Achieving Linear Speedup in Non-IID Federated Bilevel Learning

Minhui Huang [*1] **Dewei Zhang** [*2] **Kaiyi Ji** [3]

Abstract

Federated bilevel learning has received increasing attention in various emerging machine learning and communication applications. Recently, several Hessian-vector-based algorithms have been proposed to solve the federated bilevel optimization problem. However, several important properties in federated learning such as the partial client participation and the linear speedup for convergence (i.e., the convergence rate and complexity are improved linearly with respect to the number of sampled clients) in the presence of non-i.i.d. datasets, still remain open. In this paper, we fill these gaps by proposing a new federated bilevel algorithm named FedMBO with a novel client sampling scheme in the federated hypergradient estimation. We show that FedMBO achieves a convergence rate of $\mathcal{O}\big(\frac{1}{\sqrt{nK}} + \frac{1}{K} + \frac{\sqrt{n}}{K^{3/2}}\big)$ on non-i.i.d. datasets, where n is the number of participating clients in each round, and K is the total number of iteration. This is the first theoretical linear speedup result for non-i.i.d. federated bilevel optimization. Extensive experiments validate our theoretical results and demonstrate the effectiveness of our proposed method.

1. Introduction

Federated learning is a privacy-preserving training paradigm over distributed networks that are designed for edge computing (McMahan et al., 2017). In federated learning, multiple edge devices (or clients) work together to learn a global model under the coordination of a central server. Instead of transmitting user data directly to the central server, each client stores data and computes locally and only transmits the privacy-preserving information. This paradigm is increasingly attractive due to the growing computational power of edge devices and the increasing demand for privacy protection. Federated learning is facing more challenges than traditional distributed optimization due to the high communication cost, data and system heterogeneity, and privacy concerns. Recent years have witnessed great progress in the algorithmic design and system deployment to address such challenges (Wang & Joshi, 2021; Karimireddy et al., 2019; Stich & Karimireddy, 2020).

Recently, federated bilevel learning has received increasing attention (Chen et al., 2018; Fallah et al., 2020; Zeng et al., 2021) because many modern machine learning problems naturally exhibit a bilevel optimization structure. For example, Chen et al. 2018; Fallah et al. 2020 studied the federated meta-learning problems, Khodak et al. 2021 proposed federated hyperparameter optimization approaches, and Zeng et al. 2021 improved the fairness in federated learning using a bilevel method. This motivates us to study the following federated bilevel optimization problem.

$$\min_{x\in\mathbb{R}^p} \Phi(x) = f(x, y^*(x)) := \frac{1}{m}\sum_{i=1}^{m} f_i(x, y^*(x)) \quad \text{(1a)}$$

$$\text{s.t.} \quad y^*(x) \in \arg\min_{y\in\mathbb{R}^q} g(x,y) := \frac{1}{m}\sum_{i=1}^{m} g_i(x,y), \quad \text{(1b)}$$

where $f_i(x,y) = \mathbb{E}f_i(x,y;\xi^i)$, $g_i(x,y) = \mathbb{E}g_i(x,y;\zeta^i)$ are stochastic upper- and lower-level loss functions of client i, and m is the total number of clients. Existing federated learning algorithms like FedAvg and its variants (McMahan et al., 2017) cannot be applied to solve the federated bilevel problem Equation (1) due to the nested optimization structure, the global Hessian inverse estimation in the hypergradient (i.e., $\nabla\Phi(x)$) computation, and the data heterogeneity in both the upper- and lower-level problems.

Recently, several approaches (Tarzanagh et al., 2022; Gao, 2022; Li et al., 2022) have been proposed to efficiently solve Equation (1). Gao 2022; Li et al. 2022 focused on the homogeneous setting and proposed momentum-based distributed bilevel algorithms. In the more practical but challenging heterogeneous setting, Tarzanagh et al. 2022 proposed FedNest based on an implicit differentiation based federated hypergradient estimator. In the inner loop, FedNest calls T times

[*]The first two authors contributed equally, and they are listed according to the alphabetical order of their last names. [1]Department of Electrical and Computer Engineering, University of California, Davis, USA [2]Department of Industrial and Systems Engineering, Ohio State University, Columbus, USA [3]Department of Computer Science and Engineering, University at Buffalo, New York, USA. Correspondence to: Kaiyi Ji <kaiyiji@buffalo.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Table 1. Comparison of FedMBO with existing federated bilevel algorithms. m is the total number of clients, n is the size of sampled clients, and ϵ is the required accuracy. The dependence on κ in LocalBSGVR and AdaFBiO are missing in their papers.

Algorithm	Sample Complexity	Partial Client Participation	Linear Speedup	Data Heterogeneity
LocalBSGVR (Gao, 2022)	$\mathcal{O}(\epsilon^{-3/2}m^{-1})$	✗	✓	✗
AdaFBiO (Huang, 2022)	$\mathcal{O}(\epsilon^{-3/2})$	✗	✗	✓
FedNest (Tarzanagh et al., 2022)	$\mathcal{O}(\kappa^9\epsilon^{-2})$	✗	✗	✓
FedMBO	$\mathcal{O}(\kappa^9\epsilon^{-2}n^{-1})$	✓	✓	✓

of FedInn, which is a federated stochastic variance reduced gradient (FedSVRG) algorithm, to solve the lower-level problem. Then FedNest calls FedOut, which constructs a federated hypergradient estimator, to optimize the upper-level problem. However, FedNest fails to achieve a linear speedup for convergence in training due to the high correlation among the individual hypergradient estimators computed by all clients. In addition, FedNest is restricted to the full client participation. Then, an important question remains:

- *Can we develop an easy-to-implement federated method, which achieves a linear speedup for convergence in the general heterogeneous setting, and allows flexible partial client participation?*

Our contributions. In this paper, we provide an affirmative answer to the above question by proposing a novel federated algorithm called Federated Minibatch Bilevel Optimization (FedMBO). Our contributions are summarized as follows.

- The proposed FedMBO follows a double-loop scheme in bilevel optimization and consists of two important components. For the inner loop, FedMBO adopts a simple Minibatch Stochastic Gradient Descent (SGD) algorithm. Compared with FedAvg and FedSVRG, the minibatch SGD and its accelerated variant are more immune to the heterogeneity of the problem (Woodworth et al., 2020b), which is critical in achieving the linear speedup for convergence under the bilevel optimization structure. For the outer loop, FedMBO features a Parallel Hypergradient Estimator (PHE) with a novel multi-round client sampling scheme. Compared to IHGP (Tarzanagh et al., 2022), our PHE procedure allows either full or partial client participation, and more importantly, achieves a variance bound linearly decreasing w.r.t. the number of participating clients. We anticipate that PHE can be of independent interest to other settings such as decentralized or asynchronous bilevel optimization.

