First-Order Federated Bilevel Learning

Yifan Yang[1], Peiyao Xiao[1], Shiqian Ma[2], Kaiyi Ji[1]

[1]Department of Computer Science and Engineering, University at Buffalo
[2]Department of Computational Applied Math and Operations Research, Rice University
yyang99@buffalo.edu, peiyaoxi@buffalo.edu, sqma@rice.edu, kaiyiji@buffalo.edu

Abstract

Federated bilevel optimization (FBO) has garnered significant attention lately, driven by its promising applications in meta-learning and hyperparameter optimization. Existing algorithms generally aim to approximate the gradient of the upper-level objective function (hypergradient) in the federated setting. However, because of the nonlinearity of the hypergradient and client drift, they often involve complicated computations. These computations, like multiple optimization sub-loops and second-order derivative evaluations, end up with significant memory consumption and high computational costs. In this paper, we propose a computationally and memory-efficient FBO algorithm named MemFBO. MemFBO features a fully single-loop structure with all involved variables updated simultaneously, and uses only first-order gradient information for all local updates. We show that MemFBO exhibits a linear convergence speedup with milder assumptions in both partial and full client participation scenarios. We further implement MemFBO in a novel FBO application for federated data cleaning. Our experiments, conducted on this application and federated hyper-representation, demonstrate the effectiveness of the proposed algorithm.

1 Introduction

Bilevel optimization has received significant attention recently in a wide range of applications including few-shot meta-learning (Finn, Abbeel, and Levine 2017; Rajeswaran et al. 2019), hyperparameter search (Franceschi et al. 2018; Feurer and Hutter 2019), fairness-aware machine learning (Roh et al. 2021), lifelong learning (Hao, Ji, and Liu 2023), etc. Due to the significant increase in the demand for efficient computing and growing concerns about user privacy, recent studies have shifted their attention to efficiently solving the following federated bilevel optimization (FBO) problem:

$$\min_{x\in\mathbb{R}^p} \Phi(x) = F\big(x, y^*(x)\big) := \frac{1}{n}\sum_{i=1}^n f_i(x, y^*(x))$$
$$\text{s.t. } y^*(x) = \arg\min_{y\in\mathbb{R}^q} G(x,y) := \frac{1}{n}\sum_{i=1}^n g_i(x,y), \quad (1)$$

where n is the total number of clients; p and q are the dimension of x and y; $f_i(x, y^*(x)) = \mathbb{E}_\xi\big[f_i\big(x, y^*(x); \xi_i\big)\big]$ and $g_i(x,y) = \mathbb{E}_\zeta\big[g_i(x,y;\zeta_i)\big]$ represent the expectation of the i_{th} upper- and lower-level function w.r.t. the random variables ξ_i, ζ_i, respectively. Various FBO algorithms have been developed recently (Tarzanagh et al. 2022; Xiao and Ji 2023; Yang, Xiao, and Ji 2023b) and applied to important applications such as federated meta-learning (Fallah, Mokhtari, and Ozdaglar 2020; Xiao and Ji 2023), and graph-aided federated learning (Xing et al. 2022). However, existing algorithms often suffer from some computational and memory issues due to a complex implementation and the computation of high-order model information, induced by the nested optimization structure in FBO. For example, (Huang et al. 2022; Tarzanagh et al. 2022; Huang, Zhang, and Ji 2023) and (Xiao and Ji 2023) proposed two types of FBO algorithms based on approximate implicit differentiation (AID) and iterative differentiation (ITD), which involves multiple sub-loops to approximate the global federated hypergradient (i.e., gradient of the upper-level objective function). Most recently, (Yang, Xiao, and Ji 2023b) proposed a more efficient single-loop FBO algorithm called SimFBO without any sub-loop. However, the local aggregation in each client of SimFBO needs to store multiple second-order Hessian or Jacobian-vector products. This could cause practical challenges related to computing and memory, as edge devices like smartphones typically have relatively low memory availability and computational capacity. To address these challenges, this paper proposes a computationally and memory-efficient FBO algorithm named MemFBO, and further applies MemFBO to a novel application of FBO in robust federated learning with potentially noisy data. Our specific contributions are summarized as follows.

- Inspired by the Lagrangian approximation of bilevel optimization, we use simpler single-level minimax problems as effective substitutes for the original local and global bilevel problems, making them easier to manage. We then propose a simple and efficient FBO algorithm named MemFBO. As illustrated in Figure 1, MemFBO features a fully single-loop optimization structure and computes only gradient information at each local step for all participating clients.

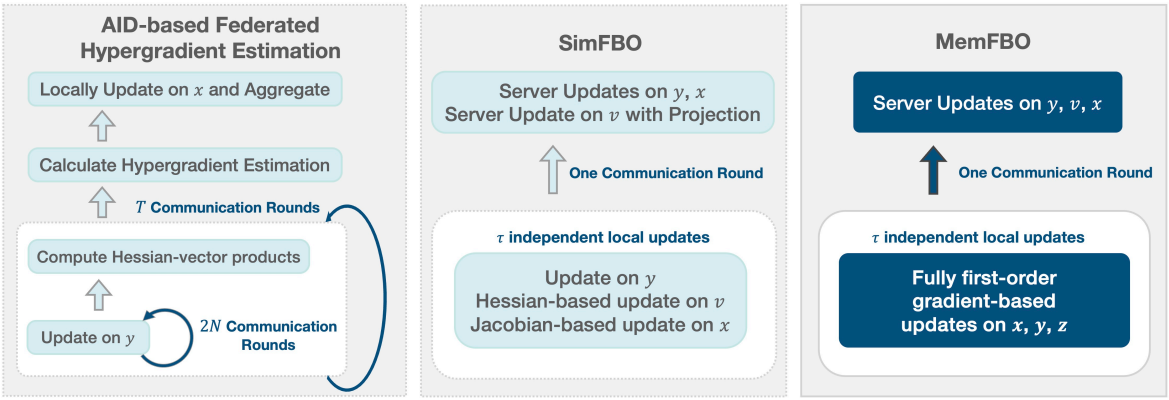

Figure 1: Algorithmic comparison of federated hypergradient-based methods FedNest (Tarzanagh et al. 2022) (left), SimFBO (Yang, Xiao, and Ji 2023b) (middle) and our MemFBO (right). MemFBO has a similar single-loop structure as SimFBO, but performs more efficient updates using only first-order gradient information and without any projection.

