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 + \alpha\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2$.

REMARK 1. Note from the statement of Theorem 1 that INTERACT requires constant step-sizes α and β that depend on the network topology, Lipschitz constants, and the number of agents. In particular, different network topology leads to different network consensus matrix $\mathbf{M}$. Recall that $\lambda < 1$ is the second-largest eigenvalue in magnitude of $\mathbf{M}$. For a dense network, λ is close to 0, which allows larger step-size α, and then leads to faster convergence.

Theorem 1 immediately implies the following sample and communication complexity results of INTERACT:

COROLLARY 2. *Under the conditions of Theorem 1, to achieve an ϵ-stationary solution, INTERACT requires: 1) Communication complexity of $O(\epsilon^{-1})$, and 2) Sample complexity of $O(n\epsilon^{-1})$.* □

4.3 Proof Sketches of the Convergence Results

Due to space limitation, we provide a proof sketch of Theorem 1. The proof details are provided in the full online technical report of this paper [23]. In this section, we organize the proof into several key steps. Our first step is to show the descent property of our algorithms, which is stated as follows:

Step 1) Descending Inequality for the Outer Function: Under Assumptions 1-2, the following inequality holds for Algorithm 1:

$$\ell(\bar{\mathbf{x}}_{t+1}) - \ell(\bar{\mathbf{x}}_t) \le -\frac{\alpha}{2}\|\nabla \ell(\bar{\mathbf{x}}_t)\|^2 - \left(\frac{\alpha}{2} - \frac{L_\ell \alpha^2}{2}\right)\|\bar{\mathbf{u}}_t\|^2$$

$$+ \frac{\alpha}{m}\sum_{i=1}^m L_\ell \|\bar{\mathbf{x}}_t - \mathbf{x}_{i,t}\|^2 + \frac{2\alpha}{m}\sum_{i=1}^m L_f^2\|\mathbf{y}_{i,t}^* - \mathbf{y}_{i,t}\|^2$$

$$+ 2\alpha\left\|\frac{1}{m}\sum_{i=1}^m \bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\right\|^2, \quad (14)$$

where $\mathbf{y}_{i,t}^* = \arg\min_{\mathbf{y}_{i,t}} g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$.

Eqs. (14) focuses on the descending upper bound for adjacent iterations of outer-level model parameters. Unlike traditional decentralized minimization optimization problems, decentralized bilevel optimization problems tackle a composition of an inner problem and an outer problem. This tightly coupled inner-outer mathematical structure in the bilevel optimization problem, together with the decentralized nature and the non-convexity of the outer problem, imposes significant challenges in the theoretical algorithm analysis. As a result, Eq. (14) contains the consensus error of the outer-level model parameters $\|\bar{\mathbf{x}}_t - \mathbf{x}_{i,t}\|^2$ and the convergence metric of inner model parameter $\|\mathbf{y}_{i,t}^* - \mathbf{y}_{i,t}\|^2$.

Step 2) Error Bound on $\mathbf{y}^*(\mathbf{x})$: Under Assumptions 1 and 2, the following inequality holds for Algorithm 1:

$$\|\mathbf{y}_{i,t+1} - \mathbf{y}_{i,t+1}^*\|^2 \leq (1+r)^2 (1 - 2\beta \frac{\mu_g L_g}{\mu_g + L_g})\|\mathbf{y}_{i,t} - \mathbf{y}_{i,t}^*\|^2$$

$$+ (\delta - 1)(1+r)^2 (2\beta \frac{1}{\mu_g + L_g} - \beta^2)\|\mathbf{v}_{i,t}\|^2 + (1 + \frac{1}{r})L_y^2\|\mathbf{x}_{i,t+1} - \mathbf{x}_{i,t}\|^2$$

$$+ [(\frac{1}{\delta} - 1)(1+r)^2 (2\beta \frac{1}{\mu_g + L_g} - \beta^2) + (1+r)(1+1/r)\beta^2]$$

$$\cdot \|\nabla_{\mathbf{y}} g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbf{v}_{i,t}\|^2. \tag{15}$$

In Step 2, we continue evaluating the fourth element in Eq. (14). From the result of Step 2, we can see that $\|\mathbf{y}_{i,t} - \mathbf{y}_{i,t}^*\|^2$ shrinks if $2(1+r)(1 - 2\beta \frac{\mu_g L_g}{\mu_g + L_g}) \leq 1$.

Step 3) Iterates Contraction: The following contraction properties of the iterates hold:

$$\|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 \leq (1+c_1)\lambda^2 \|\mathbf{x}_{t-1} - \mathbf{1} \otimes \bar{\mathbf{x}}_{t-1}\|^2$$
$$+ \left(1 + \frac{1}{c_1}\right)\alpha^2 \|\mathbf{u}_{t-1} - \mathbf{1} \otimes \bar{\mathbf{u}}_{t-1}\|^2, \tag{16}$$

$$\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2 \leq (1+c_2)\lambda^2 \|\mathbf{u}_{t-1} - \mathbf{1} \otimes \bar{\mathbf{u}}_{t-1}\|^2$$
$$+ (1 + \frac{1}{c_2})\|\mathbf{p}_t - \mathbf{p}_{t-1}\|^2, \tag{17}$$

where c_1 and c_2 are some positive constants.

Step 3 is important in analyzing the convergence performance of the our proposed INTERACT . The key step is to define $\widetilde{\mathbf{M}} = \mathbf{M} \otimes \mathbf{I}_m$. Then, we have $\|\widetilde{\mathbf{M}}\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 = \|\widetilde{\mathbf{M}}(\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t)\|^2 \leq \lambda^2 \|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2$. This is because $\mathbf{x}_t - \mathbf{1} \otimes \mathbf{x}_t$ is orthogonal to $\mathbf{1}$, which is the eigenvector corresponding to the largest eigenvalue of $\widetilde{\mathbf{M}}$, and $\lambda = \max\{|\lambda_2|, |\lambda_m|\}$.