- We show that FedMBO achieves a convergence rate of $\mathcal{O}\big(\frac{1}{\sqrt{nK}} + \frac{1}{K} + \frac{\sqrt{n}}{K^{3/2}}\big)$ and a sample complexity (i.e., the number of samples to achieve an ϵ-stationary point) of $\mathcal{O}(\epsilon^{-2}n^{-1})$, which outperforms that of FedNest (Tarzanagh et al., 2022) by an order of n due to the linear speedup. As shown in Table 1, compared to the momentum-based LocalBSGVR (Gao, 2022) and AdaFBiO (Huang, 2022), our FedMBO is more flexible with partial client participation, and more importantly, achieves the linear speedup for convergence even in the presence of data heterogeneity.

- We conduct extensive experiments to validate our theoretical results, and further demonstrate the effectiveness of our proposed federated hypergradient estimator and the FedMBO algorithm.

1.1. Related Work

Bilevel optimization approaches: Bilevel optimization was first introduced in 1970's (Bracken & McGill, 1973) and has been being studied in the past decades. Since then, tremendous efforts have been made to reformulate the bilevel problem as a single-level optimization problem and develop efficient algorithms to solve it(Aiyoshi & Shimizu, 1984; Edmunds & Bard, 1991; Hansen et al., 1992; Shi et al., 2005). Recently, several prevailing machine learning applications can be naturally formulated as a bilevel programming problem (Maclaurin et al., 2015; Pedregosa, 2016; Finn et al., 2017; Franceschi et al., 2017; 2018; Ji et al., 2020), which brings a lot of attention to the bilevel programming in the machine learning community. On the theoretical side, there are many existing works deriving both asymptotic (Franceschi et al., 2018; Shaban et al., 2019; Liu et al., 2021) and non-asymptotic (Ghadimi & Wang, 2018; Ji et al., 2021; Hong et al., 2020; Chen et al., 2021a; Guo & Yang, 2021; Huang et al., 2022) convergence analysis for the deterministic or stochastic bilevel optimization. For example, Ghadimi & Wang 2018; Hong et al. 2020; Ji et al. 2021; Arbel & Mairal 2022 proved the convergence for SGD type of bilevel methods via the approximate implicit differentiation (AID) approach. Yang et al. 2021; Chen et al. 2021b; Khanduri et al. 2021; Guo & Yang 2021; Dagréou et al. 2022 adopted the variance reduction and momentum techniques into stochastic bilevel programming to achieve better complexity results.

Federated learning: At the core of federated learning is the prevailing FedAvg algorithm and its variants (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2019; Mitra et al., 2021; Acar et al., 2021; Stich, 2018; Yu et al., 2019;

Yang et al., 2020; Qu et al., 2020) to address the communication efficiency and the data privacy concerns. We review literature with a focus on the analysis of the linear speedup for convergence. In the homogeneous setting, two variants of FedAvg were proposed to achieve linear speedup (Stich, 2018; Yu et al., 2019) under the assumptions of bounded gradient and full client participation. Later, Wang & Joshi 2021; Stich & Karimireddy 2020 removed the bounded gradient assumption and established a convergence rate of $\mathcal{O}(\epsilon^{-2}m^{-1})$. In the heterogeneous setting, the SCAFFOLD algorithm (Karimireddy et al., 2019) achieves the first linear speedup convergence rate using a variance reduction framework and is independent of the level of heterogeneity. After that, several variants of FedAvg (Yang et al., 2020; Qu et al., 2020) have also been proved to achieve linear speedup. Another interesting line of work focuses on the comparison between FedAvg and minibatch SGD (Woodworth et al., 2020a;b). In the homogeneous case, FedAvg provably outperforms minibatch SGD and its accelerated versions (Woodworth et al., 2020a). However, when the heterogeneity level is high, FedAvg is shown to be worse than minibatch SGD.

Distributed bilevel optimization: For the decentralized stochastic bilevel optimization (DSBO) problem, Lu et al. 2022; Terashita & Hara 2022 studied the setting where the clients have their own local lower problems and thus the communication for the lower-level part can be saved, and Yang et al. 2022; Chen et al. 2022a;b considered a more general global setup, in which all the clients target solving a global lower-level problem together. Ji & Ying 2023 proposed a distributed bilevel method for learning the best utility surrogate functions for network utility maximization. The most related works to this paper is the FedNest algorithm (Tarzanagh et al., 2022), which achieves a sample complexity of $\mathcal{O}(\epsilon^{-2})$. This result was further improved by the momentum-based federated bilevel algorithms in Gao 2022; Huang 2022. A concurrent work (Xiao & Ji, 2023) proposed an iterative differentiation-based federated bilevel method named FBO-AggITD, which achieves the same sampling complexity as FedNest. Our proposed FedMBO achieves the first linear speedup result in the heterogeneous setting.

2. Definitions and Assumptions

Throughout this paper, we make the following standard assumptions, as typically adopted in bilevel optimization.

Definition 1. *A function $h : \mathbb{R}^{n_1} \mapsto \mathbb{R}^{n_2 \times n_3}$ is Lipschitz continuous with constant L if*

$$\|h(z_1) - h(z_2)\| \leq L \|z_1 - z_2\| \quad \forall z_1, z_2 \in \mathbb{R}^{n_1},$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector or matrix depending on the value of n_3.

Definition 2. *A solution x is ϵ-accurate stationary point if $\mathbb{E}\|\nabla\Phi(x)\|^2 \leq \epsilon$, where x is the output of an algorithm.*

Let $z = (x, y) \in \mathbb{R}^{p+q}$ denotes all parameters.

Assumption 1. *(Lipschitz properties). For all $i \in [m]$: $f_i(z)$, $\nabla f_i(z)$, $\nabla g_i(z)$, $\nabla^2 g_i(z)$ are $\ell_{f,0}$, $\ell_{f,1}$, $\ell_{g,1}$, $\ell_{g,2}$-Lipshitz continuous, respectively.*

Assumption 2. *(Strong convexity) For all $i \in [m] : g_i(x, y)$ is μ_g-strongly convex in y for any fixed $x \in \mathbb{R}^q$.*

Assumption 3. *(Unbiased estimators). For all $i \in [m]$: $\nabla f_i(z;\xi)$, $\nabla g_i(z;\zeta)$, $\nabla^2 g_i(z;\zeta)$ are unbiased estimators of $\nabla f_i(z)$, $\nabla g_i(z)$, $\nabla^2 g_i(z)$, respectively.*

