A Stochastic Polyhedral Approximation Method for Composite Decentralized Bilevel Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Bilevel optimization has received increasing attention in many machine learning problems, e.g., meta-learning, reinforcement learning, and hyperparameter optimization. Existing studies on bilevel optimization are primarily geared toward centralized and smooth settings. In practice, the data and learning tasks may be located at different computing nodes. Besides, nonsmooth terms in objectives arise naturally in machine learning applications that involve regularization or penalty. In this paper, we propose a **S**tochastic **P**olyhedral **A**pproximation **M**ethod (SPAM) for composite decentralized bilevel optimization problems. The proposed SPAM allows networked agents to solve bilevel optimization problems in which both upper-level and lower-level objective functions are nonconvex and contain nonsmooth terms in a fully decentralized manner. We also establish that the proposed SPAM can achieve an $\mathcal{O}(1/\epsilon)$ convergence rate. Numerical experiments on public datasets corroborate the effectiveness and efficiency of the proposed algorithm.

1 Introduction

In recent years, bilevel optimization has gained increasing interests, with a variety of applications found in machine learning problems, such as meta-learning Ji et al. [2020], Likhosherstov et al. [2021], reinforcement learning Hong et al. [2020], Chen et al. [2019], neural architecture search Jiang et al. [2020], Jiao et al. [2022a], and hyperparameter optimization Khanduri et al. [2021], Liu et al. [2021]. In bilevel optimization, two levels of the minimization subproblems are nested with each other. From the perspective of classical optimization, bilevel optimization can be considered as a special case of constrained optimization since the lower-level (LL) optimization problem can be viewed as a constraint to the upper-level (UL) optimization problem Sinha et al. [2017].

The majority of existing bilevel optimization works focus on algorithm designs in the classic centralized setting, requiring passing large amounts of data to a centralized server. These single-agent methods may pose privacy risks Subramanya and Riggio [2021]. Besides, the dramatically increasing model sizes and the large amount of data generated distributedly by the ubiquitous Internet of Things (IoT) devices also set requirements of using multiple computational resources Lu et al. [2019]. Since using a parameter server architecture Li et al. [2013] which consists of a server and several clients still encounters communication bottlenecks, there is a great demand for decentralized bilevel optimization (DBO) methods. Although the research on DBO has made progress recently Liu et al. [2022], Lu et al. [2022], Yang et al. [2022], Qiu et al. [2022], Chen et al. [2022], designing effective and efficient algorithms for solving composite decentralized bilevel optimization (CDBO) problems remains underexplored. In practice, a wide range of problems in the fields of machine learning and multiagent system can fall into composite optimization form Wang et al. [2020]. For-

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

mally, the CDBO problem can be expressed as

$$\begin{aligned} &\min \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i, \mathbf{y}_i) \triangleq G_i(\mathbf{x}_i, \mathbf{y}_i) + R(\mathbf{x}_i) \\ &\text{s.t. } \mathbf{y}_i = \arg\min_{\mathbf{y}_i'} \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}_i, \mathbf{y}_i') \triangleq g_i(\mathbf{x}_i, \mathbf{y}_i') + r(\mathbf{y}_i'), \forall i \in [N] \\ &\text{var. } \{\mathbf{x}_i\}, \{\mathbf{y}_i\}, \end{aligned} \tag{1}$$

where N is the number of agents over a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{E}$ and $\mathcal{V}$ represent the edges and vertices. Each agent can only communicate with its neighbors. F_i and f_i denote the UL and LL objective functions of agent i, and each of them is the summation of a (nonconvex) differentiable function and a nonsmooth convex regularizer Jiang et al. [2022]. Vector $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^m$ are the UL and LL optimization variables of agent i, respectively.

Major Contributions. In this paper, we focus on the nonconvex composite decentralized bilevel optimization (CDBO) problem, where objective functions of both levels are nonconvex and contain nonsmooth terms. Besides, the optimization problems of both levels include consensus constraints. By regarding the LL optimization problem as a constraint of the UL optimization problem, we develop a single-loop algorithm named **S**tochastic **P**olyhedral **A**pproximation **M**ethod (SPAM). In the proposed SPAM, each agent utilizes cutting planes Yang et al. [2014] to approximate the feasible set of the reformulated single-level problem collaboratively based on stochastic data and local communication. Besides, proximal operators Shi et al. [2015] are adopted to cope with nonsmooth terms, and the gradient tracking mechanism Di Lorenzo and Scutari [2016], Koloskova et al. [2021] is employed to accelerate the convergence through better information sharing Li et al. [2020]. Moreover, we prove that SPAM is guaranteed to converge even if both the UL and LL objectives are nonconvex.

Our main contributions are summarized below:

1. We propose a novel algorithm named **S**tochastic **P**olyhedral **A**pproximation **M**ethod (SPAM), which is single-loop and computationally efficient. To the best of our knowledge, this is the first stochastic method for nonconvex composite decentralized bilevel optimization.

2. We establish convergence guarantees of SPAM with nonconvex nonsmooth objectives under consensus constraints of both UL and LL problems. The theoretical results shows that SPAM enjoys a convergence rate of $\mathcal{O}(1/\epsilon)$ to achieve an ϵ-stationary point.

3. To examine the performance of the proposed SPAM algorithms, we conduct experiments on meta-learning and hyperparameter optimization tasks with multiple public datasets. Our results show that SPAM outperforms other baseline algorithms.

2 Related Works

Bilevel optimization is defined as a hierarchical mathematical program. A kind of classical approaches for solving the bilevel optimization problem is to reformulate it as a single-level constrained optimization problem by replacing the LL optimization problem with its analytical solution Zhang et al. [2022] or KKT conditions Biswas and Hoyle [2019]. However, analytical solutions are often difficult to obtain and the KKT replacement could lead to a mass of conditions. Motivated by machine learning applications, some approaches use gradient descent based on the hypergradient Liao et al. [2018] to solve the bilevel optimization problem. These approaches can be further categorized into the approximate implicit differentiation (AID) based approaches Lorraine et al. [2020] and the iterative differentiation (ITD) based approaches Grazzi et al. [2020]. Recently, with the rapid growth of data volume, stochastic bilevel optimization Ghadimi and Wang [2018], Yang et al. [2021] has been proposed to achieve better efficiency than deterministic approaches. Most of the existing approaches for bilevel optimization focus on centralized settings. These approaches may face data privacy risks Subramanya and Riggio [2021] since they require collecting data from large amount of distributed devices. Some (centralized) distributed bilevel optimization Ji et al. [2021], Jiao et al. [2023] methods apply the parameter-server architecture to avoid the data privacy risks, but the communication bottlenecks of the server hinders the practical application of these approaches. Therefore, decentralized bilevel optimization Liu et al. [2022], Yang et al. [2022], Qiu et al. [2022] has attracted more attention recently.