Step 4) Potential Function: With results from Steps 1-3, we have:

$$\ell(\bar{\mathbf{x}}_{T+1}) - \ell(\bar{\mathbf{x}}_0) + \frac{1-\lambda}{32(1+1/r)L_y^2}\left[\|\bar{\mathbf{y}}_{T+1} - \mathbf{y}_{T+1}^*\|^2 - \|\mathbf{y}_0^* - \bar{\mathbf{y}}_0\|^2\right]$$

$$\leq -\frac{\alpha}{2}\sum_{t=0}^{T}\|\nabla\ell(\bar{\mathbf{x}}_t)\|^2 - (\frac{\alpha}{2} - \frac{L_\ell \alpha^2}{2} - 4(1 + \frac{1}{r})L_y^2\alpha^2 m\frac{1-\lambda}{32(1+1/r)L_y^2})$$

$$\cdot \sum_{t=0}^{T}\|\bar{\mathbf{u}}_t\|^2 + (\frac{\alpha L_\ell}{m} + (1 + \frac{1}{r})L_y^2\frac{1-\lambda}{4(1+1/r)L_y^2})\sum_{t=0}^{T}\|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2$$

$$+ [\frac{2\alpha}{m}L_f^2 + \frac{1-\lambda}{32(1+1/r)L_y^2}3r - \frac{1-\lambda}{32(1+1/r)L_y^2}(1+3r)(2\beta\frac{\mu_g L_g}{\mu_g + L_g})]$$

$$\cdot \sum_{t=0}^{T}\|\mathbf{y}_t - \mathbf{y}_t^*\|^2 + 2\alpha\sum_{t=0}^{T}\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\|^2$$

$$+ \frac{1-\lambda}{32(1+1/r)L_y^2}(\delta - 1)(1+r)^2(2\beta\frac{1}{\mu_g + L_g} - \beta^2)\sum_{t=0}^{T}\|\mathbf{v}_t\|^2$$

$$+ 4(1 + \frac{1}{r})L_y^2\alpha^2\frac{1-\lambda}{32(1+1/r)L_y^2}(\sum_{t=0}^{T}\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2)$$

$$+ \frac{1-\lambda}{32(1+1/r)L_y^2} \cdot [\frac{(1+r)^2\beta^2}{r} + (\frac{1}{\delta} - 1)(1+r)^2(2\beta\frac{1}{\mu_g + L_g} - \beta^2)]$$

$$\cdot \sum_{i=1}^{m}\sum_{t=0}^{T}\|\mathbf{v}_{i,t} - \nabla_{\mathbf{y}} g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})\|^2. \tag{18}$$

Note that the coefficients of $\|\bar{\mathbf{u}}_t\|^2$ and $\|\mathbf{v}_t\|^2$ can be made negative when step-sizes are chosen properly and $\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\|^2$ can be decomposed later, we define our potential function as:

$$\mathfrak{B}_t = \ell(\bar{\mathbf{x}}_t) + \frac{(1-\lambda)\|\mathbf{y}_t - \mathbf{y}_t^*\|^2}{32(1+1/r)L_y^2} + \|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 + \alpha\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2.$$

Step 5) Main proof of Theorem 1: Note that we use the full gradient estimator in INTERACT, thus we have $\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\|^2 = 0$. Also, we have

$$\|\mathbf{p}_t - \mathbf{p}_{t-1}\|^2 \leq L_K^2 m(\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \beta^2\|\mathbf{v}_{t-1}\|^2), \tag{19}$$

where $\mathbf{p}_t = \left[\mathbf{p}_1(\mathbf{x}_{1,t}, \mathbf{y}_{1,t})^\top, \cdots, \mathbf{p}_m(\mathbf{x}_{m,t}, \mathbf{y}_{m,t})^\top\right]^\top$.

Combing the above results and choosing $c_1 = c_2 = \frac{1}{\lambda} - 1$ yield:

$$\ell(\bar{\mathbf{x}}_{T+1}) - \ell(\bar{\mathbf{x}}_0) + \frac{1-\lambda}{32(1+1/r)L_y^2}\left[\|\bar{\mathbf{y}}_{T+1} - \mathbf{y}_{T+1}^*\|^2 - \|\mathbf{y}_0^* - \bar{\mathbf{y}}_0\|^2\right]$$

$$+ [\|\mathbf{x}_{T+1} - \mathbf{1} \otimes \bar{\mathbf{x}}_{T+1}\|^2 - \|\mathbf{x}_0 - \mathbf{1} \otimes \bar{\mathbf{x}}_0\|^2]$$

$$+ \alpha[\|\mathbf{u}_{T+1} - \mathbf{1} \otimes \bar{\mathbf{u}}_{T+1}\|^2 - \|\mathbf{u}_0 - \mathbf{1} \otimes \bar{\mathbf{u}}_0\|^2]$$

$$\leq -\frac{\alpha}{2}\sum_{t=0}^{T}\|\nabla\ell(\bar{\mathbf{x}}_t)\|^2 + C_1\sum_{t=0}^{T}\|\bar{\mathbf{u}}_t\|^2 + C_2\sum_{t=0}^{T}\|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2$$

$$+ C_3\sum_{t=0}^{T}\|\mathbf{y}_t - \mathbf{y}_t^*\|^2 + C_4\sum_{t=0}^{T}\|\mathbf{v}_t\|^2 + C_5\sum_{t=0}^{T}\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2, \tag{20}$$

where the constants are relegated to our technical report [23]. With appropriately chosen parameters to ensure $C_1, C_4, C_5 \leq 0, C_2 \leq -\frac{1-\lambda}{4}, C_3 \leq -\frac{3r^2(1-\lambda)}{32(1+r)L_y^2}$, we have the following convergence results:

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathfrak{M}_t \leq \frac{\mathfrak{B}_0 - \ell^*}{(T+1)\min\{\frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2}\}}. \tag{21}$$

This completes the proof of Theorem 1.