- We provide a comprehensive convergence analysis for MemFBO under both partial and full participation. We show that MemFBO achieves a linear speedup (i.e., the communication complexity is improved linearly with respect to the number of sampled clients) under client sampling without replacement. Technically, we eliminate the restrictive assumption of SimFBO (Yang, Xiao, and Ji 2023b) on the third-order Lipschitz continuity of all local lower-level objective functions. Compared to the analysis of Lagrangian methods in standard bilevel optimization, our characterizations need to 1) control the variance associated with local stochastic estimation and the errors from client sampling, 2) mitigate the impact of client drift on the final convergence, and 3) mitigate the negative impact of the large Lagrange multiplier on amplifying the client drifts and gradient estimation variances.

- We further explore a new application of FBO: robust federated data cleaning via adaptive weight selection. We are interested in the potential setting where there are two sets of data in each client: one is clean and of high quality, while the other is possibly noisy and constitutes the majority. For example, in mobile devices, the noisy data often contain errors or variations due to measurement or transmission issues, such as sensor errors, network delays, or channel noise. To deal with this challenge, we propose a federated data cleaning approach via the lens of FBO, where the lower-level is weighted loss over noisy samples of all participating clients, and the upper-level is to fine-tune these weights to maximize the prediction accuracy on the clean data.

- We implement MemFBO in this application and demonstrate its more robust performance in the noisy setting than multiple popular (personalized) federated learning baselines. Moreover, our experiments on federated hyper-representation also demonstrate improved efficiency and accuracy over existing FBO approaches.

2 Related Work

Bilevel optimization. Bilevel optimization has been extensively studied since first introduced by (Bracken and McGill 1973). Early works (Hansen, Jaumard, and Savard 1992; Shi, Lu, and Zhang 2005; Gould et al. 2016; Sinha, Malo, and Deb 2017) solved the bilevel problem from a constrained optimization perspective. More recently, gradient-based bilevel methods have received intensive attention due to the effiency and effectiveness in machine learning problems, which approximate implicit differentiation (AID) (Domke 2012; Liao et al. 2018; Pedregosa 2016; Lorraine, Vicol, and Duvenaud 2020; Grazzi et al. 2020; Ji, Yang, and Liang 2021; Arbel and Mairal 2021; Hong et al. 2023) and iterative differentiation (ITD) (Finn, Abbeel, and Levine 2017; Franceschi et al. 2017; Shaban et al. 2019; Grazzi et al. 2020; Liu et al. 2021b; Ji, Yang, and Liang 2021) based approaches. Stochastic bilevel algorithms have been studied recently using techniques such as Neumann series (Chen, Sun, and Yin 2021; Ji, Yang, and Liang 2021; Arbel and Mairal 2021), variance reduction (Yang, Ji, and Liang 2021; Dagréou et al. 2022; Yang, Ji, and Liang 2021; Huang and Huang 2021; Guo et al. 2021), finite-difference matrix-vector estimation (Yang, Xiao, and Ji 2023a). Another class of approaches formulated the lower-level problem as a value-function-basaed constraint (Sabach and Shtern 2017; Liu et al. 2020; Li, Gu, and Huang 2020; Liu et al. 2021b,a, 2022; Sow et al. 2022; Shen and Chen 2023). The more relevant works (Kwon et al. 2023; Wang et al. 2023) proposed first-order algorithms by solving a single-level Lagrangian-induced problem. Inspired by a similar idea, we propose effective single-level surrogates for local and global problems in FBO, and further develop a novel efficient algorithm with strong resilience to the client drifts.

Federated bilevel optimization. Federated bilevel optimization has not been widely explored except in a few studies. (Gao 2022; Li, Huang, and Huang 2022) proposed momentum-based algorithms in homogeneous settings. (Tarzanagh et al. 2022; Huang, Zhang, and Ji 2023) proposed FedNest and FedMBO using an AID-based fed-

erated hypergradient estimator in heterogeneous settings. (Xiao and Ji 2023) and (Yang, Xiao, and Ji 2023b) further improved the communication efficiency of AID-based methods via ITD-based hypergradient aggregation and simple single-loop optimization. (Huang 2022; Li, Huang, and Huang 2023) exploited momentum-based variance reduction to reduce the overall sample complexity. In addition, there are some studies focused on other distributed settings, including decentralized bilevel optimization (Chen, Huang, and Ma 2022; Chen et al. 2023; Yang, Zhang, and Wang 2022; Lu et al. 2022; Dong et al. 2023), distributed network utility maximization (Ji and Ying 2023), and asynchronous optimization over directed network (Yousefian 2021). In this paper, we propose a simple first-order FBO method and explore its performance in an important federated application.

3 Existing Methods and Challenges

3.1 Computing and Memory Challenges

To solve the problem in eq. (1) efficiently, the high-level idea of existing works like (Tarzanagh et al. 2022; Gao 2022; Li, Huang, and Huang 2022; Xiao and Ji 2023; Yang, Xiao, and Ji 2023b) is to approximate the following federated hypergradient (i.e., the gradient of the upper-level Φ function) induced by implicit function theorem:

$$\begin{aligned}\nabla\Phi(x) =&\nabla_x F^* - \nabla^2_{xy} G^* \left[\nabla^2_{yy} G^*\right]^{-1} \nabla_y F^*, \\ \neq& \frac{1}{n}\sum_{i=1}^n \nabla_x f_i^* - \nabla^2_{xy} g_i^* \left[\nabla^2_{yy} g_i^*\right]^{-1} \nabla_y f_i^*, \quad (2)\end{aligned}$$

where we denote $f_i^* := f_i(x, y^*(x))$, $g_i^* := g_i(x, y^*(x))$, $F^* := F(x, y^*(x))$ and $G^* := G\big(x, y^*(x)\big)$. As shown in eq. (2), one cannot use the aggregation of local hypergradients to approximate the global hypergradient $\nabla\Phi(x)$ due to the nonlinearity from the matrix multiplication and the inversion of the global Hessian matrix $\nabla_{yy} G^*$. Various approaches have been used to address this challenge. For example, (Tarzanagh et al. 2022; Gao 2022; Li, Huang, and Huang 2022) extended the idea of AID-based hypergradient estimation (Ghadimi and Wang 2018) to the federated setting, and proposed AID-based FBO algorithms, which require each participating client to compute a large number of Hessian-vector products in estimating the global Hessian-inverse-vector product of the federated hypergradient. (Xiao and Ji 2023) proposed an ITD-based FBO algorithm to reduce the communication cost of AID-based methods, but still requires numerous Hessian-vector computations in each communication round. Most recently, (Yang, Xiao, and Ji 2023b) proposed a more efficient single-loop FBO algorithm named SimFBO by updating variables y, v, x simultaneously, whose each local step is given by