Assumption 4. *(Bounded variances). For all $i \in [m]$: there exist constants σ_f^2, $\sigma_{g,1}^2$, and $\sigma_{g,2}^2$, such that*

$$\mathbb{E}_\xi \|\nabla f_i(z;\xi) - \nabla f_i(z)\|^2] \leq \sigma_f^2,$$
$$\mathbb{E}_\zeta \|\nabla g_i(z;\zeta) - \nabla g_i(z)\|^2 \leq \sigma_{g,1}^2,$$
$$\mathbb{E}_\zeta \left\|\nabla^2 g_i(z;\zeta) - \nabla^2 g_i(z)\right\|^2 \leq \sigma_{g,2}^2.$$

Assumption 5. *There exists a constant σ_g, such that $\mathbb{E}\|\nabla g_i(z) - \nabla g(z)\|^2 \leq \sigma_g^2$, where the expectation $\mathbb{E}$ is taken over the client index i.*

Remark 1. *The assumptions outlined above are quite common and have been broadly adopted in the existing literature. Assumption 1 imposes certain levels of Lipschitz smoothness, which is a standard condition to derive the non-asymptotic convergence in nonconvex optimization. In addition, the Lipshitz continuity of $\nabla^2 g_i(z)$ enables us to control the error between the inverse of the Hessian matrix and its Neumann series-based approximation, as also adopted by other non-asymptotic studies (e.g, Ji et al. 2021; Chen et al. 2021b). Assumption 2 supposes the strong convexity of the lower-level objective g, which triggers the implicit function theorem to guarantee the hypergradient $\nabla\Phi(x)$ to exist and enjoy an explicit form. Assumptions 3 and 4 require the stochastic estimators to be unbiased with bounded variances, and such conditions are widely adopted in the stochastic optimization. Assumption 5, often employed in the analysis for partial client participation in federated learning, controls the disparity between the local gradient $\nabla g_i(z)$ and the global gradient $\nabla g(z)$.*

3. Algorithms

To solve the bilevel problem in Equation (1), the biggest challenge lies in computing the federated hypergradient $\nabla\Phi(x) = (1/m)\sum_{i=1}^m \nabla f_i(x, y^*(x))$, whose explicit form can be obtained as follows via implicit differentiation.

Lemma 1. *Under Assumptions 1 and 2, we have*

$$\nabla f(x, y^*(x)) = \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x)) \times \left[\nabla^2_{yy} g(x, y^*(x))\right]^{-1} \nabla_y f(x, y^*(x)), \quad (2)$$

where $\nabla^2_{yy} g(x,y)$ is defined as the Hessian matrix of g with respect to y and $\nabla^2_{xy} g(x,y)$ is defined as

$$\nabla^2_{xy} g(x,y) := \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial y_1} g(x,y) & \dots & \frac{\partial^2}{\partial x_1 \partial y_q} g(x,y) \\ & \dots & \\ \frac{\partial^2}{\partial x_p \partial y_1} g(x,y) & \dots & \frac{\partial^2}{\partial x_p \partial y_q} g(x,y) \end{bmatrix}.$$

To employ the above lemma, several challenges arise. First, the evaluation of the federated hypergradient in Equation (2) requires the approximation of the minimizer $y^*(x)$ of the lower-level problem, which may introduce a big bias due to the client drift. We propose to use the simple minibatch SGD as the lower-level optimizer, as elaborated in Section 3.1, to mitigate the impact of the lower-level client drift on the final convergence rate. Second, the stochastic approximation of the infeasible Hessian inverse matrix $\left[\nabla^2_{yy} g(x, y^*(x))\right]^{-1} \nabla_y f(x, y^*(x))$ in Lemma 1 often involves the computation of a series of global Hessian-vector products in a nonlinear manner, which complicates the implementation and may introduce a large estimation variance. Third, the federated hypergradient estimation may suffer from a large bias due to both the upper- and lower-level client drifts. In this paper, we propose a new algorithm FedMBO, which contains two main components, i.e., a minibatch SGD based lower-level optimizer and a novel federated hypergradient estimator, to address the above challenges, respectively.

Algorithm 1 Heterogeneous Distributed Minibatch Bilevel Optimization with Partial Clients Participation

Input: full client index set $[m]$, partial clients n participation, batch size S of local SGD at inner loop, initial point (x^0, y^0), $N \in \mathbb{N}^+$
for $k = 0, 1, \dots K-1$ **do**
  $y^{k,0} = y^k$
  for $t = 0, 1, \dots T-1$ **do**
    Sample client index subset $C^{k,t} = \left\{c_1^{k,t}, ..., c_n^{k,t}\right\}$ with $\left|C^{k,t}\right| = n$
    for $i \in [n]$ in parallel **do**
      Sample batch $S_i^{k,t} = \{\xi_{i,0}^{k,t}, \xi_{i,1}^{k,t}, \dots, \xi_{i,S-1}^{k,t}\}$
      Compute $G_i^{k,t} = \frac{1}{S} \sum_{j=0}^{S-1} \nabla g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,j}^{k,t})$
    end for
    $G^{k,t} = \frac{1}{n} \sum_{i=1}^n G_i^{k,t}$
    $y^{k,t+1} = y^{k,t} - \beta_{k,t} G^{k,t}$
  end for
  $y^{k+1} = y^{k,T}$
  $\{\mathcal{H}_i\} = \mathbf{PHE}(x^k, y^{k+1}, N, n)$
  $h = \frac{1}{n} \sum_{i=1}^n \mathcal{H}_i$
  $x^{k+1} = x^k - \alpha_k h$
end for

3.1. Minibatch SGD for Lower-level Updates

To efficiently solve the lower-level problem, one popular approach is FedAvg. Starting from a common initialization, the clients in FedAvg run multiple local SGD updates on its own objective, which are then aggregated to update the inner variable y. However, it has been shown in Tarzanagh et al. 2022 that FedAvg introduces an undesirable hypergradient estimation bias due to the large client drift. This is typically caused by the multiple local updates under the data heterogeneity, where the local iterates of each client can drift away from the global minimum and converge to the minimum of their own local objectives. Thus, they proposed FedLin, as a variant of the variance reduction method FedSVRG (Mitra et al., 2021), to mitigate the impact of the client drift. However, FedLin has a more complex implementation due to the nest SVRG loop, and more importantly, as shown in Tarzanagh et al. 2022, its convergence error induced by the client drift is not linearly decreasing w.r.t. the number of sampled clients, which is one crucial factor in missing the linear speedup in the convergence rate.