5 THE SVR-INTERACT ALGORITHM

Note that, in the INTERACT algorithm (Algorithm 1), each agent needs to evaluate the full local gradient in each iteration, which might not be feasible for many large-scale problems. This motivates us to develop a stochastic version of INTERACT that circumvents the need to compute full local gradients in each iteration. To this end, we leverage the variance reduce methods to design SVR-INTERACT,

a stochastic version of INTERACT that achieves low sample complexity for solving decentralized bilevel optimization problem. In this section, we first present the SVR-INTERACT method, and then provide its convergence guarantees.

In particular, we further define an (approximate) stochastic gradient estimate of the local gradient in (5). We define the stochastic gradient of $\bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ by $\bar{\nabla} f_i(x_{i,t}, y_{i,t}; \bar{\xi}_i)$ as

$$\bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i) = \nabla_{\mathbf{x}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \xi_i^0) - \frac{K}{L_g} \nabla_{\mathbf{xy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \zeta_i^0)$$

$$\cdot \prod_{j=1}^{\mathrm{k}(K)} (I - \frac{\nabla_{\mathbf{yy}}^2 g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \zeta_i^j)}{L_g}) \nabla_{\mathbf{y}} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \xi_i^0), \qquad (22)$$

where $\mathrm{k}(K) \sim \mathcal{U}\{0, \ldots, K-1\}$ denotes a random variable uniformly chosen from $\{0, \ldots, K-1\}$. Note that the stochastic gradient estimator collects $K + 2 \in \mathbb{N}$ independent samples $\bar{\xi}_i := \left\{\xi_i', \xi_i^0, \ldots, \xi_i^K, \mathrm{k}(K)\right\}$, where $\xi_i', \xi_i^j \sim \mu, \zeta_i^j \sim \pi, j = 0, \ldots, K$ denote the samples of the outer and inner objectives. Next, we define the convergence metric that we will utilize for solving the decentralized bilevel learning problem.

5.1 Algorithm Description

In SVR-INTERACT, we use the same network consensus approach as in INTERACT.

The proposed SVR-INTERACT algorithm is illustrated in Algorithm 2. In each iteration, every agent $i \in [m]$ estimates its full gradients every q steps. For other iterations, when $mod(t, q) \neq 0$, SVR-INTERACT estimates local gradients $\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ and $\mathbf{d}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ using the following gradient estimators:

$$\mathbf{p}_t = \mathbf{p}_{t-1} + \frac{1}{|\mathcal{S}|} \sum_{i=1}^{\mathcal{S}} \left[\bar{\nabla} f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i) - \bar{\nabla} f_i(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1}; \bar{\xi}_i) \right]; \quad (23)$$

$$\mathbf{d}_t = \mathbf{d}_{t-1} + \frac{1}{|\mathcal{S}|} \sum_{i=1}^{\mathcal{S}} \left[\nabla g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i') - \nabla g_i(\mathbf{x}_{i,t-1}, \mathbf{y}_{i,t-1}; \bar{\xi}_i') \right], \quad (24)$$

where we define:

$$\mathbf{p}_t \triangleq \left[\mathbf{p}_1(\mathbf{x}_{1,t}, \mathbf{y}_{1,t})^\top, \cdots, \mathbf{p}_m(\mathbf{x}_{m,t}, \mathbf{y}_{m,t})^\top \right]^\top;$$
$$\mathbf{d}_t \triangleq \left[\mathbf{d}_1(\mathbf{x}_{1,t}, \mathbf{y}_{1,t})^\top, \cdots, \mathbf{d}_m(\mathbf{x}_{m,t}, \mathbf{y}_{m,t})^\top \right]^\top. \quad (25)$$

Note that in contrast to INTERACT, SVR-INTERACT utilizes a variance-reduced gradient estimator. Moreover, SVR-INTERACT makes use of gradient tracking and consensus updates similar to the INTERACT algorithm. Next, we will present the convergence rate results of SVR-INTERACT.

5.2 Convergence Results of SVR-INTERACT

For SVR-INTERACT, we need an extra assumption on the outer and inner objective functions in Problem (1) due to the stochastic gradient estimator we use.

ASSUMPTION 3 (STOCHASTIC FUNCTIONS). *Assumptions 1 and 2 hold for $f_i(\mathbf{x}_i, \mathbf{y}_i; \xi)$ and $g_i(x_i, y_i; \xi)$, for all $\xi \in \mathrm{supp}\,(\pi)$, where $\mathrm{supp}(\pi)$ represents the support of π.*

Algorithm 2 The SVR-INTERACT Algorithm.