A decentralized framework refers to serverless architectures where each agent can only communicate with its neighbors Sun et al. [2021]. The main challenge of the decentralized learning is the data heterogeneity across agents Chen et al. [2022], which could be mitigated by consensus communication strategies Zhao and Song [2018]. In the literature, gradient tracking strategies Di Lorenzo and Scutari [2016], Pu and Nedić [2021], Koloskova et al. [2021] have been applied to improve the convergence rate of decentralized optimization. There have been recent works considering bilevel optimization under decentralized setting. Liu et al. [2022] and Chen et al. [2022] proposed deterministic and stochastic algorithms for the decentralized bilevel optimization (DBO) problem with nonconvex UL optimization problems and strongly-convex LL optimization problems. The stochastic algorithm developed in Lu et al. [2022] and Yang et al. [2022] can solve DBO problem where both UL and LL optimization problems are nonconvex. However, all of these works are designed for smooth objectives. None of the existing work tackles the nonconvex and nonsmooth composite decentralized bilevel optimization (CDBO) problem.

Polyhedral approximation methods have been shown to be powerful for approximating complex optimization problems. The main concept behind polyhedral approximation-based algorithms is to construct a polyhedral outer approximation of the feasible set, and use the approximation to form a linear relaxation of the target problem Lundell and Kronqvist [2022]. Cutting plane Yang et al. [2014] (or outer linearization) is one of the classical polyhedral approximation methods. Jiao et al. [2022b] and Jiao et al. [2023] have verified the effectiveness of cutting plane methods on bilevel optimization, but they focus on smooth objectives and parameter-server architecture. In addition, these deterministic algorithms require all the data to update cutting planes during the iteration, which is inefficient and difficult to realize in practice.

3 The SPAM Algorithm

Problem formulation. We first present the procedure for approximating the problem in Eq. (1). A consensus version Liu et al. [2022] of Eq. (1) is expressed as

$$\begin{aligned} \min_{\{\mathbf{x}_i\}} \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i,\mathbf{y}_i) &\triangleq G_i(\mathbf{x}_i,\mathbf{y}_i) + R(\mathbf{x}_i) \\ \text{s.t. } \mathbf{x}_i &= \mathbf{x}_j, \forall i \in [N], j \in \mathcal{N}_i \\ \{\mathbf{y}_i\} = \arg\min_{\{\mathbf{y}_i'\}} \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}_i,\mathbf{y}_i') &\triangleq g_i(\mathbf{x}_i,\mathbf{y}_i') + r(\mathbf{y}_i') \\ \text{s.t. } \mathbf{y}_i' &= \mathbf{y}_j', \forall i \in [N], j \in \mathcal{N}_i, \end{aligned} \tag{2}$$

where G_i and g_i are smooth, possibly nonconvex function known only to agent i; R and r are convex, possibly nonsmooth function common to all agents. $F_i = G_i + R$ and $f_i = g_i + r$ denote the local UL and LL objective functions. G_i and g_i can be written as $G_i \triangleq \mathbb{E}_{\xi_i}[G_i(\mathbf{x}_i,\mathbf{y}_i;\xi_i)]$ and $g_i \triangleq \mathbb{E}_{\zeta_i}[g_i(\mathbf{x}_i,\mathbf{y}_i;\zeta_i)]$, where ξ_i and ζ_i denote the data sampled from heterogeneous distributions across agents for UL and LL problems. $\mathcal{N}_i$ denotes the set of neighboring agents for agent i. The equality constraint $\mathbf{x}_i = \mathbf{x}_j$ and $\mathbf{y}_i' = \mathbf{y}_j'$, $\forall i \in [N]$, $j \in \mathcal{N}_i$ enforce the model agreements ("consensus") at each level of the problems Lu et al. [2022]. In each iteration, every agent shares and receives information from its neighbors, and aggregates the neighbor information through a consensus weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ of the communication graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$.

Next, we define that $\phi(\{\mathbf{x}_i\}) = \arg\min_{\{\mathbf{y}_i'\}}\{\frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}_i,\mathbf{y}_i') : \mathbf{y}_i' = \mathbf{y}_j', \forall i \in [N], j \in \mathcal{N}_i\}$ and $h(\{\mathbf{x}_i\},\{\mathbf{y}_i\}) = \|\{\mathbf{y}_i\} - \phi(\{\mathbf{x}_i\})\|_1 + \lambda_1 \|\{\mathbf{y}_i\} - \phi(\{\mathbf{x}_i\})\|_2^2$, where λ_1 is a weight coefficient. It has been proved that the combination of l_1 norm l_2 norm leads to better robustness and efficiency Gokcesu and Gokcesu [2021]. Then we can reformulate the decentralized bilevel optimization problem in Eq. (2) as the following single-level optimization problem with constraints:

$$\begin{aligned} \min_{\{\mathbf{x}_i\},\{\mathbf{y}_i\}} \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i,\mathbf{y}_i) &\triangleq G_i(\mathbf{x}_i,\mathbf{y}_i) + R(\mathbf{x}_i) \\ s.t.\ \mathbf{x}_i &= \mathbf{x}_j, \forall i \in [N], j \in \mathcal{N}_i, \\ h(\{\mathbf{x}_i\},\{\mathbf{y}_i\}) &= 0. \end{aligned} \tag{3}$$

LL Optimization Problem Estimation. Many previous works Gould et al. [2016], Li et al. [2022] have pointed out that the LL problem does not require an exact solution. Ji et al. [2021], Yang et al. [2021] approximate the optimal solution of LL optimization problem by K steps of gradient descent, and Jiao et al. [2023] utilizes the results after K communication rounds between the master and workers. To estimate $\phi(\{\mathbf{x}_i\})$, we first obtain the first-order Taylor approximation of $g_i(\mathbf{x}_i, \mathbf{y}'_i)$ with respect to $\mathbf{x}_i$. That is, for a given point $\bar{\mathbf{x}}_i$, $\tilde{g}_i(\mathbf{x}_i, \mathbf{y}'_i) = g_i(\bar{\mathbf{x}}_i, \mathbf{y}'_i) + \nabla_{\mathbf{x}_i} g_i(\bar{\mathbf{x}}_i, \mathbf{y}'_i)^\top (\mathbf{x}_i - \bar{\mathbf{x}}_i)$, $\tilde{f}_i(\mathbf{x}_i, \mathbf{y}'_i) = \tilde{g}_i(\mathbf{x}_i, \mathbf{y}'_i) + r(\mathbf{y}'_i)$. Then, given $\tilde{f}_i(\mathbf{x}_i, \mathbf{y}'_i)$, the augmented Lagrangian function of the LL optimization problem in Eq. (2) is

$$g_p(\{\mathbf{x}_i\}, \{\mathbf{y}'_i\}, \{\boldsymbol{\varphi}_{ij}\}) = \frac{1}{N} \sum_{i=1}^{N} \left(\tilde{f}_i(\mathbf{x}_i, \mathbf{y}'_i) + \sum_{j \in \mathcal{N}_i} \left(\boldsymbol{\varphi}_{ij}^\top (\mathbf{y}'_i - \mathbf{y}'_j) + \frac{\mu}{2} ||\mathbf{y}_i - \mathbf{y}'_j||_2^2 \right) \right), \quad (4)$$

where $\boldsymbol{\varphi}_{i,j} \in \mathbb{R}^m$ is the dual variable, and $\mu > 0$ is a penalty parameter. Inspired by Shi et al. [2015], we use a variant of Eq. (4) as follows:

$$g'_p(\{\mathbf{x}_i\}, \{\mathbf{y}'_i\}, \{\boldsymbol{\varphi}_{ij}\}) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(\tilde{g}_i(\mathbf{x}_i, \mathbf{y}'_i) + \sum\nolimits_{j \in \mathcal{N}_i} \left(\boldsymbol{\varphi}_{ij}^\top (\mathbf{y}'_i - \mathbf{y}'_j) + \frac{\mu}{2} ||\mathbf{y}'_i - \mathbf{y}'_j||_2^2 \right) \right). \quad (5)$$