Inspired by a recent work (Woodworth et al., 2020b), we use the minibatch SGD as the lower-level solver, where the clients compute their local minibatch stochastic gradients, which are further aggregated for a one-step update on y. In specific, we first sample a subset $C^{k,t} = \left\{c_1^{k,t}, ..., c_n^{k,t}\right\}$ of clients, and each of them draws a local data batch $S_i^{k,t} = \{\xi_{i,0}^{k,t}, \xi_{i,1}^{k,t}, \dots, \xi_{i,S-1}^{k,t}\}$ with $\left|S_i^{k,t}\right| = S$ and computes the local stochastic gradient $\nabla g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,j}^{k,t})$. Then, the server aggregates the gradients as

$$G^{k,t} = \frac{1}{nS} \sum_{i=1}^n \sum_{j=0}^{S-1} \nabla g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,j}^{k,t}),$$

and further run one-step SGD to update $y^{k,t}$ as

$$y^{k,t+1} = y^{k,t} - \beta_{k,t} G^{k,t}.$$

Compared with FedAvg and FedLin, the minibatch SGD admits a simpler implementation, and more importantly, is more resilient to the data heterogeneity by a more aggressive single update at all clients. As will be seen later, minibatch SGD provides a more accurate estimation of the lower-level solution, which is necessary in achieving the linear speedup.

Remark 2. *In the minibatch SGD implementation, we set the batch size to be larger than FedAvg, and hence more aggressive per-iteration progress is made. Thus, the computational cost of minibatch SGD is comparable to FedAvg. More importantly, minibatch SGD admits a much smaller client drift, which is critical in achieving the linear speedup.*

Remark 3. *In the experiments (see Section 5), we demonstrate the great advantages of minibatch SGD over FedAvg in mitigating the client drift during the bilevel training, and in improving the overall communication efficiency.*

3.2. Federated Hypergradient Procedure

In the non-federated setting, one often defines the surrogate

$$\overline{\nabla} f(x,y) = \nabla_x f(x,y) - \nabla^2_{xy} g(x,y)[\nabla^2_{yy} g(x,y)]^{-1} \nabla_y f(x,y) \quad (3)$$

to efficiently approximate the hypergradient $\nabla f(x, y^*(x))$ in Equation (2). Compared with Equation (2), the surrogate simply replaces $y^*(x)$ by its approximation y. A typical approach for efficiently approximating the surrogate is to use the Neumann series-based stochastic estimator.

$$\overline{\nabla} f(x,y) \approx \nabla_x f(x,y;\xi) - \nabla^2_{xy} g\big(x,y,;\zeta^{N'+1}\big) \times \frac{N}{l_{g,1}} \prod_{n=1}^{N'} \Big(I - \frac{1}{l_{g,1}} \nabla^2_{yy} g(x,y;\zeta^n)\Big) \nabla_y f(x,y;\xi) \quad (4)$$

where N' is chosen from $\{0, ..., N-1\}$ uniformly at random and $\{\xi, \zeta^1, ..., \zeta^{N'+1}\}$ are i.i.d. samples. Particularly, Ghadimi & Wang 2018; Hong et al. 2020 show that the inverse Hessian estimation bias exponentially decreases with the number of samples N, i.e.,

$$\Big\| \Big[\nabla^2_{yy} g(x,y)\Big]^{-1} - \mathbb{E}\Big[\frac{N}{\ell_{g,1}} \prod_{n=1}^{N'} \Big(I - \frac{1}{\ell_{g,1}} \nabla^2_{yy} g(x,y;\zeta^n)\Big)\Big]\Big\| \le \frac{1}{\mu_g}\Big(1 - \frac{\mu_g}{\ell_g}\Big)^N \quad (5)$$

where the expectation is taken with respect to both N' and ζ. However, in the federated setting, the computation of the hypergradient is challenging due to client drift by the data heterogeneity, and the computation of a series of global Hessian matrices in a nonlinear manner, as shown in Equation (4). To address such challenges, Tarzanagh et al. 2022 proposed the following federated hypergradient estimator:

$$h_i := \nabla_x f_i(x,y;\xi) - \nabla^2_{xy} g_i(x,y;\zeta^{N'+1}) p_{N'}$$

where the global estimator $p_{N'}$ of the Hessian-inverse-vector product $[\nabla^2_{yy} g(x,y)]^{-1} \nabla_y f(x,y)$ takes the form of

$$p_{N'} = \frac{N}{\ell_{g,1}} \prod_{n=1}^{N'} \Big(I - \frac{1}{\ell_{g,1}\,|S_n|} \sum_{i=1}^{|S_n|} \nabla^2_{yy} g_i(x,y;\zeta_{i,n})\Big) \times \frac{1}{|S_0|} \sum_{i \in S_0} \nabla_y f_i(x,y;\xi_{i,0}),$$

which is constructed by computing and aggregating local Hessian-vector products in N' communication rounds.

However, there are two main limitations of the above federated hypergradient estimator. First, the estimator requires full client participation because each client i needs to compute an h_i. Second, the $h_i's$ are highly correlated due to the shared global estimation $p_{N'}$. As a result, the variance of $\frac{1}{m}\sum_{i=1}^m h_i$ cannot be shown to decay w.r.t. m, which turns out to be the bottleneck for achieving the linear speedup.

Algorithm 2 Parallel Hypergradient Estimator with n Clients Participation (PHE)

1: Sample clients $C^0 = \{c_1^0, ..., c_n^0\}$
2: **for** $i \in [n]$ in parallel **do**
3: Sample data points θ^i, ϕ^i
4: $d_i = \nabla_y f_{c_i^0}(x^k, y^{k+1}; \phi^i)$
5: $p_{i,0} = \frac{N}{l_{g,1}} \nabla_y f_{c_i^0}(x^k, y^{k+1}; \theta^i)$
6: Generate $N_i \in \{0, 1, ..., N-1\}$ UAR
7: **end for**
8: **for** $l = 1, \ldots, \max\{N_i, i \in [n]\}$ **do**
9: Sample $C^l = \{c_1^l, ..., c_n^l\}$
10: **for** $i \in [n]$ in parallel **do**
11: Sample a data point $\zeta^{i,l}$
12: **if** $l \le N_i$ **then**
13: $p_{i,l} = \big(I - \frac{1}{\ell_{g,1}} \nabla^2_{yy} g_{c_i^l}(x^k, y^{k+1}; \zeta^{i,l})\big) p_{i,l-1}$
14: **else**
15: $p_{i,l} = p_{i,l-1}$
16: **end if**
17: **end for**
18: **end for**
19: Sample $C^{\max\{N_i\}+1} = \{c_1^{\max\{N_i\}+1}, ..., c_n^{\max\{N_i\}+1}\}$
20: **for** $i \in [n]$ in parallel **do**
21: Sample a data point ω^i
22: $\mathcal{H}_i = d_i - \nabla^2_{xy} g_{c_i^{\max\{N_i\}+1}}(x^k, y^{k+1}; \omega^i) p_{i,\max\{N_i\}}$
23: **end for**
24: Return $\mathcal{H} = \{\mathcal{H}_i\}_{i \in [n]}$