$$\begin{pmatrix} y_i^{k+1} \\ v_i^{k+1} \\ x_i^{k+1} \end{pmatrix} \leftarrow \begin{pmatrix} y_i^k \\ v_i^k \\ x_i^k \end{pmatrix} - \begin{pmatrix} \eta_y \nabla_y g_i\big(x_i^k, y_i^k; \zeta_i^k\big) \\ \eta_v \nabla_v R_i\big(x_i^k, y_i^k, v_i^k; \psi_i^k\big) \\ \eta_x \bar{\nabla} f_i\big(x_i^{(k}, y_i^k, v_i^k; \xi_i^k\big) \end{pmatrix}$$

where $\bar{\nabla} f_i(x, y, v; \xi) = \nabla_x f_i(x, y; \xi) - \nabla^2_{xy} g_i(x, y; \xi) v_i$ is the local hypergradient estimate for client i and

$$R_i(x, y, v) = \frac{1}{2} v^T \nabla^2_{yy} g_i(x, y) v - v^T \nabla_y f_i(x, y)$$

is the loss function of client i for approximating the global Hessian-inverse-vector product in eq. (2). Then, SimFBO performs a local aggregation in each client i as

$$q_i = \sum_{k=0}^{\tau-1} a_i^k \nabla_v R_i\big(x_i^k, y_i^k v_i^k; \psi_i^k\big), \quad (3)$$

and similar aggregations are also applied to the local updates of x and y. Then, it can be seen from eq. (3) that the local aggregation of SimFBO on v and x needs to store τ (i.e., the number of local steps) Hessian- or Jacobian-vector products, which can incur substantial computing and memory costs as further shown in Figure 4 in the experiments.

3.2 Our Solution: First-Order FBO

Inspired by the recent advance in first-order bilevel optimization (Lin, Xu, and Ye 2014; Liu et al. 2021a; Kwon et al. 2023; Wang et al. 2023), we reformulate the FBO problem in eq. (1) into an equivalent constrained problem

$$\min_x F(x, y) \quad \text{s.t.} \quad G(x, y) - G\big(x, y^*(x)\big) \le 0,$$

where $G\big(x, y^*(x)\big)$ is the optimal value function of the lower-level problem. One effective way to deal with this constrained problem is to solve its Lagrangian problem:

$$\min_{x,y} \max_z \mathcal{L}(x, y, z) = F(x, y) + \lambda\big[G(x, y) - G(x, z)\big],$$

where λ is the Lagrange multiplier. Note that the maximizer over z equals $y^*(x)$. The above Lagrangian approximation provides several advantages. First, differently from existing hypergradient-based methods, its gradients w.r.t. all variables x, y, z do not contain second-order derivatives such as Hessian or Jacobian matrices. Second, existing methods include complex sub-loops to tackle the nested optimization structure of the original FBO problem. In contrast, this Lagrangian problem is a minimax problem with a single objective, and hence facilitates simpler implementation in federated learning. Third, the Lagrangian function can be decomposed into local Lagrangian functions as

$$\min_{x,y} \max_z \mathcal{L}(x, y, z) := \frac{1}{n}\sum_{i=1}^n \mathcal{L}_i(x, y, z). \quad (4)$$

where each local Lagrangian is given by

$$\mathcal{L}_i(x_i, y_i, z_i) := f_i(x_i, y_i) + \lambda\big[g_i(x_i, y_i) - g_i(x_i, z_i)\big].$$

As a result, each client can focus on its local Lagrangian problem as

$$\text{(Local Lagrangian:)} \quad \min_{x_i, y_i} \max_{z_i} \mathcal{L}_i(x_i, y_i, z_i), \quad (5)$$

which also facilitates single-loop updates on x_i, y_i and z_i. For simplicity, we use the same Lagrange multiplier λ for all local problems. However, it is also feasible to use different multipliers $\lambda_i, i = 1, ..., n$ for different local Lagrangian problems because it can be shown that the gap between the hypergradient of the original problem and the gradient of the Lagrangian function $\mathcal{L}^*(x) = \min_y \max_z \mathcal{L}(x, y, z)$ w.r.t. x can vanish if we choose λ_i sufficiently large.

4 The MemFBO Algorithm

In this section, we introduce an efficient FBO algorithm named MemFBO, which features flexible updates and aggregation on both the client and server sides, as described in Algorithm 1.

Local updates. Specifically, at the beginning of each communication round, we sample a subset $\mathcal{C}_t$ of client **without replacement**. Then, each client in $\mathcal{C}_t$ performs τ local stochastic gradient descent (SGD) steps to optimize its local Lagrangian problem as

$$\begin{aligned} z_i^{t,k+1} \leftarrow & z_i^{t,k} - \eta_z \nabla_y g_i\big(x_i^{t,k}, z_i^{t,k}; \zeta_i^{t,k}\big), \\ y_i^{t,k+1} \leftarrow & y_i^{t,k} - \eta_y \nabla_y \mathcal{L}_i\big(x_i^{t,k}, y_i^{t,k}, z_i^{t,k}; \psi_i^{t,k}\big), \\ x_i^{t,k+1} \leftarrow & x_i^{t,k} - \eta_x \nabla_x \mathcal{L}_i\big(x_i^{t,k}, y_i^{t,k}, z_i^{t,k}; \xi_i^{t,k}\big), \end{aligned} \tag{6}$$

where η_x, η_y, η_x are positive local stepsizes and the stochastic Lagrangian function is defined as $\mathcal{L}_i(x, y, z; \xi) = f_i(x, y; \xi) + \lambda\big(g_i(x, y; \xi) - g_i(x, z; \xi)\big)$. Note from the above eq. (6) that the local updates on z, y and x are conducted simultaneously and hence can benefit a lot from the hardware parallelism. In addition, differently from SimFBO (Yang, Xiao, and Ji 2023b) whose local updates involve second-order derivatives, all steps in eq. (6) use only first-order gradient information and hence are more computing and memory friendly. It is also worth mentioning that the local SGD steps can be extended to momentum-based updates to achieve improved theoretical complexity.

Local aggregation and communication. In this stage, each client $i \in \mathcal{C}_t$ aggregates all τ local updates

$$\begin{aligned} h_{z,i}^t = & \frac{1}{\tau} \sum_{k=0}^{\tau-1} \nabla_y g_i\big(x_i^{t,k}, z_i^{t,k}; \zeta_i^{t,k}\big), \\ h_{y,i}^t = & \frac{1}{\tau} \sum_{k=0}^{\tau-1} \nabla_y \mathcal{L}_i\big(x_i^{t,k}, y_i^{t,k}, z_i^{t,k}; \psi_i^{t,k}\big), \\ h_{x,i}^t = & \frac{1}{\tau} \sum_{k=0}^{\tau-1} \nabla_x \mathcal{L}_i\big(x_i^{t,k}, y_i^{t,k}, z_i^{t,k}; \xi_i^{t,k}\big). \end{aligned} \tag{7}$$

These normalized aggregations $h_{z,i}^t, h_{y,i}^t$ and $h_{x,i}^t$ are then communicated to the server. Differently from existing AID-based and ITD-based FBO algorithms that contain multiple sub-loops and communication rounds to approximate the federated hypergradient in this stage, our updates here are much simpler with a single communication round involved.