Finally, we use K communication rounds among agents to approximate $\phi(\{\mathbf{x}_i\})$, which contains the following steps in the $(k+1)^{th}$ iteration:

(1) Communicate with neighbors and update the $\mathbf{y}-$ variables by proximal mappings:

$$\mathbf{y}'_{i,k+1} = \arg\min_{\mathbf{y}_i} \{\eta_y r(\mathbf{y}_i) + \frac{1}{2} \|\mathbf{y}_i - (\sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{y}'_{j,k} - \eta_y \mathbf{p}_{i,k})\|^2\}, \quad (6)$$

where η_y is the step-size, $\mathbf{p}_{i,k}$ is an auxiliary vector for gradient tracking.

(2) Update $\boldsymbol{\varphi}-$ variables:

$$\boldsymbol{\varphi}_{ij,k+1} = \boldsymbol{\varphi}_{ij,k} + \eta_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\varphi}_{ij}} g'_p(\{\mathbf{x}_i\}, \{\mathbf{y}'_{i,k+1}\}, \{\boldsymbol{\varphi}_{ij,k}\}; \zeta_{i,k+1}), \quad (7)$$

where $\eta_{\boldsymbol{\varphi}}$ is the step-size.

(3) Update the gradient tracking variables:

$$\begin{aligned} \mathbf{p}_{i,k+1} = &\sum_{j \in \mathcal{N}_i} \mathbf{W}_{ij} \mathbf{p}_{i,k} + \nabla_{\mathbf{y}_i} g'_p(\{\mathbf{x}_i\}, \{\mathbf{y}'_{i,k+1}\}, \{\boldsymbol{\varphi}_{ij,k+1}\}; \zeta_{i,k+1}) \\ &- \nabla_{\mathbf{y}_i} g'_p(\{\mathbf{x}_i\}, \{\mathbf{y}'_{i,k}\}, \{\boldsymbol{\varphi}_{ij,k}\}; \zeta_{i,k}). \end{aligned} \quad (8)$$

Finally, the results after K communication rounds are utilized to approximate $\phi(\{\mathbf{x}_i\})$:

$$\phi(\{\mathbf{x}_i\}) = \{\mathbf{y}'_{i,K}\}. \quad (9)$$

Polyhedral Approximation. A relaxed problem of Eq.(3) can be expressed as:

$$\begin{aligned} \min\ & \frac{1}{N} \sum_{i=1}^{N} F_i(\mathbf{x}_i, \mathbf{y}_i) \triangleq G_i(\mathbf{x}_i, \mathbf{y}_i) + R(\mathbf{x}_i) \\ \text{s.t. } & \mathbf{x}_i = \mathbf{x}_j, \forall i \in [N], j \in \mathcal{N}_i, \\ & h(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) \leq \varepsilon \\ \text{var. } & \{\mathbf{x}_i\}, \{\mathbf{y}_i\}, \end{aligned} \quad (10)$$

where $\varepsilon > 0$ is a constant. Assuming that $h(\{\mathbf{x}_i\}, \{\mathbf{y}_i\})$ is convex w.r.t. $(\{\mathbf{x}_i\}, \{\mathbf{y}_i\})$, which is always satisfied when we set $r(\mathbf{y}_i) = ||\mathbf{y}_i||_1$ in Eq.(2) and $K = 1$ in Eq.(9) according to the soft-thresholding operator Shi et al. [2015] and operations that preserve convexity Boyd et al. [2004]. Then the feasible set with respect to constraint $h(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) \leq \varepsilon$ is a convex set Jiao et al. [2023]. In order to approximate the feasible region with respect to constraint $h(\{\mathbf{x}_i\}, \{\mathbf{y}_i\}) \leq \varepsilon$ in Eq. (10), we utilize the cutting plane method Boyd and Vandenberghe [2007], Franc et al. [2011], Yang et al. [2014], which has been proved to be effective and efficient in nonconvex optimization Michalka [2013], Vieira and Lisboa [2019], Jiao et al. [2023]. Denoting the parameters of l^{th} cutting plane as $\boldsymbol{a}_{i,l} \in \mathbb{R}^n, \boldsymbol{b}_{i,l} \in \mathbb{R}^m$ and $\kappa_l \in \mathbb{R}^1$, the polytope $\boldsymbol{\mathcal{P}}^t$ formed by cutting planes in the $(t+1)^{th}$ iteration is

$$\boldsymbol{\mathcal{P}}^t = \left\{ \sum_{i=1}^{N} \boldsymbol{a}_{i,l}^\top \mathbf{x}_i + \sum_{i=1}^{N} \boldsymbol{b}_{i,l}^\top \mathbf{y}_i + \kappa_l \leq 0, \forall l \in [|\boldsymbol{\mathcal{P}}^t|] \right\}, \quad (11)$$

where $|\boldsymbol{\mathcal{P}}^t|$ is the number of cutting planes in $\boldsymbol{\mathcal{P}}^t$. Then we have an approximate problem as:

$$\begin{aligned} &\min \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i,\mathbf{y}_i) \triangleq G_i(\mathbf{x}_i,\mathbf{y}_i) + R_i(\mathbf{x}_i) \\ &\text{s.t. } \mathbf{x}_i = \mathbf{x}_j, \forall i \in [N], j \in \mathcal{N}_i \\ &\quad \sum_{i=1}^{N} \boldsymbol{a}_{i,l}^\top \mathbf{x}_i + \sum_{i=1}^{N} \boldsymbol{b}_{i,l}^\top \mathbf{y}_i + \kappa_l \le 0, \forall l \in [|\boldsymbol{\mathcal{P}}^t|] \\ &\text{var. } \{\mathbf{x}_i\}, \{\mathbf{y}_i\}. \end{aligned} \tag{12}$$