To this end, we propose a new federated hypergradient estimator with a novel client sampling and communication scheme. As shown by Algorithm 2 and illustrated by Figure 1, each communication round l (highlighted by the yellow shallow in Figure 1) samples n clients ($n \le m$) indexed by $\{c_0^l, ..., c_n^l\}$, and then the sampled clients compute the Hessian-vector product $(I - \frac{1}{\ell_{g,1}} \nabla^2_{yy} g_{c_i^l}(x^k, y^{k+1}; \zeta^{i,l})) p_{i,l-1}$, which are used for the Hessian-vector construction in the next communication round. In the vertical direction of Figure 1 (i.e., from line 8 to line 18 in Algorithm 2), the clients in each column are involved to construct an individual component $\mathcal{H}_i$ of the federated hypergradient estimator. The proposed estimators $\{H_i\}$ take the form of

$$\begin{aligned}\mathcal{H}_i(x^k, y^{k+1}) =& \nabla_x f_{c_i^0}(x^k, y^{k+1}; \phi^i) - \nabla^2_{xy} g_{c_i^{\max\{N_i\}+1}}(x^k, y^{k+1}; \omega^i) \\ &\times \Big[\frac{N}{l_{g,1}} \prod_{l=1}^{N_i} \Big(I - \frac{1}{\ell_{g,1}} \nabla^2_{yy} g_{C_i^l}(x^k, y^{k+1}; \zeta^{i,l})\Big)\Big] \\ &\times \nabla_y f_{c_i^0}(x^k, y^{k+1}; \theta^i). \end{aligned} \quad (6)$$

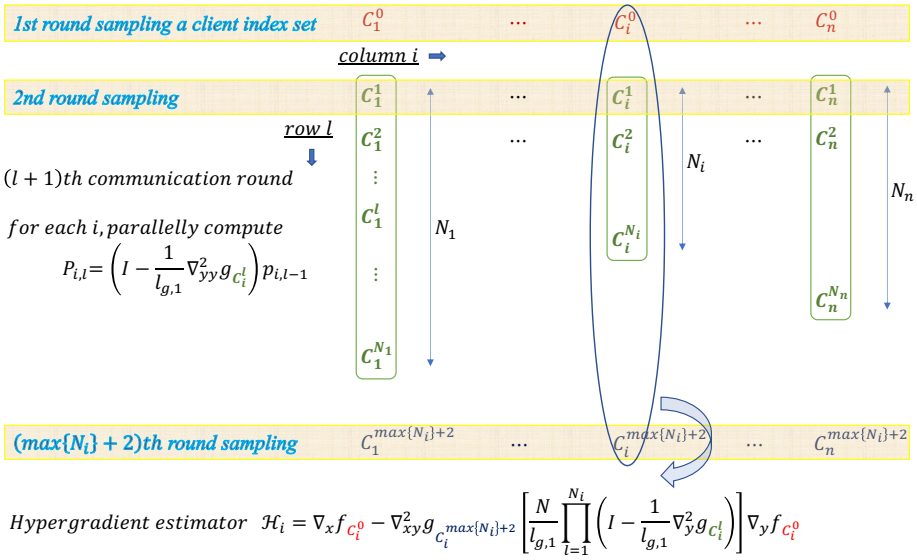

Figure 1. Illustration diagram of client sampling for the federated hypergradient estimation in Algorithm 2.

3.3. Entire Procedure

The previous two sections describe the lower-level updating procedure on y and the federated hypergradient estimator of the proposed FedMBO method. In this section, we briefly summarize the whole algorithm, which is formally described in Algorithm 1. At the beginning of FedMBO, we specify the number of participating clients $n \leq m$, the batch size S for the minibatch SGD implemented at the inner loop, and the constant N controlling the Hessian inverse approximation accuracy. At each round $k = 0, 1, ..., K-1$, FedMBO first runs minibatch SGD to update y, then constructs the federated hypergraident estimator using Algorithm 2, and finally updates the outer variable x based on the hypergradient estimator. We do not run multiple local updates in the updates of x because the federated hypergradient estimator h requires the global information, which is unavailable for local updates of each client.

4. Main Results

As discussed in the previous section, Algorithm 2 generates the federated hypergradient estimators $\{\mathcal{H}_i\}$ for estimating $\nabla f(x, y^*(x))$. With slight abuse of notation, we define $\mathcal{H}_i(x^k, y^{k+1})$ to be the output of Algorithm 2 at the k-th round of Algorithm 1. For different i, j, we have $\mathbb{E}\left[\mathcal{H}_i(x^k, y^{k+1})\right] = \mathbb{E}\left[\mathcal{H}_j(x^k, y^{k+1})\right]$ and

$$\mathbb{E}\left[\mathcal{H}_i(x^k, y^{k+1})|\mathcal{F}^k\right] = \mathbb{E}\left[\mathcal{H}_j(x^k, y^{k+1})|\mathcal{F}^k\right],$$

where $\mathcal{F}^k := \sigma\left\{y^0, x^0, ..., y^k, x^k, y^{k+1}\right\}$ denotes the filtration that captures all the randomness up to the k-th outer loop. We denote $\overline{\mathcal{H}}(x, y) := \mathbb{E}\left[\mathcal{H}_i(x^k, y^{k+1})|\mathcal{F}^k\right]$. Referring to Section 3.2, the resulted $\overline{\mathcal{H}}(x, y)$ is "close" to the surrogate function $\overline{\nabla} f$ defined in Equation (3), except for its matrix inverse approximation. Indeed, the following Proposition 1 shows that the bias between $\overline{\mathcal{H}}(x, y)$ and $\overline{\nabla} f$ decreases exponentially with respect to N.

Proposition 1. *Under Assumptions 1 to 4, we have*

$$\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \overline{\nabla} f(x^k, y^{k+1}))\right\| \leq b,$$

where $b = \kappa_g \ell_{f,1} \left((\kappa_g - 1)/\kappa_g\right)^N$ *and* N *is the input parameter to Algorithm 1.*

The following two propositions explore the bounded variances of $\mathcal{H}_i$. Particularly, the $\mathcal{O}(1/n)$ factor of the bounded variance of the average of $\{\mathcal{H}_i\}_{i\in[n]}$ is presented in Proposition 3. Such a property highly relies on the independence among all the hypergradient estimators and plays an essential role in establishing the linear speedup. This is a key property that can be achieved by our proposed minibatch SGD and PHE algorithms and is missing in FedNest (Tarzanagh et al., 2022) in the non-i.i.d setting.