Global aggregation and updates. The server collects the local aggregations $h_{z,i}^t, h_{y,i}^t$ and $h_{x,i}^t$, which are then further aggregated into

$$\big\{h_z^t, h_y^t, h_x^t\big\} = \frac{1}{|\mathcal{C}_t|} \sum_{i \in \mathcal{C}_t} \big\{h_{z,i}^t, h_{y,i}^t, h_{x,i}^t\big\}. \tag{8}$$

Based on the global aggregations $\{h_z^t, h_y^t, h_x^t\}$, the next step is to update global variables x_t, y_t and z_t simultaneously as

$$\begin{aligned} z_{t+1} &= z_t - \gamma_z h_z^t, \\ y_{t+1} &= y_t - \gamma_y h_y^t, \\ x_{t+1} &= x_t - \gamma_x h_x^t. \end{aligned} \tag{9}$$

Algorithm 1: MemFBO

Input: initialization x_0, y_0, z_0, local learning rates η_z, η_y, η_x, global learning rates $\gamma_z, \gamma_y, \gamma_x$, local update rounds τ, communication rounds T.
for $t = 0, 1, 2, ..., T-1$ **do**
  for $i \in \mathcal{C}_t$ **in parallel do**
    $z_i^{t,0} = z_t$, $y_i^{t,0} = y_t$, $x_i^{t,0} = x_t$
    for $k = 0, 1, 2, ..., \tau - 1$ **do**
      Locally update $z_i^{t,k}$, $y_i^{t,k}$ and $x_i^{t,k}$ simultaneously via eq. (6)
    end for
    Client i computes local aggregations $h_{z,i}^t, h_{y,i}^t, h_{x,i}^t$ via eq. (7)
    Client i communicates $\{h_{z,i}^t, h_{y,i}^t, h_{x,i}^t\}$ to server
  end for
  Server computes global aggregations $\{h_z^t, h_y^t, h_x^t\}$ via eq. (8) and update x_t, y_t, z_t via eq. (9)
end for

Note that the global updates in the above equation use a different set $\{\gamma_z, \gamma_y, \gamma_x\}$ of stepsizes from $\{\eta_z, \eta_y, \eta_x\}$ in the local updates. This flexible design not only facilitates the algorithmic deployment because the local stepsizes are often less restrictive than global stepsizes, but also improves the overall communication efficiency in theory and practice.

5 Convergence Analysis

5.1 Assumptions and Definitions

Definition 5.1. A mapping f is L-Lipschitz continuous if $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$, for any x_1, x_2.

Definition 5.2. We call $\bar{x}$ as an ϵ-accurate stationary point of the objective function $\Phi(x)$ if $\mathbb{E}\|\Phi(\bar{x})\|^2 \leq \epsilon$, where $\bar{x}$ is an output of an algorithm.

The following assumption characterizes the geometries of objective functions.

Assumption 5.3. For any $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$ and $i \in \{1, 2, ..., n\}$, $f_i(x, y)$ and $g_i(x, y)$ are twice continuously differentiable, and $g_i(x, y)$ is μ-strongly convex w.r.t. y.

We next impose the Lipschitzness conditions on the function f_i and g_i and their derivatives.

Assumption 5.4. For any $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$ and i, $f_i(x, y)$ is $L_{f,0}$-Lipschitz continuous and $g_i(x, y)$ is $L_{g,0}$-Lipschitz continuous w.r.t. x, the gradients $\nabla f_i(x, y)$ and $\nabla g_i(x, y)$ are $L_{f,1}$ and $L_{g,1}$-Lipschitz continuous respectively, and the second-order derivatives $\nabla^2 f_i(x, y)$ and $\nabla^2 g_i(x, y)$ are $L_{f,2}$ and $L_{g,2}$-Lipschitz continuous respectively.

Note from Theorem 5.4 that we only assume $g_i(x, y)$ is $L_{g,0}$-Lipschitz continuous w.r.t. x rather than y because otherwise it contradicts with the fact that $g_i(x, y)$ is μ-strongly convex w.r.t. y. Next, we assume that the stochastic gradients and second-order derivatives have bounded estimation variances.

Assumption 5.5. $\nabla f_i(x, y; \xi)$ and $\nabla g_i(x, y; \zeta)$ are unbiased estimators of $\nabla f_i(x, y)$ and $\nabla g_i(x, y)$. Furthermore,

there exist constants σ_f and σ_g such that $\mathbb{E}\|\nabla f_i(x,y) - \nabla f_i(x,y;\xi)\|^2 \le \sigma_f^2$, $\mathbb{E}\|\nabla g_i(x,y) - \nabla g_i(x,y;\zeta)\|^2 \le \sigma_g^2$.

The following assumption characterizes the degree of heterogeneity among the individual gradients $\nabla_y g_i(x,y), i \in \{1,...,n\}$.

Assumption 5.6. For any $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$, there exist constants $\beta_{gh} \ge 1$ and $\sigma_{gh} \ge 0$ such that

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\|\nabla_y g_i(x,y)\|^2 \le \beta_{gh}^2\mathbb{E}\|\nabla_y G(x,y)\|^2 + \sigma_{gh}^2.$$

We have $\beta_{gh} = 1$ and $\sigma_{gh} = 0$ when all g_i's are identical.

This assumption uses β_{gh} and σ_{gh} to describe the degree of heterogeneity among the gradients of lower-level objective functions of all clients. Note that we do not directly make the heterogeneity assumption on the target Lagrangian objective $\mathcal{L}$ in eq. (4) because this assumption can be hard to justify due to the dependence of $\mathcal{L}$ on λ that can be sufficiently large. Instead, we impose the heterogeneity assumption only on the lower-level gradients without λ, and then use this assumption to prove the heterogeneity result for $\mathcal{L}$ (see eq. (18) in the appendix for more details).

5.2 Convergence and Complexity Analysis

For simplicity, we let $|\mathcal{C}_t| = P$ for all t. Let $\mathcal{L}^*(x) := \min_y \max_z \mathcal{L}(x,y,z)$ be the Lagrangian function w.r.t. x by taking minimum and maximum over y and z. Since it can be shown that the gradient $\mathcal{L}^*(x)$ has a gap of $\mathcal{O}(\frac{1}{\lambda})$ with the gradient $\nabla\Phi(x)$ of the original problem, we can focus on finding an ϵ-accurate stationary point for λ sufficiently large. Next, we provide an upper bound on $\mathbb{E}\|\nabla\mathcal{L}^*(x_t)\|^2$.