In the decentralized setting, each agent maintains a subset of cutting planes in $\boldsymbol{\mathcal{P}}^t$, denoted as $\boldsymbol{\mathcal{P}}_i^t$, $\sum_{i=1}^{N} |\boldsymbol{\mathcal{P}}_i^t| = |\boldsymbol{\mathcal{P}}^t|$. We set that $|\boldsymbol{\mathcal{P}_i}^t| < M, \forall t$. In addition, only variables of neighbors are observable for agent i's cutting planes. That is, $\boldsymbol{\mathcal{P}}_i^t = \{\sum_{j\in\mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j\in\mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l \le 0, \forall l \in [|\boldsymbol{\mathcal{P}}_i^t|]\}$. Therefore, Eq.(12) is replaced by the following formulation:

$$\begin{aligned} &\min \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i,\mathbf{y}_i) \triangleq G_i(\mathbf{x}_i,\mathbf{y}_i) + R(\mathbf{x}_i) \\ &\text{s.t. } \mathbf{x}_i = \mathbf{x}_j, \forall i \in [N], j \in \mathcal{N}_i \\ &\quad \sum_{i=1}^{N} (\sum_{j\in\mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j\in\mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l \le 0), \forall i \in [N], j \in \mathcal{N}_i, \forall l \in [|\boldsymbol{\mathcal{P}}_i^t|] \\ &\text{var. } \{\mathbf{x}_i\}, \{\mathbf{y}_i\}. \end{aligned} \tag{13}$$

Algorithm Design. The Lagrangian function of Eq. (13) can be written as:

$$L_p = \frac{1}{N}\sum_{i=1}^{N} F_i(\mathbf{x}_i,\mathbf{y}_i) + \sum_{i=1}^{N}\sum_{l=1}^{|\boldsymbol{\mathcal{P}}_i^t|} \lambda_{i,l} \left(\sum_{j\in\mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j\in\mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l \right) + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_{i,j}^\top (\mathbf{x}_i - \mathbf{x}_j), \tag{14}$$

where L_p is simplified form of $L_p(\{\mathbf{x}_i\},\{\mathbf{y}_i\},\{\lambda_{i,l}\},\{\boldsymbol{\theta}_{i,j}\})$, $\lambda_{i,l} \in \mathbb{R}^1$ and $\boldsymbol{\theta}_{i,j} \in \mathbb{R}^n$ are dual variables. When replace the F_i in Eq.(14) by G_i, we have

$$L_p' = \frac{1}{N}\sum_{i=1}^{N} G_i(\mathbf{x}_i,\mathbf{y}_i) + \sum_{i=1}^{N}\sum_{l=1}^{|\boldsymbol{\mathcal{P}}_i^t|} \lambda_{i,l} \left(\sum_{j\in\mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j\in\mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l \right) + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}_{i,j}^\top (\mathbf{x}_i - \mathbf{x}_j). \tag{15}$$

Then, following Xu et al. [2023], a regularized version of Eq.(15) can be expressed as:

$$\tilde{L}_p = L_p' - \sum_{i=1}^{N}\sum_{l=1}^{|\boldsymbol{\mathcal{P}}_i^t|} \frac{c_1^t}{2}||\lambda_{i,l}||^2 - \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \frac{c_2^t}{2}||\boldsymbol{\theta}_{i,j}||^2 = \sum_{i=1}^{N} \tilde{L}_{p\_i}(\{\mathbf{x}_i\},\{\mathbf{y}_i\},\{\lambda_{i,l}\},\{\boldsymbol{\theta}_{i,j}\}), \tag{16}$$

where $c_1^t = \frac{1}{\eta_\lambda (t+1)^{\frac{1}{2}}} \ge \underline{c}_1$ and $c_2^t = \frac{1}{\eta_\theta (t+1)^{\frac{1}{2}}} \ge \underline{c}_2$ are nonnegative non-increasing regularization sequences. We set that $\underline{c}_1 \ge (8NML(1-\rho)^3/\eta_\lambda)^{\frac{1}{2}}$, $\underline{c}_2 \ge (8N^2L(1-\rho)^3/\eta_\theta)^{\frac{1}{2}}$.

The proposed SPAM algorithm contains the following steps in the $(t+1)^{th}$ iteration:

(1) Consensus update with proximal gradient descent. Agents communicate with neighbors and update the $\mathbf{x}-$ variables and $\mathbf{y}-$ variables by using proximal mappings as follows,

$$\begin{cases} \mathbf{d}_i^t = \sum\limits_{j\in\mathcal{N}_i^{t+1}} \mathbf{W}_{ij}\mathbf{x}_j^t \\ \mathbf{x}_i^{t+1} = \arg\min\limits_{\mathbf{x}_i} \left\{ \eta_x R(\mathbf{x}_i) + \frac{1}{2}\left\|\mathbf{x}_i - \left(\mathbf{d}_i^t - \eta_x \mathbf{e}_i^t\right)\right\|^2 \right\} \end{cases}, \tag{17}$$

$$\begin{cases} \mathbf{u}_i^t = \sum\limits_{j\in\mathcal{N}_i^{t+1}} \mathbf{W}_{ij}\mathbf{y}_j^t \\ \mathbf{y}_i^{t+1} = \mathbf{u}_i^t - \eta_{\mathbf{y}} \mathbf{v}_i^t \end{cases}, \tag{18}$$

where η_x and η_y are step-sizes. $\mathbf{e}_i^k$ and $\mathbf{v}_i^k$ are gradient tracking vectors for $\mathbf{x}-$ and $\mathbf{y}-$ variables.

(2) Local update of dual variables. Agents update dual variables as follows,

$$\lambda_{i,l}^{t+1} = \lambda_{i,l}^t + \eta_\lambda \nabla_{\lambda_{i,l}} \tilde{L}_{p\_i}(\{\mathbf{x}_i^{t+1}\}, \{\mathbf{y}_i^{t+1}\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}; \xi_i^{t+1}), \tag{19}$$

$$\boldsymbol{\theta}_{i,j}^{t+1} = \boldsymbol{\theta}_{i,j}^t + \eta_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_{i,j}} \tilde{L}_{p\_i}(\{\mathbf{x}_i^{t+1}\}, \{\mathbf{y}_i^{t+1}\}, \{\lambda_{i,l}^{t+1}\}, \{\boldsymbol{\theta}_{i,j}^t\}; \xi_i^{t+1}), \tag{20}$$

where η_λ and $\eta_{\boldsymbol{\theta}}$ are step-sizes.