Proposition 2. *Suppose Assumptions 1 to 4 hold for all* $i \in [n]$. *Then, we have*

$$\mathbb{E}\left[||\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1})||^2\right] \leq \tilde{\sigma}_f^2,$$
$$\mathbb{E}\left[||\mathcal{H}_i(x^k, y^{k+1})||^2|\mathcal{F}^k\right] \leq \tilde{D}_f^2,$$

where the constants $\tilde{\sigma}_f^2$ *and* $\tilde{D}_f^2$ *are given by*

$$\begin{aligned}\tilde{\sigma}_f^2 :=& \sigma_f^2 + \frac{3}{\mu_g^2}\left[(\sigma_f^2 + \ell_{f,0}^2)(\sigma_{g,2}^2 + 2\ell_{g,1}^2) + \sigma_f^2\ell_{g,1}^2\right]\\ =& \mathcal{O}(\kappa_g^2),\\ \tilde{D}_f^2 :=& \left(\ell_{f,0} + \frac{\ell_{f,0}\ell_{g,1}}{\mu_g} + \frac{\ell_{f,1}\ell_{g,1}}{\mu_g}\right)^2 + \tilde{\sigma}_f^2 = \mathcal{O}(\kappa_g^2).\end{aligned}$$

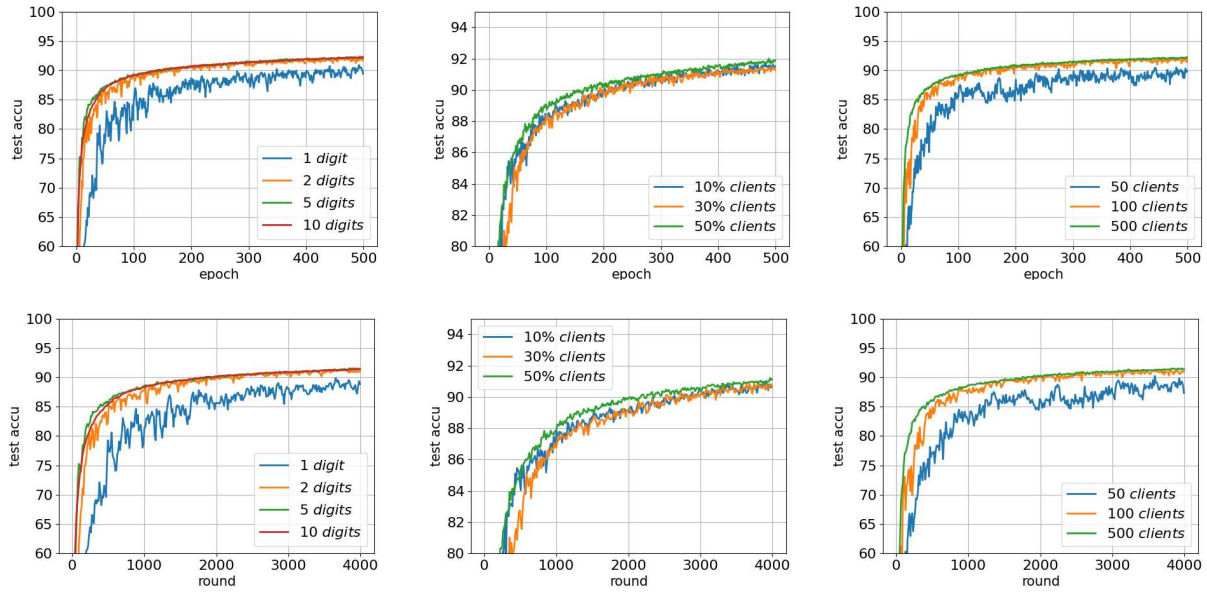

Figure 2. **Left column:** Comparison among different levels of heterogeneity. **Middle column:** Comparison between different numbers of total clients. **Right column:** Comparison among different sampling rates. The first row plots the test accuracy against the epoch. The second plots the test accuracy against the number of communication rounds.

Proposition 3. *Under Assumptions 1 to 4, we have*

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}(\mathcal{H}_i(x^k,y^{k+1})-\overline{\mathcal{H}}(x^k,y^{k+1}))\right\|^2\right]\leq\frac{\tilde{\sigma}_f^2}{n},$$

where $\tilde{\sigma}_f := \sigma_f^2+\frac{3}{\mu_g^2}\left((\sigma_f^2+\ell_{f,0}^2)(\sigma_{g,2}^2+2\ell_{g,1}^2)+\sigma_f^2\ell_{g,1}^2\right)$.

We next characterize the convergence and complexity performance of the proposed algorithm.

Theorem 1. *Suppose Assumptions 1 to 5 hold and set*

$$\alpha_k=\min\left\{\hat{\alpha}_1,\hat{\alpha}_2,\sqrt{\frac{n}{K}}\hat{\alpha}_3\right\}$$

$$\beta_{k,t}=\left(\frac{5M_fL_y}{\mu_g}+\frac{\eta L_{yx}\tilde{D}_f^2\hat{\alpha}_1}{2n\mu_g}\right)\frac{\alpha_k}{T}$$

for some positive constants $\hat{\alpha}_i$, $i=1,2,3$ *independent of* K, *where the definition of the constant parameters* $M_f, L_y, \eta, L_{yx}, \tilde{D}_f^2$ *can be found in the appendix. Then, for any* $K\geq 1$, *the iterates* $\{(x^k,y^k)\}_{k\geq 0}$ *generated by Algorithm 1 satisfy*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]=\mathcal{O}\left(\frac{\hat{\alpha}_3+\hat{\alpha}_3^{-1}}{\sqrt{nK}}+\frac{1}{\min(\hat{\alpha}_1,\hat{\alpha}_2)K}+b^2\right).$$

where $b=\kappa_g l_{f,1}\left(\left(\kappa_g-1\right)/\kappa_g\right)^N$ *and* N *is the controlling input parameter to Algorithm 2,*

Theorem 1 shows that for any given inner loop T, with a proper choice of the step sizes $\alpha_k, \beta_{k,t}$ and hyperparameters, the proposed FedMBO algorithm converges with a sublinear rate. Moreover, the major term in the error bound $\mathcal{O}\left(\frac{1}{\sqrt{nK}}\right)$ has a linear speedup w.r.t. the number n of the participating clients.