Set parameter pair $\forall i \in [m], (\mathbf{x}_{i,0}, \mathbf{y}_{i,0}) = (\mathbf{x}^0, \mathbf{y}^0)$.
At each agent $i \in [m]$, compute the local gradients:

$$\mathbf{u}_{i,0} = \bar{\nabla} f_i(\mathbf{x}_{i,0}, \mathbf{y}_{i,0}); \mathbf{v}_{i,0} = \nabla_{\mathbf{y}} g_i(\mathbf{x}_{i,0}, \mathbf{y}_{i,0}); \quad (26)$$

for $t = 1, \cdots, T$ **do**
  Update local parameters using (6) and (7);
  if $mod(t, q) = 0$ **then**
    Compute local gradients using (8) and (9);
  else
    Compute local gradients using (23) and (24);
  end if
  Track the gradient using (10);
end for

LEMMA 3. *Under Assumptions 1–2, $\forall i \in [m], (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$, the stochastic gradient estimator in (22) satisfies:*

$$\left\| \bar{\nabla} f_i(\mathbf{x}, \mathbf{y}) - \mathbb{E}_{\bar{\xi}_i}[\bar{\nabla} f(\mathbf{x}, \mathbf{y}; \bar{\xi}_i)] \right\| \leq \frac{C_{g_{xy}} C_{f_y}}{\mu_g} \left(1 - \frac{\mu_g}{L_g} \right)^K.$$

Besides, we have an additional lemma to characterizes the bias of the stochastic estimator shown in eqs.(22), and it also holds for the deterministic estimator shown in eqs.(5).

LEMMA 4. *Under Assumption 1-3, we have*

$$\mathbb{E}_{\bar{\xi}} \left\| \nabla f_i\left(\mathbf{x}_1, \mathbf{y}_1; \bar{\xi}\right) - \nabla f_i\left(\mathbf{x}_2, \mathbf{y}_2; \bar{\xi}\right) \right\|^2 \leq L_K^2 [\| \mathbf{x}_1 - \mathbf{x}_2 \|^2$$
$$+ \|\mathbf{y}_1 - \mathbf{y}_2\|^2], \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d_1}, \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^{d_2}. \quad (27)$$

In the above expressions, $\exists L_K \geq L_{K_s}$ and L_{K_s} is defined as:

$$L_{K_s}^2 = 2L_{f_x}^2 + 6C_{g_{xy}}^2 L_{f_y}^2 \left(\frac{K}{2\mu_g L_g - \mu_g^2} \right) + 6C_{f_y}^2 L_{g_{xy}}^2 \left(\frac{K}{2\mu_g L_g - \mu_g^2} \right)$$
$$+ 6C_{g_{xy}}^2 C_{f_y}^2 \frac{K^4}{L_g^2} \frac{1}{L_g^2} L_{g_{yy}}^2.$$

To establish the convergence of SVR-INTERACT, we use the same convergence metric as that of INTERACT as shown in (11).

1) Main Theorem of SVR-INTERACT: We state the main convergence rate result of SVR-INTERACT in the following theorem:

THEOREM 3. *(Convergence of SVR-INTERACT) Under Assumptions 1-3, if the step-sizes satisfy $\beta \leq \min\{ \frac{(1-\lambda)\mu_g L_g}{768 L_K^2 (\mu_g + L_g)}, \frac{(1-\lambda)(\mu_g + L_g)}{4096 L_K^2}, \frac{3(\mu_g + L_g)}{\mu_g L_g}, \frac{L_y^2 \mu_g L_g}{24 L_K^2 (\mu_g + L_g)}, \frac{(1-\lambda)(\mu_g + L_g)}{512 L_K^2}, \frac{1}{2(\mu_g + L_g)}, \frac{16}{(1-\lambda)(\mu_g + L_g)} \}, \alpha \leq \{ \frac{1}{8L_f}, \frac{r}{16 m L_y^2 (r+1)}, \frac{1}{8 L_K \sqrt{m}}, \frac{1}{m(1-\lambda)}, \frac{(1-\lambda)^2}{128 L_K^2}, \frac{(1-\lambda)}{4}(\frac{m}{L_f + 16 L_K^2 m}), \frac{1}{16 L_K}\sqrt{\frac{1-\lambda}{m}}, \frac{288 r(1+r) m L_y^2}{(1-\lambda) L_f^2}, \frac{r(1+r)(1-\lambda)}{256 L_y^2 (\mu_g + L_g) L_K^2 \beta}, \frac{r(1+r)(1-\lambda)^2}{512 L_y^2 (\mu_g + L_g)(L_K^2 \beta)}, \frac{\sqrt{1-\lambda}}{8 L_K}, \frac{(1-\lambda)^2}{4}, \frac{32 L_y^2}{16(1+\frac{1}{r})L_y^2}, \sqrt{\frac{1-\lambda}{64 L_K^2}}, \frac{32 L_y^2}{(1-\lambda)} \}, r = \frac{1}{3} \beta \frac{\mu_g L_g}{\mu_g + L_g}$, then the iterates $\{\mathbf{x}_t, \mathbf{y}_t\}$ generated by SVR-INTERACT satisfy*

$$\frac{1}{T+1} \sum_{t=0}^{T} \mathfrak{M}_t \leq \frac{\mathfrak{B}_0 - \ell^*}{(T+1) \min\{ \frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2} \}} + C_{\mathrm{bias}}$$

$$= O(\frac{1}{T+1}) + C_{\mathrm{bias}}, \quad (28)$$

where $\mathfrak{B}_t = \ell(\bar{\mathbf{x}}_t) + \|\bar{\mathbf{y}}_t - \mathbf{y}_t^*\|^2 + \|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 + \alpha \|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2$, $C_{\text{bias}} \triangleq \frac{2\alpha(\frac{C_{gxy}C_{fy}}{\mu_g}(1-\frac{\mu_g}{L_g})^K)^2}{\min\{\frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2}\}}$.

REMARK 2. From the statement of Theorem 3, it can be seen that the step-sizes α and β for SVR-INTERACT depend on the network topology, the Lipschitz constants, and the number of agents. Additionally, the convergence performance of SVR-INTERACT is affected by the constant C_{bias}, which is the bias term affected by the stochastic gradient estimator.