(3) Communicate and update gradient tracking variables. Agents updates the gradient tracking variable via communication as follows,

$$\begin{cases} \mathbf{e}_i^{t+1} = \sum\limits_{j \in \mathcal{N}_i^{t+1}} \mathbf{W}_{ij} \mathbf{e}_j^t + \nabla_{\mathbf{x}_i} \tilde{L}_{p\_i}(\{\mathbf{x}_i^{t+1}\}, \{\mathbf{y}_i^{t+1}\}, \{\lambda_{i,l}^{t+1}\}, \{\boldsymbol{\theta}_{i,j}^{t+1}\}; \xi_i^{t+1}) \\ \qquad - \nabla_{\mathbf{x}_i} \tilde{L}_{p\_i}(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}; \xi_i^t) \\ \mathbf{v}_i^{t+1} = \sum\limits_{j \in \mathcal{N}_i^{t+1}} \mathbf{W}_{ij} \boldsymbol{v}_j^t + \nabla_{\mathbf{y}_i} \tilde{L}_{p\_i}(\{\mathbf{x}_i^{t+1}\}, \{\mathbf{y}_i^{t+1}\}, \{\lambda_{i,l}^{t+1}\}, \{\boldsymbol{\theta}_{i,j}^{t+1}\}; \xi_i^{t+1}) \\ \qquad - \nabla_{\mathbf{y}_i} \tilde{L}_{p\_i}(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}; \xi_i^t) \end{cases}. \tag{21}$$

Updating Cutting Planes When $t < T_1$, the cutting planes will be updated every k_{pre} iterations. The update of cutting planes contains two major step:

(1) Removing the inactive cutting planes.

$$\boldsymbol{\mathcal{P}}_i^{t+1} = \begin{cases} \text{Drop}(\boldsymbol{\mathcal{P}}_i^t, cp_l), \text{if } \lambda_{i,l}^{t+1} \text{and } \lambda_{i,l}^t = 0 \\ \boldsymbol{\mathcal{P}}_i^t, \text{otherwise} \end{cases}, \tag{22}$$

$$\{\lambda_{i,l}^{t+1}\} = \begin{cases} \text{Drop}(\{\lambda_{i,l}^t\}, \lambda_{i,l}), \text{if} \quad \lambda_{i,l}^{t+1} \text{and} \lambda_{i,l}^t = 0 \\ \{\lambda_{i,l}^t\}, \text{otherwise} \end{cases}, \tag{23}$$

where $\text{Drop}(\boldsymbol{\mathcal{P}}_i^t, cp_l)$ represents the l^{th} cutting plane cp_l is removed from $\boldsymbol{\mathcal{P}}_i^t$, and $\text{Drop}(\{\lambda_{i,l}^t\}, \lambda_{i,l})$ represents that the $\lambda_{i,l}$ is removed from the dual variable set $\{\lambda_{i,l}^t\}$.

(2) Adding new cutting planes. Since only variables of neighbors are observable at each agent, we define a local estimation of $h(\{\mathbf{x}_i\}, \{\mathbf{y}_i\})$, denoted as $\bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}) = h(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}), j \in \mathcal{N}_i$. Then we investigate whether $(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})$ is feasible for the constraint $\bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}) \leq \varepsilon$. We can obtain $\bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})$ according to $\phi(\{\mathbf{x}_j^{t+1}\})$ in Eq.(9). If $(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})$ is not a feasible solution to the original problem (Eq.(10)), new cutting plane cp_{new}^{t+1} will be generated in agent i to separate the point $(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})$ from the feasible region of constraint $\bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}) \leq \varepsilon$. Specifically, we aim to find a valid cutting plane Boyd and Vandenberghe [2007] that satisfies

$$\begin{cases} \sum_{j \in \mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j \in \mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l \leq 0, \forall (\{\mathbf{x}_j\}, \{\mathbf{y}_j\}), j \in \mathcal{N}_i \text{ satisfies } \bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}) \leq \varepsilon \\ \sum_{j \in \mathcal{N}_i} \boldsymbol{a}_{j,l}^\top \mathbf{x}_j + \sum_{j \in \mathcal{N}_i} \boldsymbol{b}_{j,l}^\top \mathbf{y}_j + \kappa_l > 0 \end{cases}. \tag{24}$$

Since $\bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\})$ is a convex function, we have that

$$\bar{h}_i(\{\mathbf{x}_j\}, \{\mathbf{y}_j\}) \geq \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\}) + \begin{bmatrix} \{\frac{\partial \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})}{\partial \mathbf{x}_j}\} \\ \{\frac{\partial \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})}{\partial \mathbf{y}_j}\} \end{bmatrix}^\top \left(\begin{bmatrix} \{\mathbf{x}_j\} \\ \{\mathbf{y}_j\} \end{bmatrix} - \begin{bmatrix} \{\mathbf{x}_j^{t+1}\} \\ \{\mathbf{y}_j^{t+1}\} \end{bmatrix} \right). \tag{25}$$

Combing Eq.(24) with Eq.(25), we can find a valid cutting plane cp_{new}^{t+1} w.r.t. point $(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})$ as

$$\bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\}) + \begin{bmatrix} \{\frac{\partial \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})}{\partial \mathbf{x}_j}\} \\ \{\frac{\partial \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\})}{\partial \mathbf{y}_j}\} \end{bmatrix}^\top \left(\begin{bmatrix} \{\mathbf{x}_j\} \\ \{\mathbf{y}_j\} \end{bmatrix} - \begin{bmatrix} \{\mathbf{x}_j^{t+1}\} \\ \{\mathbf{y}_j^{t+1}\} \end{bmatrix} \right) \leq \varepsilon. \tag{26}$$

By adding cp_{new}^{t+1}, the polytope $\boldsymbol{\mathcal{P}}_i^{t+1}$ and dual variable set $\{\lambda_i^{t+1}\}$ will be updated as follows,

$$\boldsymbol{\mathcal{P}}_i^{t+1} = \begin{cases} \text{Add}(\boldsymbol{\mathcal{P}}_i^{t+1}, cp_{new}^{t+1}), \text{if } \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\}) > \varepsilon \\ \boldsymbol{\mathcal{P}}_i^{t+1}, \text{otherwise.} \end{cases}, \tag{27}$$

$$\{\lambda_i^{t+1}\} = \begin{cases} \text{Add}(\{\lambda_i^{t+1}\}, \lambda_{|\boldsymbol{\mathcal{P}}_i^{t+1}|}^{t+1}), \text{if } \bar{h}_i(\{\mathbf{x}_j^{t+1}\}, \{\mathbf{y}_j^{t+1}\}) > \varepsilon \\ \{\lambda_i^{t+1}\}, \text{otherwise} \end{cases}, \tag{28}$$

where $\text{Add}(\{\lambda_i^{t+1}\}, \lambda_{|\boldsymbol{\mathcal{P}}_i^{t+1}|}^{t+1})$ represents that $\lambda_{|\boldsymbol{\mathcal{P}}_i^{t+1}|}^{t+1}$ is added into the dual variable set $\{\lambda_i^{t+1}\}$. Finally, each agent transmits the updated $\boldsymbol{\mathcal{P}}_i^{t+1}$ and $\{\lambda_{i,l}^{t+1}\}$ to its neighbors. The details of the proposed algorithm are summarized in Algorithm 1.

Algorithm 1 SPAM: Stochastic Polyhedral Approximation Method

Initialization: global iteration $t = 0$; local variables $\{\mathbf{x}_i^0\}$, $\{\mathbf{y}_i^0\}$, $\{\lambda_{i,l}^0\}$, $\{\boldsymbol{\theta}_{i,j}^0\}$, $\{\mathbf{d}_i^0\}$, $\{\mathbf{u}_i^0\}$, $\{\mathbf{e}_i^0\}$, $\{\mathbf{v}_i^0\}$; polytope $\{\boldsymbol{\mathcal{P}}_i^0\}$.
repeat
  Communicate and update local variables according to (17) and (18);
  Update dual variables according to (19) and (20);
  Track the gradients by (21);
  if $(t+1)$ mod $k == 0$ and $t < T_1$ **then**
    Computes $\phi(\{\mathbf{x}_i\})$ by (9);
    Update $\boldsymbol{\mathcal{P}}_i^{t+1}$ and λ_i^{t+1} by (22), (23), (27), (28);
  end if
  $t = t + 1$;
until termination.