Proposition 5.7. *Under Assumptions 5.4 and 5.5, the iterates of Algorithm 1 satisfy:*

$$\begin{aligned}\mathbb{E}\|\nabla\mathcal{L}^*(x_t)\|^2 \le{}& \mathcal{O}(1/\gamma_x)\cdot\mathbb{E}\big[\mathcal{L}^*(x_t) - \mathcal{L}^*(x_{t+1})\big]\\ &- \mathcal{O}(1)\cdot\mathbb{E}\|\widetilde{h}_x^t\|^2 + \mathcal{O}(\gamma_x)\mathbb{E}\|h_x^t\|^2\\ &+ \mathcal{O}(\lambda^2)\cdot\big(\Delta_x^t + \Delta_y^t + \Delta_z^t\big)\\ &+ \mathcal{O}(\lambda^2)\cdot\mathbb{E}\big(\|z_t - y^*(x_t)\|^2 + \|y_t - y_\lambda^*(x_t)\|^2\big),\end{aligned}$$

where we define $y_\lambda^*(x) := \arg\min_y \mathcal{L}(x,\cdot,z)$, $\widetilde{h}_x^t := \mathbb{E}[h_x^t]$ *and* $\Delta_x^t := \frac{1}{n}\sum_{i=1}^n \frac{1}{\tau}\sum_{k=0}^{\tau-1}\mathbb{E}\|x_i^{t,k} - x_t\|^2$ *(Δ_y^t and Δ_z^t can be defined similarly).*

The proof of Theorem 5.7 refers to that of Theorem C.8 in the appendix. Theorem 5.7 shows that the gradient norm $\mathbb{E}\|\nabla\mathcal{L}^*(x_t)\|^2$ can be effectively bounded by the gap of $\mathcal{L}^*(\cdot)$ at two consecutive iterates, the norm of the generalized gradient h_x^t, the client drifts Δ_x^t, Δ_y^t, Δ_z^t, and the distances of global iterates z_t and y_t to their optimal solutions at each iteration t. Note that Theorem 5.7 also indicates an important trade-off in the selection of the Lagrange multiplier λ. To see this, a large λ guarantees a small gap between the surrogate $\nabla\mathcal{L}^*(\cdot)$ and $\nabla\Phi(\cdot)$ of the original problem, but also exacerbates the client drifts and approximations errors by z_t and y_t. Thus, an appropriate choice of λ is crucial to achieve a good convergence and complexity performance, as shown later in our theorems.

We next provide the following propositions to upper bound the key quantities $\|h_x^t\|$, Δ_x^t, Δ_y^t, Δ_z^t and $\mathbb{E}\|z_t - y^*(x_t)\|^2$, $\|y_t - y_\lambda^*(x_t)\|^2$ in Theorem 5.7, respectively.

Proposition 5.8. *Under Assumptions 5.3, 5.4, 5.5, the client drift of y_t induced by its local updates satisfies:*

$$\begin{aligned}\Delta_y^t \le{}& \mathcal{O}(\eta_y^2\tau\lambda^2)(M_{se} + M_{cs}) + \mathcal{O}(\eta_y^2\tau\lambda^2)\Delta_x^t,\\ &+ \mathcal{O}(\eta_y^2\tau\lambda^2)\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2,\end{aligned}$$

where the factors $M_{se} := \frac{1}{\lambda^2}(\sigma_f^2 + \lambda^2\sigma_g^2)$, $M_{cs} := \frac{1}{\lambda^2}\big(L_{f,0}^2(1 + \frac{2L_{g,1}^2}{\mu^2}) + \lambda^2\sigma_{gh}^2\big)$ *are correlated with variances from gradient estimation and client sampling.*

The proof of Theorem 5.8 can be found in the proof of Theorem C.7 in the appendix. We can see from the above proposition that the client drift w.r.t. y can be bounded by three error terms. The first term $\mathcal{O}(\eta_y^2\tau\lambda^2)(M_{se} + M_{cs})$ is induced by the variances of the stochastic gradient estimators $\nabla f_i(x,y;\xi)$ and $\nabla g_i(x,y;\zeta)$. The second term $\mathcal{O}(\eta_y^2\tau\lambda^2)\Delta_x^t$ correlates to the client drift by the local updates on variable x because the local gradient estimator

$$\begin{aligned}&\nabla_y\mathcal{L}_i(x_i^{t,k}, y_i^{t,k}, z_i^{t,k}; \xi_i^{t,k})\\ &\quad= \nabla_y f_i(x_i^{t,k}, y_i^{t,k}; \xi_i^{t,k}) + \lambda\nabla_y g_i(x_i^{t,k}, y_i^{t,k}; \xi_i^{t,k})\end{aligned}$$

to update $y_i^{t,k}$ also relies on the local iterate $x_i^{t,k}$. The third term $\mathcal{O}(\eta_y^2\tau\lambda^2)\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2$ is associated with the gap between the global iterate y_t and the corresponding optimal point $y_\lambda^*(x_t)$. Note that all these three errors are scaled by the local stepsize η_y^2. Thus, by choosing properly small local stepsizes, we can mitigate the impact of client drifts on the final convergence analysis.

Proposition 5.9. *Under the Assumptions 5.3, 5.4, 5.5, the iterates on y according to Algorithm 1 satisfy*

$$\begin{aligned}&\mathbb{E}\|y_{t+1} - y_\lambda^*(x_{t+1})\|^2 - \mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2\\ &\quad\le -\mathcal{O}(\gamma_y\lambda)\cdot\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2 + \mathcal{O}(\gamma_y^2)\cdot\mathbb{E}\|h_y^t\|^2\\ &\qquad+ \mathcal{O}(\gamma_x^2)\cdot\mathbb{E}\|h_x^t\|^2 + \mathcal{O}\left(\frac{\gamma_x^2}{\gamma_y\lambda}\right)\cdot\mathbb{E}\|\widetilde{h}_x^t\|^2\\ &\qquad+ \mathcal{O}(\gamma_y\lambda)(\Delta_x^t + \Delta_y^t).\end{aligned} \tag{10}$$

The proof of Theorem 5.9 refers to that of Theorem C.10 in the appendix. A similar result can be established for $\mathbb{E}\|z_t - y^*(x_t)\|^2$. It shows that when the stepsize γ_y is properly small, there is a descent in terms of the optimality gap $\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2$, but the bound also contains multiple additional errors by the client drifts Δ_x^t, Δ_y^t, the norm of generalized gradient h_y^t, h_x^t and its expectation $\widetilde{h}_x^t$. The client drifts are controllable to be small based on Theorem 5.8, and the gradient norm $\mathbb{E}\|h_y^t\|$ (similarly for $\mathbb{E}\|h_x^t\|$) can be upper bounded using the following proposition.