Theorem 3 immediately implies the following sample and communication complexity results of SVR-INTERACT:

COROLLARY 4. *Under the conditions of Theorem 3, to achieve an ϵ-stationary solution (cf. definition in Eq. (2)), with the number of samples for the gradient estimator in (22) chosen as $K = O(\log(1/\epsilon))$, SVR-INTERACT requires: 1) communication complexity: $O(\epsilon^{-1})$; and 2) sample complexity: $O(\sqrt{n}\epsilon^{-1})$).*

REMARK 3. Corollary 4 indicates that SVR-INTERACT has the same communication complexity as that of INTERACT, but improves the sample complexity of INTERACT.

5.3 Proof Sketch of the Convergence Results

The proof of Theorem 3 follows the same structure as that of Theorem 1, with Step 1)–Step 4) being identical. Thus, we omit the first four steps for brevity and only focus on the last step in this paper. In Algortihm 2, we can show the following relations: $\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\|^2 = 0$ every q iterations. Let n_t denotes the largest positive integer that satisfies $(n_t - 1)q \leq t$. With $t \in ((n_t - 1)q, n_tq - 1] \cap \mathbb{Z}$, we have $\left\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\right\|^2 \leq \frac{1}{m}\sum_{i=1}^{m}\|\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})\|^2$. We note that the next step is one of the most crucial steps in our proofs. The bias term has to be eliminated from the $\|\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})\|^2$ in advance; otherwise, we will not be able to use the mean variance theorem later. From the algorithm update of SVR-INTERACT, we have

$$\mathbb{E}_t\|\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})\|^2$$

$$\leq \mathbb{E}_t\|\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \mathbb{E}_{\bar{\xi}_i}[\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})]\|^2 + \left(\frac{C_{gxy}C_{fy}}{\mu_g}\left(1 - \frac{\mu_g}{L_g}\right)^K\right)^2.$$

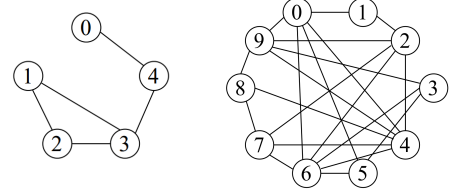Next, telescoping over t from $((n_t - 1)q + 1$ to t and noting $|\mathcal{S}| = q$ implies that

$$\mathbb{E}_t\|\mathbf{p}_i(\mathbf{x}_{i,(n_t-1)q}, \mathbf{y}_{i,(n_t-1)q}) - \mathbb{E}_{\bar{\xi}_i}[\mathbf{p}_i(\mathbf{x}_{i,(n_t-1)q}, \mathbf{y}_{i,(n_t-1)q})]\|^2 = 0,$$

Then, we have

$$\sum_{t=0}^{T}\|\frac{1}{m}\sum_{i=1}^{m}\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) - \bar{\mathbf{u}}_t\|^2 \leq \left[\frac{C_{gxy}C_{fy}}{\mu_g}\left(1 - \frac{\mu_g}{L_g}\right)^K\right]^2 (T+1)$$

$$+ L_K^2 \sum_{t=0}^{T}\mathbb{E}_t(\|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 + \|\mathbf{y}_t - \mathbf{y}_{t-1}\|^2).$$

Combing the above inequality and results in Steps 1)–4) and choosing $c_1 = c_2 = \frac{1}{\lambda} - 1$, we have:

$$\ell(\bar{\mathbf{x}}_{T+1}) - \ell(\bar{\mathbf{x}}_0) + \frac{1-\lambda}{32(1 + 1/r)L_y^2}\left[\|\bar{\mathbf{y}}_{T+1} - \mathbf{y}_{T+1}^*\|^2 - \|\mathbf{y}_0^* - \bar{\mathbf{y}}_0\|^2\right]$$

(a) A five-agent network. (b) A 10-agent network.

Figure 1: Different network topology.

$$+ \left[\|\mathbf{x}_{T+1} - \mathbf{1} \otimes \bar{\mathbf{x}}_{T+1}\|^2 - \|\mathbf{x}_0 - \mathbf{1} \otimes \bar{\mathbf{x}}_0\|^2\right] + \alpha\left[\|\mathbf{u}_{T+1} - \mathbf{1} \otimes \bar{\mathbf{u}}_{T+1}\|^2\right.$$

$$\left. - \|\mathbf{u}_0 - \mathbf{1} \otimes \bar{\mathbf{u}}_0\|^2\right] \leq -\frac{\alpha}{2}\sum_{t=0}^{T}\|\nabla\ell(\bar{\mathbf{x}}_t)\|^2 + C_1'\sum_{t=0}^{T}\|\bar{\mathbf{u}}_t\|^2$$

$$+ C_2'\sum_{t=0}^{T}\|\mathbf{x}_t - \mathbf{1} \otimes \bar{\mathbf{x}}_t\|^2 + C_3'\sum_{t=0}^{T}\|\mathbf{y}_t - \mathbf{y}_t^*\|^2 + C_4'\sum_{t=0}^{T}\|\mathbf{v}_t\|^2$$

$$+ C_5'\sum_{t=0}^{T}\|\mathbf{u}_t - \mathbf{1} \otimes \bar{\mathbf{u}}_t\|^2 + 2\alpha\left[\frac{C_{gxy}C_{fy}}{\mu_g}\left(1 - \frac{\mu_g}{L_g}\right)^K\right]^2(T+1), \quad (29)$$

where the definitions of the constants can be found in our technical report [23]. With appropriately chosen parameters to ensure $C_1', C_4', C_5' \leq 0, C_2' \leq -\frac{1-\lambda}{4}, C_3' \leq -\beta\frac{\mu_g L_g}{\mu_g + L_g}$, we have the following convergence results:

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathfrak{M}_t \leq \frac{\mathfrak{B}_0 - \ell^*}{(T+1)\min\{\frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2}\}} + C_{\text{bias}}, \quad (30)$$

where $C_{\text{bias}} \triangleq \frac{2\alpha(\frac{C_{gxy}C_{fy}}{\mu_g}(1-\frac{\mu_g}{L_g})^K)^2}{\min\{\frac{1-\lambda}{4}, \frac{3r^2(1-\lambda)}{32(1+r)L_y^2}, \frac{\alpha}{2}\}}$. This completes the proof.