4 Discussion

Definition 1 *(Proximal gradient mapping)* Following standard definitions in Xin et al. [2021a], given $\mathbf{a} \in \text{dom}(r)$, $\mathbf{b}$, and $\eta > 0$, define the proximal gradient mapping of $\mathbf{b}$ at $\mathbf{a}$ to be

$$P(\mathbf{a}, \mathbf{b}, \eta) \triangleq \frac{1}{\eta}(\mathbf{a} - \text{prox}_{\eta R}(\mathbf{a} - \eta \mathbf{b})), \tag{29}$$

where *prox* denotes the proximal operator $\text{prox}_g(\mathbf{v}) = \arg\min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \mathbf{v}\|^2 \right\}$.

Definition 2 *(Convergence metric)* Inspired by Mancino-ball et al. [2023], Jiao et al. [2023], we use the following convergence metric for the proposed SPAM algorithm:

$$\Psi^t \triangleq \mathbb{E}[||\mathcal{G}^t||^2 + L^2 C^t]. \tag{30}$$

The first term quantifies the convergence of variables to a stationary gap of the global objective, while the second term measures the consensus error among local copies of both the UL and LL variables. Specifically, $\mathcal{G}^t$ is defined as

$$\mathcal{G}^t = \begin{bmatrix} \{P(\mathbf{x}_i, \bar{\nabla}_{\mathbf{x}_i} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}), \eta_x)\} \\ \{\bar{\nabla}_{\mathbf{y}_i} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\})\} \\ \left\{ \nabla_{\lambda_{i,l}} \tilde{L}_p\left(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}\right) \right\} \\ \left\{ \nabla_{\boldsymbol{\theta}_{i,j}} \tilde{L}_p\left(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}\right) \right\} \end{bmatrix}, \tag{31}$$

where $\bar{\nabla}_{\mathbf{x}_i} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}) \triangleq \frac{1}{N} \sum_{j=1}^N \nabla_{\mathbf{x}_j} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\})$ and $\bar{\nabla}_{\mathbf{y}_i} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}) \triangleq \frac{1}{N} \sum_{j=1}^N \nabla_{\mathbf{y}_j} \tilde{L}_p(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\})$.

And C^t is defined as

$$\sum_{i=1}^N (||\mathbf{x}_i^t - \bar{\mathbf{x}}^t||^2 + ||\mathbf{y}_i^t - \bar{\mathbf{y}}^t||^2), \tag{32}$$

where $\bar{\mathbf{x}}^t = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^t$, $\bar{\mathbf{y}}^t = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i^t$.

Assumption 1 Following Qian et al. [2019], Ji et al. [2021], Jiao et al. [2023], Mancino-ball et al. [2023], we assume that functions and variables satisfy:

a) $L'_{p\_i}, i \in [N]$ has Lipschitz continuous gradients, i.e., for any $\mathbf{a}, \mathbf{b}$, there exists $L > 0$ satisfying that $||\nabla L'_{p\_i}(\mathbf{a}) - \nabla L'_{p\_i}(\mathbf{b})|| \leq L||\mathbf{a} - \mathbf{b}||$.

b) R and r are convex, possibly nonsmooth functions, such as l_1 norm and Huber norm Zadorozhnyi et al. [2016]. They admit proximal mappings that are easily computable.

c) The dual variables are bounded, i.e., $||\lambda_{i,l}||^2 \leq \alpha_1$, $||\boldsymbol{\theta}_{ij}||^2 \leq \alpha_2$.

d) The estimates of gradient have bounded variances, i.e.,

$$\mathbb{E}_{\xi_i}||\nabla_{\mathbf{x}_i} L'_{p\_i}(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}) - \nabla_{\mathbf{x}_i} L'_{p\_i}(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\}; \xi_i^t||^2 \leq \sigma_1^2.$$

Similarly, the gradient estimation variances of $\mathbf{y}_i$, $\lambda_{i,l}$ and $\boldsymbol{\theta}_{i,j}$ are bounded by σ_2^2, σ_3^2 and σ_4^2.

Assumption 2 As is standard in decentralized learning Lian et al. [2017], Xin et al. [2021b], Mancino-ball et al. [2023], we assume the mixing matrix $\mathbf{W}$ satisfies the following properties:

a) Network-defined sparsity: $\mathbf{W}_{ij} > 0$ if $(i, j) \in \mathcal{E}$; otherwise $\mathbf{W}_{ij} = 0, \forall i, j \in \mathcal{V}$.

b) Symmetric: $\mathbf{W} = \mathbf{W}^\top$.

c) Null-space property: null $(\mathbf{I} - \mathbf{W})$ = span$\{\mathbf{e}\}$, where $\mathbf{e} \in \mathbb{R}^N$ is the vector of all ones.

d) Spectral property: The eigenvalues of $\mathbf{W}$ lie in the range $(1, 1]$ with $\rho \triangleq \left\|\mathbf{W} - \frac{1}{N}\mathbf{e}\mathbf{e}^\top\right\|_2 < 1$, where the value ρ indicates the connectedness of the graph Mancino-ball et al. [2023].

Theorem 1 (*Convergence of SPAM*) Under Assumptions 1 and 2, when step-sizes satisfy $\eta_x = \eta_y = \eta_\lambda = \eta_\theta = \eta$, and $\eta \leq \min\{(\frac{1}{16NL})^{\frac{1}{2}}, (\frac{1}{4L^2(M+N)N})^{\frac{1}{2}}, \frac{(1-\rho)^3}{16L}, \frac{(1-\rho)^2}{48L}, \frac{N(N+M)}{L}, \frac{12}{N}, \frac{1}{2N(1-\rho)^3}, \frac{2}{LN(1-\rho)^3}, \frac{1}{2(1+L)}, \frac{N(1-\rho)^3}{104L}, (\frac{(1+\frac{2}{1-\rho})4NL^2(N+M)}{26L^2})^{\frac{1}{2}}, (\frac{1}{2(M+N)NL^2})^{\frac{1}{2}}, (\frac{N(1-\rho)}{16(1-\rho)+128NL^2(M+N)})^{\frac{1}{2}}$ $(\frac{(1-\rho)^3}{24L^2+(1-\rho)^2L(\gamma_3+48NL^2(N+M))+10L(1-\rho)^3})^{\frac{1}{2}}\}$, then $(\{\mathbf{x}_i^t\}, \{\mathbf{y}_i^t\}, \{\lambda_{i,l}^t\}, \{\boldsymbol{\theta}_{i,j}^t\})$ generated by SPAM satisfies

$$\frac{1}{T-1}\sum_{t=T_1+2}^{T_1+T}(||\mathcal{G}^t||^2 + C^t) \leq \frac{(8+32NL(N+M))d}{(T-1)\eta^2} + C_{bias} = \mathcal{O}(\frac{1}{T-1}) + C_{bias}, \tag{33}$$

where C_{bias} and d are constants. Note that C_{bias} is a bias term affected by the stochastic gradient estimator. Details of the proof and the value of constant are given in the supplementary material.