Remark 4. *Our theoretical analysis is mainly conducted on the case of partial client participation, i.e.* $n<m$. *For the full clients participation scenario, the analysis is easier and similar results (constants slightly different) can be obtained by following the proof steps in Appendix C.*

Corollary 1. *Under the same conditions as in Theorem 1, if we set* $N=\Omega(\kappa_g\log K)$ *and* $ST=\Omega(\kappa_g^4)$, *then*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]=\mathcal{O}\left(\frac{\kappa_g^{5/2}}{\sqrt{nK}}+\frac{\kappa_g^3}{K}+\frac{\kappa_g^{7/2}\sqrt{n}}{K^{3/2}}\right).$$

In addition, we need $K=\mathcal{O}(\kappa_g^5\epsilon^{-2}/n)$ *to achieve an* ϵ*-accurate stationary point.*

To achieve ϵ-optimal solution, the samples we require in ξ and θ are $\mathcal{O}(\kappa_g^9\epsilon^{-2})$ and $\mathcal{O}(\kappa_g^5\epsilon^{-2})$ respectively. Compared with FedNest (Tarzanagh et al., 2022) in the non-i.i.d. setting, our complexity has the same dependence on κ and ϵ, but a better dependence on n due to the linear speedup. As far as we know, this is the first linear speedup result for non-i.i.d. federated bilevel optimization.

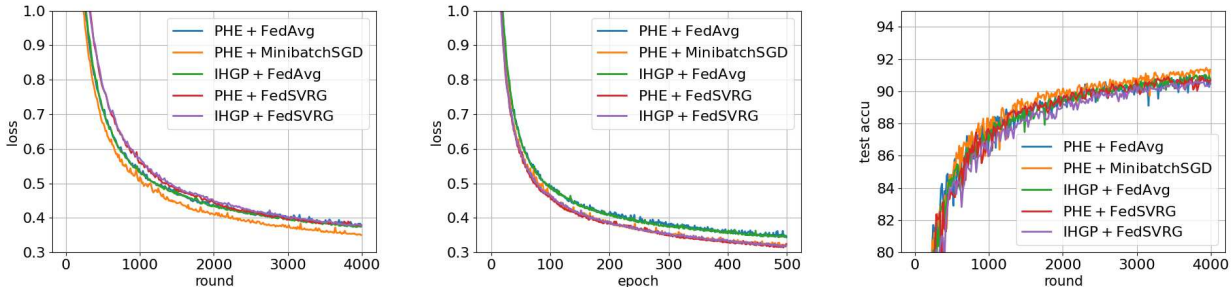

Figure 3. Comparison between our PHE with IHGP (Tarzanagh et al., 2022) under different lower-level optimizers.

.

5. Experiments

In this section, we conduct experiments on hyper-representation, which is an important problem in multi-task machine learning, to validate our theoretical results. We focus on the hyper-representation problem in the federated setting, which can be formulated as

$$\min_{\phi} \mathcal{L}_{\mathcal{D}_v}(\phi, \omega^*) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|\mathcal{D}_v^i|} \sum_{\xi \in \mathcal{D}_v^i} \mathcal{L}(\phi, \omega^*; \xi)$$

$$\text{s.t. } \omega^* = \arg\min_{\omega} \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|\mathcal{D}_t^i|} \sum_{\zeta \in \mathcal{D}_t^i} \mathcal{L}(\phi, \omega; \zeta),$$

where $\mathcal{D}_t^i$ and $\mathcal{D}_v^i$ are the training and the validation datasets respectively. Specifically, the upper-level problem learns the shared hyper feature representations using the validation data, and the lower-level objective learns the prediction head for each client on the training data. In all experiments, we use a multi-layer perceptron (MLP) with 2 linear layers and 1 ReLU activation layer as our model architecture and focus on the heterogeneous case with non-i.i.d. datasets. All experiments are implemented in Python 3.7 on a Linux server with an Nvidia GeForce RTX 2080ti GPU. Note that our current experiments and the results in other related works are all simulations on a single machine. The linear speedup improvement can be shown by implementing the model and the algorithms on a distributed setting with multiple machines.

5.1. Case Study

In this section, we conduct experiments on several case study to demonstrate the efficiency of our proposed algorithm. We first study the impact of heterogeneity in each client's dataset. We fix the client sampling ratio to 10%, and the number of clients to be 100 and sample the dataset in a digit-based manner. In particular, the whole MNIST dataset is split into 10 subsets, where each subset contains all images with the same digit. The data in each client is sampled from a certain number of subsets. In a 2-digit case, for each client, we first randomly pick 2 digits, and then sample data from the images with these two digits. Note that the 10-digit case is equivalent to the homogeneous case. In this way, the number of digits measures the degree of heterogeneity. The result is summarized in the left column of Figure 2. The proposed algorithm performs the worst in the 1-digit case with the highest data heterogeneity, and the performance is improved as we increase the number of digits due to the reduced data heterogeneity. This demonstrates the negative impact of data heterogeneity on convergence performance.

Second, we study the impact of different client sampling ratios. We fix the 2-digit sampling strategy for each client and the total number of clients to be 100. From the middle column of Figure 2, it is seen that the case of 50% client sampling ratio performs the best. Therefore, increasing the sampling ratio helps the performance of our algorithm.

Finally, we test the impact of different numbers of total clients. We fix a 2-digit sampling strategy for each client and the client sampling ratio to be 10%. We select $n \in \{50, 100, 500\}$ for the test. As shown in the right column of Figure 2, the performance of our proposed algorithm becomes better as we increase the number of clients.

5.2. Comparison with FedNest

We compare our approaches with FedNest (Tarzanagh et al., 2022) in the non-i.i.d. setting. We notice that Tarzanagh et al. 2022 also proposed a Lite FedNest (LFedNest) to reduce the communication rounds. However, LFedNest diverges in some of our non-i.i.d experiments and performs worse than the FedNest. So we focus on the comparison of FedNest and our proposed algorithm only. Two major components of the FedNest algorithm are IHGP for estimating the hypergradient and FedSVRG (or FedLin) for solving the lower-level problem. We compare the performance among different pairs of PHE, IHGP, and MinibatchSGD, FedSVRG, FedAvg. In this case, we set the number of total clients to 100 and the sampling ratio to be 10%. For the dataset of each client, we first sort the MNIST dataset according to their labels and then equally split it into 100 subsets and assign one subset to each client. In this way, we guarantee a high-level heterogeneity among all the clients. We set

$T = 5$ for all cases and fine-tune the step sizes so that each setting achieves its best performance.

In Figure 3, we plot the loss and test accuracy against epoch and communication round respectively. The left figure plots the loss against the communication round. From the left figure, we conclude that among all the settings, the proposed PHE + MinibatchSGD converges the fastest. The middle figure plots the loss against data epochs and shows that the MinibatchSGD for the lower-level problem achieves similar performance to FedSVRG and both are better than the FedAvg Algorithm. The right figure shows that PHE + MinibatchSGD achieves the best test accuracy among all algorithms.

6. Conclusion

This paper studies the federated bilevel optimization problem in the presence of data heterogeneity, and proposes a novel federated bilevel algorithm named FedMBO. We show that FedMBO is flexible with partial client participation and achieves a linear speedup for convergence. Numerical experiments are conducted to demonstrate the advantages of our proposed algorithms. We anticipate that our theoretical results and the proposed hypergradient estimator can be applied to other distributed scenarios such as decentralized bilevel optimization.