6 EXPERIMENTAL EVALUATION

In this section, we conduct numerical experiments to demonstrate the performance of our INTERACT and SVR-INTERACT algorithms on a meta-learning problem. In particular, we evaluate and compare the performance of the proposed algorithms against two stochastic algorithms discussed below as the baselines:

- *Decentralized Gradient-Tracking Stochastic Gradient Descent (GT-DSGD):* This algorithm can be viewed as a "stripped-down" version of INTERACT utilizing stochastic gradients instead of full gradients. It is motivated by the GT-SGD algorithm [24]. The algorithm performs local updates using gradients $\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ and $\mathbf{d}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t})$ evaluated using the following stochastic gradient estimators at each agent: $\mathbf{p}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) = \frac{1}{\mathcal{S}}\sum_{i=1}^{\mathcal{S}}\left[\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i)\right]$, and $\mathbf{d}_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}) = \frac{1}{\mathcal{S}}\sum_{i=1}^{\mathcal{S}}\left[\nabla g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i)\right]$.
- *Decentralized Stochastic Gradient Descent (D-SGD):* This algorithm can be viewed as a simplified version of GT-SGD without utilizing gradient tracking [11]. For D-SGD, each agent updates its local parameters as: $\mathbf{x}_{i,t} = \sum_{j \in \mathcal{N}_i}[\mathbf{M}]_{ij}\mathbf{x}_{j,t-1} - \alpha\frac{1}{\mathcal{S}}\sum_{i=1}^{\mathcal{S}}\left[\bar{\nabla}f_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i)\right]$ and $\mathbf{y}_{i,t} = \mathbf{y}_{i,t-1} - \beta\frac{1}{\mathcal{S}}\sum_{i=1}^{\mathcal{S}}\left[\nabla g_i(\mathbf{x}_{i,t}, \mathbf{y}_{i,t}; \bar{\xi}_i)\right]$.

1) Meta-learning Model and Datasets: We evaluate the performance of INTERACT and SVR-INTERACT across m agents that collectively aim to solve the meta-learning problem with m tasks $\{\mathcal{T}_i, i \in [m]\}$. We allocate each agent one task with f_i defining
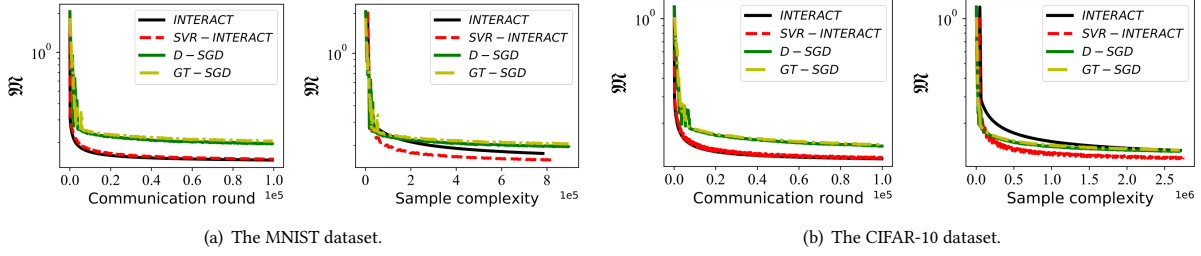
(a) The MNIST dataset.

(b) The CIFAR-10 dataset.

Figure 2: Convergence performance comparisons on the five-agent network.

(a)

(b)
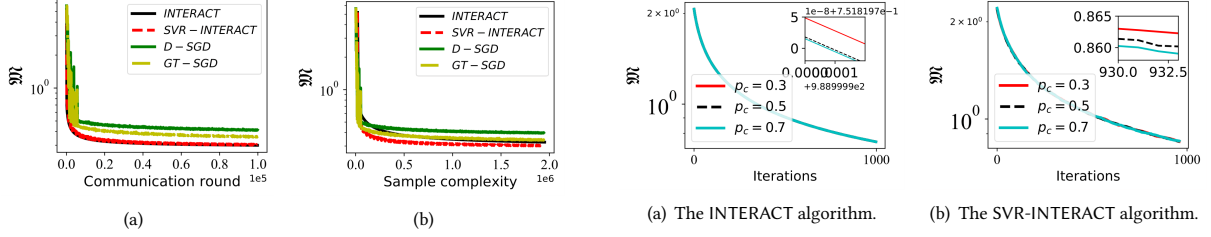
Figure 3: Convergence performance comparisons on the 10-agent network.

(a) The INTERACT algorithm.

(b) The SVR-INTERACT algorithm.

Figure 4: Convergence performance with different edge connectivity probability.

(a) The INTERACT algorithm.
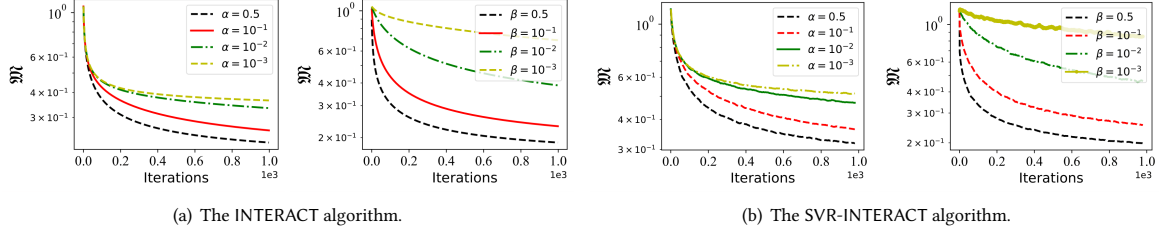
(b) The SVR-INTERACT algorithm.