5 Experiment

In this section, we evaluate SPAM in two examples: meta-learning and hyperparameter optimization. A network with 5 agents is generated artificially by the Erds-Rényi random graph approach. We use two state-of-the-art decentralized bilevel optimization methods INTERACT Liu et al. [2022] and DSBO Yang et al. [2022] as baselines. Besides, we perform ablation experiment to explore the effectiveness of different components in SPAM. In meta-learning task, experiments are carried out on Omniglot Vinyals et al. [2016] and Double MNISTMorerio et al. [2017] dataset. In hyper-parameter optimization task, datasets MNIST LeCun et al. [1998] and Fashion MNIST Xiao et al. [2017] are used.

Meta-Learning. We consider the meta-learning problem in the context of few-shot supervised learning. Following a general bi-level meta-learning setup, the decentralized meta-learning problem in our experiments can be expressed as,

$$\begin{aligned}
&\min \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\theta}_i) \triangleq \mathcal{L}_i(\boldsymbol{\varphi}_i(\boldsymbol{\theta}_i); \mathcal{D}_i^{val}) + R(\boldsymbol{\theta}_i) \\
&\text{s.t.}\{\boldsymbol{\varphi}_i(\boldsymbol{\theta}_i)\} = \arg\min_{\{\boldsymbol{\varphi}_i'\}} \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta}_i, \boldsymbol{\varphi}_i') \triangleq \mathcal{L}_i(\boldsymbol{\varphi}_i'; \mathcal{D}_i^{tr}) + r(\boldsymbol{\varphi}_i') + ||\boldsymbol{\varphi}_i' - \boldsymbol{\theta}_i||^2 \\
&\text{var.}\{\boldsymbol{\theta}_i\},
\end{aligned} \tag{34}$$

where $\boldsymbol{\theta}_i \in \mathbb{R}^n$ and $\boldsymbol{\varphi} \in \mathbb{R}^n$ denote the UL variable (meta-parameter) and LL variable (task-specific parameter) of the i^{th} agent, respectively. $\mathcal{D}_i^{tr}$ and $\mathcal{D}_i^{val}$ is the training and validation datasets on the i^{th} agent. N is the number of agents, $\mathcal{L}$ is the cross-entropy loss, r and R are l_1 norm.

Hyperparameter Optimization. We consider the hyperparameter optimization in the context of data hyper-cleaning. It involves training a model in contaminated environments where each training label is changed randomly. We consider the following decentralized data hyper-cleaning problem:

$$\begin{aligned} &\min \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{\psi}_i, \boldsymbol{w}_i) \triangleq \frac{1}{|\mathcal{D}_i^{val}|}\sum_{(\mathbf{x}_j, y_j)\in\mathcal{D}_i^{val}} \mathcal{L}_i(\mathbf{x}_j^\top \boldsymbol{w}_i, y_j) \\ &\text{s.t.}\{\boldsymbol{w}_i\} = \arg\min_{\{\boldsymbol{w}_i'\}} \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\psi}_i, \boldsymbol{w}_i') \triangleq \frac{1}{|\mathcal{D}_i^{tr}|}\sum_{(\mathbf{x}_j, y_j)\in\mathcal{D}_i^{tr}} \sigma(\psi_{i,j})\mathcal{L}_i(\mathbf{x}_j^\top \boldsymbol{w}_i', y_j) + R(\boldsymbol{w}_i') \\ &\text{var.}\{\boldsymbol{\psi}_i\}, \{\boldsymbol{w}_i\}, \end{aligned} \tag{35}$$

where $\mathcal{D}_i^{tr}$ and $\mathcal{D}_i^{val}$ is the training and validation datasets on the i^{th} agent. $(\mathbf{x}_j, y_j)$ denotes the j^{th} data and label. σ represents the sigmoid function, $\mathcal{L}$ is the cross-entropy loss, R is l_1 norm.

The results of the meta-learning task on Omniglot are shown in Figure 1 and Figure 2. The results of the hyperparameter optimization task on MNIST are shown in Figure 3 and Figure 4. In ablation experiments, D-SGD refers to the decentralized stochastic gradient descent; GT-SGD and D-CPSGD respectively apply gradient tracking techniques and cutting planes approximation on the basis of D-SGD; D-SPGD replaces the gradient descent in D-SGD with proximal gradient descent. Detailed settings and more experiments are provided in the supplementary material.

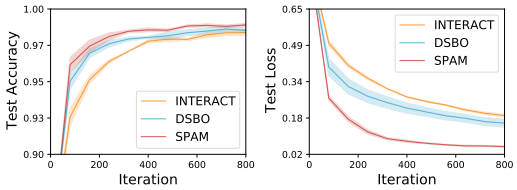

Figure 1: Performance comparison with baseline methods on Omniglot dataset.

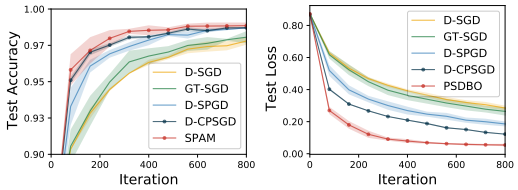

Figure 2: Results of ablation experiments on Omniglot dataset.

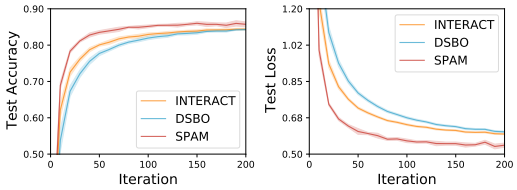

Figure 3: Performance comparison with baseline methods on MNIST dataset.

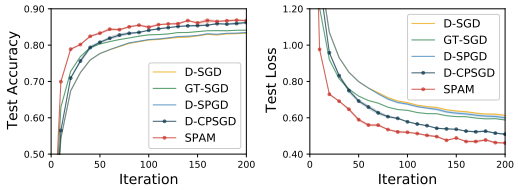

Figure 4: Results of ablation experiments on MNIST dataset.

Overall, it can be observed that SPAM exhibits superior performance over other single-loop decentralized algorithms. The observation of ablation experiments also suggests that SPAM is indeed more competitive in solving the nonconvex composite decentralized bilevel optimization since 1) the polyhedral approximation used by SPAM is efficient in the decentralized settings; 2) leveraging proximal operator and gradient tracking improves the efficiency of handling non-smooth terms and communication.

6 Conclusion

In this paper, we developed a stochastic polyhedral approximation method called SPAM for solving general nonconvex composite bilevel optimization problems in a fully decentralized way. The theoretical results show that SPAM finds a stationary point at a rate $\mathcal{O}(1/\epsilon)$. Our empirical studies on meta-learning and hyperparameter optimization corroborate the effectiveness of SPAM. As far as we are aware of, our work represents the first step to explore the composite decentralized bilevel learning.