Proposition 5.10. *Under Assumptions 5.4 and 5.5, the generalized stochastic gradient h_y^t on y updates as*

$$\begin{aligned}\mathbb{E}\|h_y^t\|^2 \le{}& \mathcal{O}(\lambda^2)\cdot\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2 + \mathcal{O}(\lambda^2)\cdot(\Delta_x^t + \Delta_y^t)\\ &+\mathcal{O}\left(\frac{\lambda^2}{P\tau}\right)M_{se} + \mathcal{O}\left(\frac{(n-P)\lambda^2}{P(n-1)}\right)M_{cs},\end{aligned} \tag{11}$$

where M_{se} and M_{cs} are the same as in Theorem 5.8.

The proof of Theorem 5.10 refers to that of Theorem C.9 in the appendix. Then, by incorporating the bound in Theorem 5.10 into Theorem 5.9, it can be seen that for the stepsize γ_y sufficiently small, the first error term in eq. (11), after being scaled by γ_y^2, can be merged into the negative term $\mathcal{O}(\gamma_y\lambda)\cdot\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2$ in eq. (10) (similarly for the other error terms). Also note that the last two error terms in eq. (11) are caused by data and client sampling, and are decreasing sublinearly w.r.t. the number τ of local steps and the number P of participating clients. This result helps to establish a linear speedup in the convergence and complexity of our algorithm, as shown in the following theorem.

Based on the above propositions, we construct a Lyapunov function $\Psi(x_t) := \mathcal{L}^*(x_t) + K_z\mathbb{E}\|z_t - y^*(x_t)\|^2 + K_y\mathbb{E}\|y_t - y_\lambda^*(x_t)\|^2$ to prove the final convergence result of our algorithm, as shown below.

Theorem 5.11. *Define $\Phi(x) = F\big(x, y^*(x)\big)$. Suppose that Assumptions 5.3, 5.4, 5.5, 5.6 hold. For partial participation, the convergence rate of Algorithm 1 satisfies*

$$\min_t \mathbb{E}\|\Phi(x_t)\|^2 = \mathcal{O}(P^{-\frac{2}{7}}T^{-\frac{2}{7}}).$$

For full participation, the convergence rate of Algorithm 1 satisfies

$$\min_t \mathbb{E}\|\Phi(x_t)\|^2 = \mathcal{O}(P^{-\frac{2}{7}}T^{-\frac{2}{7}}\tau^{-\frac{2}{7}}).$$

In both cases, γ_x, γ_y, γ_z, λ, η_x, η_y, η_z in Algorithm 1 need to be specifically chosen. Their specific choices are given in eq. (47) *and eq.* (48) *in the appendix.*

We can relax the requirement of $P = n$ for full client participation to a milder condition that $P \geq \min\big\{\frac{(\tau-1)n}{\tau} + 1, n\big\}$, and still achieve the same convergence rate. Then, there are a few remarks about the above result. First, our method achieves a linear convergence speedup under both partial and full client participation, where the clients are sampled without replacement in the case of partial client participation. Second, under full client participation, we can ignore the impact of the noise brought by the client sampling on the convergence of our method, and in turn, increasing the number of local updating steps can help to improve the convergence rate.

Next, we analyze the communication and complexity of our proposed algorithm.

Corollary 5.12. *Under the setting of Theorem 5.11, we have the following results:*
Partial participation: *we can find an ϵ-accurate stationary solution of $\Phi(x)$ after $T = \mathcal{O}(P^{-1}\epsilon^{-3.5})$ global iterations, where the communication complexity (which is defined as the number of samples) is $\mathcal{O}(P^{-1}\epsilon^{-3.5})$, and the overall sample complexity (defined as the total number of communication rounds) is $P\tau T = \mathcal{O}(\epsilon^{-3.5})$.*
Full participation: *our method finds an ϵ-stationary solution of $\Phi(x)$ after $T = \mathcal{O}(\epsilon^{-1.5})$ global iterations, where the communication complexity is $\mathcal{O}(\epsilon^{-1.5})$, and overall the sample complexity is $P\tau T = \mathcal{O}(\epsilon^{-3.5})$.*

There are several remarks about Theorem 5.12. First, note that we choose the number τ of local steps differently for the partial and full client participation settings, respectively. This difference is caused by the analysis of the last two error terms in eq. (11) of Theorem 5.10. For partial client participation, we set $\tau = \mathcal{O}(1)$ because further increasing τ cannot improve the final convergence rate due to the dominant error $\mathcal{O}\big(\frac{(n-P)\lambda^2}{P(n-1)}\big)M_{cs}$ induced by client sampling. In contrast, in the case of full client participation, the error $\mathcal{O}\big(\frac{(n-P)\lambda^2}{P(n-1)}\big)M_{cs}$ vanishes, and hence increasing τ to its maximum $\tau = \mathcal{O}(P^{-1}T^{\frac{4}{3}})$ can reduce the overall estimation variance and hence improve the final convergence rate. Third, differently from existing AID- and ITD-based FBO methods that compute a large number of Hessian- and Jacobian-vector products, our method uses only first-order gradient information, while achieving an improved communication complexity of $\mathcal{O}(\epsilon^{-1.5})$ over the typical $\mathcal{O}(\epsilon^{-2})$ result obtained by FedNest (Tarzanagh et al. 2022), FBO-AggITD (Xiao and Ji 2023) and FedMBO (Huang, Zhang, and Ji 2023).

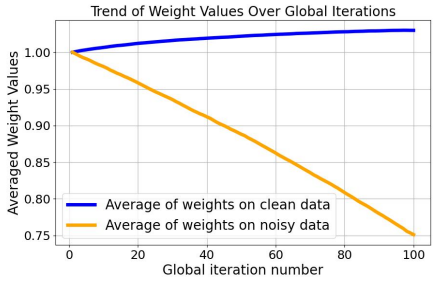

Figure 2: Change of averaged weights on clean/noisy data with the number of global iterations (*noise rate = 50%*).

6 Experiment

We first conduct an experiment on federated data cleaning to compare the performance of our proposed MemFBO algorithms with multiple benchmark personalized federated learning algorithms including FedRep (Collins et al. 2021), FedPer (Arivazhagan et al. 2019), LG-Fed (Liang et al. 2020), and popular federated learning methods including FedAvg (McMahan et al. 2017) and FedProx (Li et al. 2020). We then perform a federated hyper-representation experiment to compare the performance of our method with other FBO algorithms including FedNest (Tarzanagh et al. 2022), LFedNest (Tarzanagh et al. 2022), AggITD (Xiao and Ji 2023), and SimFBO (Yang, Xiao, and Ji 2023b). The former experiment tests the robustness of different methods on the CIFAR10 dataset with CNN backbones, following the same experimental setup as in (Collins et al. 2021). The latter experiment tests the communication and memory efficiency on MNIST with MLP backbones, following the experiment setup in (Tarzanagh et al. 2022). More details of all experiments can be found in Appendix A.