Figure 5: Convergence performance with different learning rate on the MNIST dataset.

the loss function of the i-th agent. For the loss function $f_i(\mathbf{x}, \mathbf{y}_i)$ corresponding to task $\mathcal{T}_i$, the parameter x defines the model parameters shared among all agents, while y_i denotes task specific parameters. The meta-learning problem aims at finding common parameters $\mathbf{x}$ for all tasks, and each task then adapts its own parameters $\mathbf{y}_i$ by minimizing local lower level losses. The objective function is $\min \frac{1}{m} \sum_{i=1}^{m} f_i(\mathbf{x}, \mathbf{y}_i^*)$, subject to $\mathbf{y}_i^* = \arg\min_{\mathbf{y}_i} g_i(\mathbf{x}, \mathbf{y}_i)$. In practice, $\mathbf{y}_i$ denotes the parameters of the last layer of a neural network, while $\mathbf{x}$ are the parameters of the remaining layers. The inner problem includes a strongly convex regularizer ensuring that the inner function $g_i(\mathbf{x}, \mathbf{y}_i)$ is strongly-convex with respect to $\mathbf{y}_i$ and the outer function $f_i(\mathbf{x}, \mathbf{y}_i^*)$ is nonconvex with respect to $\mathbf{x}$.

We evaluate the proposed algorithms on the MNIST [18] and CIFAR-10 [16] datasets using a two-hidden-layer neural network with 20 hidden units. The network topology $\mathcal{G}$ is generated by the Erdös-Rènyi random graph approach. The consensus matrix is chosen as $\mathbf{W} = \mathbf{I} - \frac{2\mathbf{L}}{3\lambda_{\max}(\mathbf{L})}$, where $\mathbf{L}$ is the Laplacian matrix of $\mathcal{G}$ and $\lambda_{\max}(\mathbf{L})$ denotes the largest eigenvalue of $\mathbf{L}$.

2) Performance Comparison: We set the constant learning rates $\alpha = \beta = 0.5$, edge connectivity probability $p_c = 0.5$, number of agents $m = 5$, and each agent has $n = 1000$ data samples and mini-batch size $q = \lceil\sqrt{n}\rceil$. The network topology is shown in Fig. 1(a). In Fig. 2, we compare the performance of INTERACT

and SVR-INTERACT with the GT-SGD and SGD algorithms using the convergence metric $\mathfrak{M}$ on the MNIST and CIFAR-10 datasets. We note that both INTERACT and SVR-INTERACT outperform the baseline algoirthms GT-SGD and SGD. In Fig. 3, we notice a similar behavior when the number of tasks (and agents) are increased to 10. More importantly, we note that in all experiment settings, the SVR-INTERACT algorithm has the lowest sample complexity, which corroborates our theoretical findings.

3) Impact of the Edge Connectivity: We conduct experiments using five-node network systems and compare the performance of the algorithms on three different network topologies. For the three settings, we choose the value of the edge connectivity probability from the discrete set $\{0.3, 0.5, 0.7\}$, while the rest of the settings stay the same as previous experiments. As shown in Fig. 4, we can observe that the convergence metric $\mathfrak{M}$ is relatively insensitive to the change of network topologies. Fig. 4 shows that the convergence metric $\mathfrak{M}$ slightly increases as the edge connectivity probability p_c decreases (i.e., the network becomes sparser).

4) Impact of the Learning Rate: In this experiment, we choose the learning rates α and β from the discrete set $\{0.5, 0.1, 0.01, 0.001\}$. We fix the number of agents to five and set $p_c = 0.5$. The rest of the experiment settings stay the same as in previous experiments. As shown in Fig. 5, larger values of learning rates α or

β lead to faster convergence rates for both INTERACT and SVR-INTERACTalgorithms.

7 CONCLUSION

In this paper, we developed two algorithms called INTERACT and SVR-INTERACT for solving decentralized non-convex-strongly-convex bilevel optimization problems. We showed that, to achieve an ϵ-stationary point, INTERACT and SVR-INTERACT have the communication complexity of $O(\epsilon^{-1})$ and sample complexities of $O(n\epsilon^{-1})$ and $O(\sqrt{n}\epsilon^{-1})$, respectively. Our numerical studies corroborate the theoretical performance of our proposed algorithms. We note that our paper is the first to explore decentralized bilevel learning, which opens up several interesting directions for future research. For instance, it would be of interest to develop differentially-private algorithms for decentralized bilevel learning over networks. Also, one can adopt compression techniques to further reduce communication costs, especially for large-scale deep learning models.