References

Kaiyi Ji, Jason D Lee, Yingbin Liang, and H Vincent Poor. Convergence of meta-learning with task-specific adaptation over partial parameters. *Advances in Neural Information Processing Systems*, 33:11490–11500, 2020.

Valerii Likhosherstov, Xingyou Song, Krzysztof Choromanski, Jared Q Davis, and Adrian Weller. Debiasing a first-order heuristic for approximate bi-level optimization. In *International Conference on Machine Learning*, pages 6621–6630. PMLR, 2021.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

Zhangyu Chen, Dong Liu, Xiaofei Wu, and Xiaochun Xu. Research on distributed renewable energy transaction decision-making based on multi-agent bilevel cooperative reinforcement learning. 2019.

Chenhan Jiang, Hang Xu, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Sp-nas: Serial-to-parallel backbone search for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11863–11872, 2020.

Yang Jiao, Kai Yang, Dongjing Song, and Dacheng Tao. Timeautoad: Autonomous anomaly detection with self-supervised contrastive loss for multivariate time series. *IEEE Transactions on Network Science and Engineering*, 9(3):1604–1619, 2022a.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283, 2021.

Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In *International Conference on Machine Learning*, pages 6882–6892. PMLR, 2021.

Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Tejas Subramanya and Roberto Riggio. Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond. *IEEE Transactions on Network and Service Management*, 18(1):63–78, 2021.

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

Mu Li, Li Zhou, Zichao Yang, Aaron Li, Fei Xia, David G Andersen, and Alexander Smola. Parameter server for distributed machine learning. In *Big learning NIPS workshop*, volume 6, 2013.

Zhuqing Liu, Xin Zhang, Prashant Khanduri, Songtao Lu, and Jia Liu. 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*, pages 61–70, 2022.

Songtao Lu, Siliang Zeng, Xiaodong Cui, Mark Squillante, Lior Horesh, Brian Kingsbury, Jia Liu, and Mingyi Hong. A stochastic linearized augmented lagrangian method for decentralized bilevel optimization. *Advances in Neural Information Processing Systems*, 35:30638–30650, 2022.

Shuoguang Yang, Xuezhou Zhang, and Mengdi Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. In *Advances in Neural Information Processing Systems*, volume 35, pages 238–252, 2022.

Peiwen Qiu, Yining Li, Zhuqing Liu, Prashant Khanduri, Jia Liu, Ness B Shroff, Elizabeth Serena Bentley, and Kurt Turck. Diamond: Taming sample and communication complexities in decentralized bilevel optimization. *arXiv preprint arXiv:2212.02376*, 2022.

Xuxing Chen, Minhui Huang, and Shiqian Ma. Decentralized bilevel optimization. *arXiv preprint arXiv:2206.05670*, 2022.

Qing Wang, Jie Chen, Xianlin Zeng, and Bin Xin. Distributed proximal-gradient algorithms for nonsmooth convex optimization of second-order multiagent systems. *International Journal of Robust and Nonlinear Control*, 30(17):7574–7592, 2020.

Xia Jiang, Xianlin Zeng, Jian Sun, and Jie Chen. Distributed proximal gradient algorithm for non-convex optimization over time-varying networks. *IEEE Transactions on Control of Network Systems*, 2022.

Kai Yang, Jianwei Huang, Yihong Wu, Xiaodong Wang, and Mung Chiang. Distributed robust optimization (dro), part i: Framework and example. *Optimization and Engineering*, 15:35–67, 2014.

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

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

Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34: 11422–11435, 2021.

Boyue Li, Shicong Cen, Yuxin Chen, and Yuejie Chi. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *The Journal of Machine Learning Research*, 21(1):7331–7381, 2020.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pages 26693–26712. PMLR, 2022.

Arpan Biswas and Christopher Hoyle. A literature review: solving constrained non-linear bi-level optimization problems with classical methods. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 59193, page V02BT03A025. American Society of Mechanical Engineers, 2019.

Renjie Liao, Yuwen Xiong, Ethan Fetaya, Lisa Zhang, KiJung Yoon, Xaq Pitkow, Raquel Urtasun, and Richard Zemel. Reviving and improving recurrent back-propagation. In *International Conference on Machine Learning*, pages 3082–3091. PMLR, 2018.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR, 2020.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682, 2021.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.

Yang Jiao, Kai Yang, Tiancheng Wu, Dongjin Song, and Chengtao Jian. Asynchronous distributed bilevel optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuwei Sun, Hideya Ochiai, and Hiroshi Esaki. Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. *IEEE Transactions on Artificial Intelligence*, 3(6):963–972, 2021.

Liang Zhao and WenZhan Song. Decentralized consensus in distributed networks. *International Journal of Parallel, Emergent and Distributed Systems*, 33(6):550–569, 2018.

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

Andreas Lundell and Jan Kronqvist. Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in shot. *Journal of Global Optimization*, 82(4):863–896, 2022.

Yang Jiao, Kai Yang, and Dongjin Song. Distributed distributionally robust optimization with non-convex objectives. In *Advances in Neural Information Processing Systems*, 2022b.

Kaan Gokcesu and Hakan Gokcesu. Generalized huber loss for robust learning and its efficient minimization for a robust statistics. *arXiv preprint arXiv:2108.12627*, 2021.

Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

Junyi Li, Feihu Huang, and Heng Huang. Local stochastic bilevel optimization with momentum-based variance reduction. *arXiv preprint arXiv:2205.01608*, 2022.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Stephen Boyd and Lieven Vandenberghe. Localization and cutting-plane methods. *From Stanford EE 364b lecture notes*, 2007.

Vojtech Franc, Sören Sonnenburg, and Tomáš Werner. Cutting plane methods in machine learning. *Optimization for Machine Learning*, pages 185–218, 2011.

Alexander Michalka. *Cutting planes for convex objective nonconvex optimization*. Columbia University, 2013.

Douglas AG Vieira and Adriano Chaves Lisboa. A cutting-plane method to nonsmooth multiobjective optimization problems. *European Journal of Operational Research*, 275(3):822–829, 2019.

Zi Xu, Huiling Zhang, Yang Xu, and Guanghui Lan. A unified single-loop alternating gradient projection algorithm for nonconvex–concave and convex–nonconcave minimax problems. *Mathematical Programming*, pages 1–72, 2023.

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

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

Qi Qian, Shenghuo Zhu, Jiasheng Tang, Rong Jin, Baigui Sun, and Hao Li. Robust optimization over multiple domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4739–4746, 2019.

Oleksandr Zadorozhnyi, Gunthard Benecke, Stephan Mandt, Tobias Scheffer, and Marius Kloft. Huber-norm regularization for linear prediction models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*, pages 714–730. Springer, 2016.

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

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

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, René Vidal, and Vittorio Murino. Curriculum dropout. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3544–3552, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.