6.1 Robust Federated Data Cleaning

We consider a federated learning setting where edge devices have two data sets: a clean, high-quality set and a noisy,

Method	Noise rate			
	0	30%	50%	70%
FedRep	86.01	70.56	61.68	56.73
FedPer	88.86	74.11	71.04	57.19
LG-Fed	**88.93**	76.27	57.37	49.72
FedAvg	44.86	22.52	16.74	10.34
FedProx	46.30	23.12	17.53	11.93
SimFBO	71.06	65.71	63.27	60.44
MemFBO (ours)	81.33	**78.21**	**77.94**	**73.60**

Table 1: Average test accuracy (%) comparison of different (personalized) federated algorithms with heterogeneous data on the CIFAR10 dataset under different noisy levels.

majority set. To develop a robust FL model, we propose a novel FBO problem where the lower level trains the model on noisy data with learnable sample weights, and the upper level evaluates the model on clean data to refine these weights. This FBO problem can be formulated as:

$$\min_{w\in\mathbb{R}^{m\times n}} \Phi(w) = \frac{1}{n}\sum_{i=1}^{n} f_i\big(y^*(w)\big)$$
$$\text{s.t. } y^*(w) := \arg\min_{y} \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} w_{i,j} g_{i,j}(y),$$

where w contain all weights for data samples of all clients, i is the client index, and j is the data sample index for a client, $g_{i,j}(\cdot)$ denotes the loss on the j_{th} sample from the noisy data set of client i, and $f_i(\cdot)$ denotes the loss on the clean data of client i. Following our first-order approach, the local Lagrangian function then takes the formulation as:

$$\mathcal{L}_i(w_i, y_i, z_i) := f_i(y_i) + \frac{\lambda}{m}\sum_{j=1}^{m} w_{i,j}\big[g_{i,j}(y_i) - g_{i,j}(z_i)\big].$$

Experimental setting. For the baselines, we use 100 clients with the CIFAR10 dataset split into 500 training and 100 test samples per client. To simulate noise, the 500 training samples are divided into 450 noisy samples and 50 clean samples. A subset of the 450 samples is corrupted based on a noise rate (proportion of corrupted data) and a flip rate. In this experiment, the noise rate is set as 0%, 30%, 50%, and 70%. The flip rate decides the portion of labels of a data sample, which are assigned to one of all labels randomly with equal probability. We fix the flip rate to be 80%. Finally, we set up the data heterogeneity following the same procedure as in FedRep (Collins et al. 2021), where each client is assigned with 2 classes.

Results. The results of test accuracy of different (personalized) federated learning methods under different noisy levels are shown in Table 1. The results show the baseline methods suffer from severe performance degradation, while the performance of the two FBO methods is much more robust to data corruption. However, SimFBO performance is notably lower than that of MemFBO. To have a deeper understanding of such robustness, we visualize the averaged weights learned by our FBO method on noisy and clean data for a single client in Figure 2. It clearly shows that the weights associated with the noisy data decrease dramatically, whereas the weights on the clean data increase slightly.

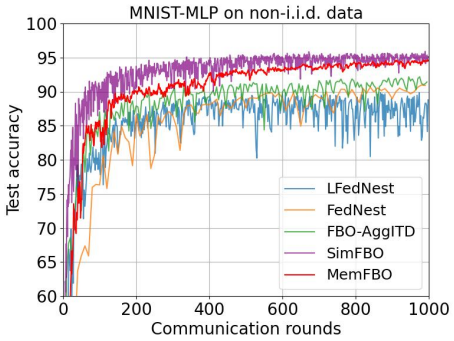

Figure 3: The test accuracy v.s. the number of communication rounds on non-i.i.d. MNIST datasets with MLP networks among our MemFBO and related baselines.

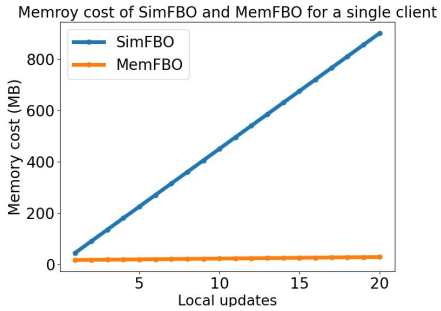

Figure 4: The memory cost comparison between SimFBO and MemFBO.

6.2 Federated Hyper Representation

In this section, we compare the performance of the proposed MemFBO method with other FBO baselines on a hyper-representation problem. We present the comparison results based on test accuracy, communication rounds, and memory cost in Figure 3 and Figure 4. From Figure 3, it can be seen that our method can achieve a faster convergence rate and a higher accuracy than second-order methods with subloops including FedNest, LFedNest, and AggITD, while achieving comparable performance with the fully single-loop second-order method SimFBO. Moreover, Figure 4 shows that the memory cost of SimFBO increases dramatically with the number of local steps, whereas our MemFBO consistently maintains low memory usage.

7 Conclusion

We propose MemFBO, a memory- and computation-efficient federated bilevel algorithm with theoretical convergence under milder assumptions. Experiments validate its efficiency, with extensions to decentralized optimization and applications in hyperparameter tuning, meta-learning, and edge computing.

Acknowledgements

Yifan Yang, Peiyao Xiao and Kaiyi Ji were partially supported by NSF grants CCF-2311274 and ECCS-2326592. Siqian Ma was partially supported by ONR grant N00014-24-1-2705, NSF grant CCF-2311275 and ECCS-2326591.

References

Arbel, M.; and Mairal, J. 2021. Amortized implicit differentiation for stochastic bilevel optimization. *arXiv preprint arXiv:2111.14580*.

Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

Bracken, J.; and McGill, J. T. 1973. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1): 37–44.

Chen, L.; Ma, Y.; and Zhang, J. 2023. Near-Optimal Fully First-Order Algorithms for Finding Stationary Points in Bilevel Optimization. *arXiv preprint arXiv:2306.14853*.

Chen, T.; Sun, Y.; and Yin, W. 2021. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*.

Chen, X.; Huang, M.; and Ma, S. 2022. Decentralized bilevel optimization. *arXiv preprint arXiv:2206.05670*.

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

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting Shared Representations for Personalized Federated Learning. *arXiv preprint arXiv:2102.07078*.

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

Domke, J. 2012. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, 318–326. PMLR.