ACKNOWLEDGMENTS

REFERENCES

[1] Baky, I. A. Fuzzy goal programming algorithm for solving decentralized bi-level multi-objective programming problems. *Fuzzy sets and systems 160*, 18 (2009), 2701–2713.
[2] Callaway, D. S., and Hiskens, I. A. Achieving controllability of electric loads. *Proceedings of the IEEE 99*, 1 (2010), 184–199.
[3] Chen, T., Sun, Y., and Yin, W. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671* (2021).
[4] Chen, T., Sun, Y., and Yin, W. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. *arXiv preprint arXiv:2106.13781* (2021).
[5] Ghadimi, S., Ruszczynski, A., and Wang, M. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization 30*, 1 (2020), 960–979.
[6] Ghadimi, S., and Wang, M. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246* (2018).
[7] Glavic, M., Fonteneau, R., and Ernst, D. Reinforcement learning for electric power system decision and control: Past considerations and perspectives. *IFAC-PapersOnLine 50*, 1 (2017), 6918–6927.
[8] Guo, Z., and Yang, T. Randomized stochastic variance-reduced methods for stochastic bilevel optimization. *arXiv e-prints* (2021), arXiv–2105.
[9] Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170* (2020).
[10] Ji, K., Yang, J., and Liang, Y. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning* (2021), PMLR, pp. 4882–4892.
[11] Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. Collaborative deep learning in fixed topology networks. *Advances in Neural Information Processing Systems 30* (2017).
[12] Jiang, Z., Mukherjee, K., and Sarkar, S. On consensus-disagreement tradeoff in distributed optimization. In *2018 Annual American Control Conference (ACC)* (2018), IEEE, pp. 571–576.
[13] Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A momentum-assisted single-timescale stochastic approximation algorithm for bilevel optimization, 2021.
[14] Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems 34* (2021).
[15] Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research 32*, 11 (2013), 1238–1274.
[16] Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images.
[17] Kunapuli, G., Bennett, K. P., Hu, J., and Pang, J.-S. Classification model selection via bilevel programming. *Optimization Methods & Software 23*, 4 (2008), 475–489.
[18] LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *Available: http://yann. lecun. com/exdb/mnist* (1998).

[19] Liu, M., Yuan, Z., Ying, Y., and Yang, T. Stochastic auc maximization with deep neural networks. *arXiv preprint arXiv:1908.10831* (2019).
[20] Liu, M., Zhang, W., Mroueh, Y., Cui, X., Ross, J., Yang, T., and Das, P. A decentralized parallel algorithm for training generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems* (2020), vol. 33.
[21] Liu, W., Mokhtari, A., Ozdaglar, A., Pattathil, S., Shen, Z., and Zheng, N. A decentralized proximal point-type method for saddle point problems. *arXiv preprint arXiv:1910.14380* (2019).
[22] Liu, Y., and Liu, R. Boml: A modularized bilevel optimization library in python for meta learning. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (2021), IEEE, pp. 1–2.
[23] Liu, Z., Zhang, X., Khanduri, P., Lu, S., and Liu, J. INTERACT: achieving low sample and communication complexities in decentralized bilevel learning over networks. https://kevinliu-osu.github.io/publications/INTERACT_TR.pdf.
[24] Lu, S., Zhang, X., Sun, H., and Hong, M. Gnsd: A gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *2019 IEEE Data Science Workshop (DSW)* (2019), IEEE, pp. 315–321.
[25] MacKay, M., Vicol, P., Lorraine, J., Duvenaud, D., and Grosse, R. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. *arXiv preprint arXiv:1903.03088* (2019).
[26] Mateos-Núñez, D., and Cortés, J. Distributed subgradient methods for saddle-point problems. In *2015 54th IEEE Conference on Decision and Control (CDC)* (2015), IEEE, pp. 5462–5467.
[27] Mehra, A., and Hamm, J. Penalty method for inversion-free deep bilevel optimization. In *Asian Conference on Machine Learning* (2021), PMLR, pp. 347–362.
[28] Moore, G. M. *Bilevel programming algorithms for machine learning model selection.* Rensselaer Polytechnic Institute, 2010.
[29] Nedic, A., and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control 54*, 1 (2009), 48–61.
[30] Okuno, T., Takeda, A., Kawana, A., and Watanabe, M. On lp-hyperparameter learning via bilevel nonsmooth optimization. *Journal of Machine Learning Research 22*, 245 (2021), 1–47.
[31] Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International conference on machine learning* (2016), PMLR, pp. 737–746.
[32] Polydoros, A. S., and Nalpantidis, L. Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent & Robotic Systems 86*, 2 (2017), 153–173.
[33] Qi, Q., Luo, Y., Xu, Z., Ji, S., and Yang, T. Stochastic optimization of areas under precision-recall curves with provable convergence. *Advances in Neural Information Processing Systems 34* (2021).
[34] Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. *Advances in neural information processing systems 32* (2019).
[35] Rhee, S. H., Kim, H.-S., and Sohn, S.-W. The effect of decentralized resource allocation in network-centric warfare. In *The International Conference on Information Network 2012* (2012), IEEE, pp. 478–481.
[36] Shi, C., Lu, J., and Zhang, G. An extended kuhn–tucker approach for linear bilevel programming. *Applied Mathematics and Computation 162*, 1 (2005), 51–63.
[37] Sun, H., Lu, S., and Hong, M. Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking. In *Proceedings of International Conference on Machine Learning* (2020), PMLR, pp. 9217–9228.
[38] Toksari, M. D., and Bilim, Y. Interactive fuzzy goal programming based on jacobian matrix to solve decentralized bi-level multi-objective fractional programming problems. *International Journal of Fuzzy Systems 17*, 4 (2015), 499–508.
[39] Yang, J., Ji, K., and Liang, Y. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems 34* (2021).
[40] Yau, C.-Y., and Wai, H.-T. Docom-sgt: Doubly compressed momentum-assisted stochastic gradient tracking algorithm for communication efficient decentralized learning. *arXiv preprint arXiv:2202.00255* (2022).
[41] Zhang, H., Chen, W., Huang, Z., Li, M., Yang, Y., Zhang, W., and Wang, J. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 7325–7332.
[42] Zhang, X., Liu, J., Zhu, Z., and Bentley, E. S. Gt-storm: taming sample, communication, and memory complexities in decentralized non-convex learning. In *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing* (2021), pp. 271–280.
[43] Zhang, X., Liu, Z., Liu, J., Zhu, Z., and Lu, S. Taming communication and sample complexities in decentralized policy evaluation for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems 34* (2021).