Dong, Y.; Ma, S.; Yang, J.; and Yin, C. 2023. A Single-Loop Algorithm for Decentralized Bilevel Optimization. *arXiv preprint arXiv:2311.08945*.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Feurer, M.; and Hutter, F. 2019. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, 3–33.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135. PMLR.

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

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

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

Ghadimi, S.; and Wang, M. 2018. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

Gould, S.; Fernando, B.; Cherian, A.; Anderson, P.; Cruz, R. S.; and Guo, E. 2016. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*.

Grazzi, R.; Franceschi, L.; Pontil, M.; and Salzo, S. 2020. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, 3748–3758. PMLR.

Guo, Z.; Hu, Q.; Zhang, L.; and Yang, T. 2021. Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*.

Hansen, P.; Jaumard, B.; and Savard, G. 1992. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5): 1194–1217.

Hao, J.; Ji, K.; and Liu, M. 2023. Bilevel Coreset Selection in Continual Learning: A New Formulation and Algorithm. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hong, M.; Wai, H.-T.; Wang, Z.; and Yang, Z. 2023. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1): 147–180.

Huang, F. 2022. Fast Adaptive Federated Bilevel Optimization. *arXiv preprint arXiv:2211.01122*.

Huang, F.; and Huang, H. 2021. Biadam: Fast adaptive bilevel optimization methods. *arXiv preprint arXiv:2106.11396*.

Huang, M.; Zhang, D.; and Ji, K. 2023. Achieving Linear Speedup in Non-IID Federated Bilevel Learning. *arXiv preprint arXiv:2302.05412*.

Huang, Y.; Lin, Q.; Street, N.; and Baek, S. 2022. Federated learning on adaptively weighted nodes by bilevel optimization. *arXiv preprint arXiv:2207.10751*.

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

Ji, K.; and Ying, L. 2023. Network Utility Maximization with Unknown Utility Functions: A Distributed, Data-Driven Bilevel Optimization Approach. *arXiv preprint arXiv:2301.01801*.

Kwon, J.; Kwon, D.; Wright, S.; and Nowak, R. D. 2023. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, 18083–18113. PMLR.

Li, J.; Gu, B.; and Huang, H. 2020. Improved bilevel model: Fast and optimal algorithm with theoretical guarantee. *arXiv preprint arXiv:2009.00690*.

Li, J.; Huang, F.; and Huang, H. 2022. Local stochastic bilevel optimization with momentum-based variance reduction. *arXiv preprint arXiv:2205.01608*.

Li, J.; Huang, F.; and Huang, H. 2023. Communication-Efficient Federated Bilevel Optimization with Local and Global Lower Level Problems. *arXiv preprint arXiv:2302.06701*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Liang, P. P.; Liu, T.; Ziyin, L.; Allen, N. B.; Auerbach, R. P.; Brent, D.; Salakhutdinov, R.; and Morency, L.-P. 2020. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Liao, R.; Xiong, Y.; Fetaya, E.; Zhang, L.; Yoon, K.; Pitkow, X.; Urtasun, R.; and Zemel, R. 2018. Reviving and improving recurrent back-propagation. In *International Conference on Machine Learning*, 3082–3091. PMLR.

Lin, G.-H.; Xu, M.; and Ye, J. J. 2014. On solving simple bilevel programs with a nonconvex lower level program. *Mathematical Programming*, 144(1-2): 277–305.

Liu, B.; Ye, M.; Wright, S.; Stone, P.; and Liu, Q. 2022. Bome! bilevel optimization made easy: A simple first-order approach. *Advances in Neural Information Processing Systems*, 35: 17248–17262.

Liu, R.; Liu, X.; Yuan, X.; Zeng, S.; and Zhang, J. 2021a. A value-function-based interior-point method for non-convex bi-level optimization. In *International Conference on Machine Learning*, 6882–6892. PMLR.

Liu, R.; Liu, Y.; Zeng, S.; and Zhang, J. 2021b. Towards gradient-based bilevel optimization with non-convex followers and beyond. *Advances in Neural Information Processing Systems*, 34: 8662–8675.

Liu, R.; Mu, P.; Yuan, X.; Zeng, S.; and Zhang, J. 2020. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In *International Conference on Machine Learning*, 6305–6315. PMLR.

Lorraine, J.; Vicol, P.; and Duvenaud, D. 2020. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, 1540–1552. PMLR.

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

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Nesterov, Y.; and Polyak, B. T. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1): 177–205.

Pedregosa, F. 2016. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, 737–746. PMLR.

Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32.

Roh, Y.; Lee, K.; Whang, S. E.; and Suh, C. 2021. FairBatch: Batch Selection for Model Fairness. In *International Conference on Learning Representations*.

Sabach, S.; and Shtern, S. 2017. A first order method for solving convex bilevel optimization problems. *SIAM Journal on Optimization*, 27(2): 640–660.

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

Shen, H.; and Chen, T. 2023. On Penalty-based Bilevel Gradient Descent Method. *arXiv preprint arXiv:2302.05185*.

Shi, C.; Lu, J.; and Zhang, G. 2005. An extended Kuhn–Tucker approach for linear bilevel programming. *Applied Mathematics and Computation*, 162(1): 51–63.

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

Sow, D.; Ji, K.; Guan, Z.; and Liang, Y. 2022. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*.

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

Wang, X.; Pan, R.; Pi, R.; and Zhang, T. 2023. Effective Bilevel Optimization via Minimax Reformulation. *arXiv preprint arXiv:2305.13153*.

Xiao, P.; and Ji, K. 2023. Communication-Efficient Federated Hypergradient Computation via Aggregated Iterative Differentiation. *arXiv preprint arXiv:2302.04969*.

Xing, P.; Lu, S.; Wu, L.; and Yu, H. 2022. BiG-Fed: Bilevel Optimization enhanced graph-aided federated learning. *IEEE Transactions on Big Data*.

Yang, J.; Ji, K.; and Liang, Y. 2021. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34: 13670–13682.

Yang, S.; Zhang, X.; and Wang, M. 2022. Decentralized gossip-based stochastic bilevel optimization over communication networks. *arXiv preprint arXiv:2206.10870*.

Yang, Y.; Xiao, P.; and Ji, K. 2023a. Achieving $\mathcal{O}(\epsilon^{-1.5})$ Complexity in Hessian/Jacobian-free Stochastic Bilevel Optimization. *arXiv preprint arXiv:2312.03807*.

Yang, Y.; Xiao, P.; and Ji, K. 2023b. SimFBO: Towards Simple, Flexible and Communication-efficient Federated Bilevel Learning. *arXiv preprint arXiv:2305.19442*.

Yousefian, F. 2021. Bilevel distributed optimization in directed networks. In *2021 American Control Conference (ACC)*, 2230–2235. IEEE.