Acknowledgements

We would like to thank anonymous referees for constructive suggestions that improve the quality of this paper and Jingyu Qian for providing an Nvidia GeForce RTX 2080ti GPU.

References

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Aiyoshi, E. and Shimizu, K. A solution method for the static constrained stackelberg problem via penalty method. *IEEE Transactions on Automatic Control*, 29(12):1111–1114, 1984.

Arbel, M. and Mairal, J. Amortized implicit differentiation for stochastic bilevel optimization. In *International Conference on Learning Representations (ICLR)*, 2022.

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

Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Chen, T., Sun, Y., and Yin, W. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34:25294–25307, 2021a.

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

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

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

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

Edmunds, T. A. and Bard, J. F. Algorithms for nonlinear bilevel mathematical programs. *IEEE transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991.

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

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

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

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

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

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

Guo, Z. and Yang, T. Randomized stochastic variance-reduced methods for stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

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

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

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

Huang, M., Ji, K., Ma, S., and Lai, L. Efficiently escaping saddle points in bilevel optimization. *arXiv preprint arXiv:2202.03684*, 2022.

Ji, K. and Ying, L. Network utility maximization with unknown utility functions: A distributed, data-driven bilevel optimization approach. *arXiv preprint arXiv:2301.01801*, 2023.

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

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

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for on-device federated learning. 2019.

Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34:30271–30283, 2021.

Khodak, M., Tu, R., Li, T., Li, L., Balcan, M.-F. F., Smith, V., and Talwalkar, A. Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing. *Advances in Neural Information Processing Systems*, 34: 19184–19197, 2021.

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

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

Liu, R., Liu, X., Yuan, X., Zeng, S., and Zhang, J. A value-function-based interior-point method for non-convex bilevel optimization. In *International Conference on Machine Learning (ICML)*, 2021.

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

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

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

Mitra, A., Jaafar, R., Pappas, G. J., and Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34:14606–14619, 2021.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Qu, Z., Lin, K., Kalagnanam, J., Li, Z., Zhou, J., and Zhou, Z. Federated learning's blessing: Fedavg has linear speedup. *arXiv preprint arXiv:2007.05690*, 2020.

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

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

Stich, S. U. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates. *Journal of Machine Learning Research*, 21:1–36, 2020.

Tarzanagh, D. A., Li, M., Thrampoulidis, C., and Oymak, S. Fednest: Federated bilevel, minimax, and compositional optimization. *arXiv preprint arXiv:2205.02215*, 2022.

Terashita, N. and Hara, S. Personalized decentralized bilevel optimization over stochastic and directed networks. *arXiv preprint arXiv:2210.02129*, 2022.

Wang, J. and Joshi, G. Cooperative sgd: A unified framework for the design and analysis of local-update sgd algorithms. *Journal of Machine Learning Research*, 22, 2021.

Woodworth, B., Patel, K. K., Stich, S., Dai, Z., Bullins, B., Mcmahan, B., Shamir, O., and Srebro, N. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020a.

Woodworth, B. E., Patel, K. K., and Srebro, N. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33: 6281–6292, 2020b.

Xiao, P. and Ji, K. Communication-efficient federated hypergradient computation via aggregated iterative differentiation. *arXiv preprint arXiv:2302.04969*, 2023.

Yang, H., Fang, M., and Liu, J. Achieving linear speedup with partial worker participation in non-iid federated learning. In *International Conference on Learning Representations*, 2020.

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

Yang, S., Zhang, X., and Wang, M. Decentralized gossip-based stochastic bilevel optimization over communication networks. In *Advances in Neural Information Processing Systems*, 2022.

Yu, H., Yang, S., and Zhu, S. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5693–5700, 2019.

Zeng, Y., Chen, H., and Lee, K. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021.

Supplementary Materials

A. Supporting Lemmas

The following two lemmas are commonly used in the previous literature on (federated) bilevel optimization. We refer to the corresponding works for detailed proofs.

Lemma 2. *([Ghadimi & Wang 2018, Lemma 2.2]) Under Assumptions 1 and 2, we have*

$$||\nabla\Phi(x_1) - \nabla\Phi(x_2)|| \leq L_f||x_1 - x_2||,$$
$$||y^*(x_1) - y^*(x_2)|| \leq L_y||x_1 - x_2||,$$

where

$$L_f := l_{f,1} + \frac{l_{g,1}(l_{f,1} + M_f)}{\mu_g} + \frac{l_{f,0}}{\mu_g}(l_{g,2} + \frac{l_{g,1}l_{g,2}}{\mu_g}) = \mathcal{O}(\kappa_g^3),$$
$$L_y := \frac{l_{g,1}}{\mu_g} = \mathcal{O}(\kappa_g).$$

For all $i \in [m]$, we have

$$||\overline{\nabla} f_i(x_1, y) - \overline{\nabla} f_i(x_1, y^*(x_1))|| \leq M_f||y - y^*(x_1)||,$$
$$||\overline{\nabla} f_i(x_1, y) - \overline{\nabla} f_i(x_2, y)|| \leq M_f||x_1 - x_2||,$$

where the constant M_f is given by

$$M_f := l_{f,1} + \frac{l_{g,1}l_{f,1}}{\mu_g} + \frac{l_{f,0}}{\mu_g}(l_{g,2} + \frac{l_{g,1}l_{g,2}}{\mu_g}) = \mathcal{O}(\kappa_g^2)$$

and $\overline{\nabla} f_i$ is defined as

$$\overline{\nabla} f_i(x, y) := \nabla_x f_i(x, y) - \nabla_{xy}^2 g(x, y)[\nabla_{yy}^2 g(x, y)]^{-1}\nabla_y f_i(x, y).$$

Proof of Lemma 2. The proof is similar to Lemma 2.2 in Ghadimi & Wang 2018. □

Lemma 3. *([Chen et al. 2021a, Lemma 2]) Under Assumptions 1 to 3, we have*

$$||\nabla y^*(x_1) - \nabla y^*(x_2)|| \leq L_{yx}||x_1 - x_2||,$$

where the constant L_{yx} is given by

$$L_{yx} := \frac{l_{g,2} + l_{g,2}L_y}{\mu_g} + \frac{l_{g,1}}{\mu_g^2}(l_{g,2} + l_{g,2}L_y) = \mathcal{O}(\kappa_g^3).$$

Proof of Lemma 3. The proof is similar to Lemma 2 in Chen et al. 2021a. □

B. Proof of Proposition in Section 4

Proof of Proposition 1. The independency of $c_i^0, c_i^{\max\{N_i\}+1}, c_i^\ell, \ell = 1, ..., n, N_i$ and their data points sample is guaranteed based on the algorithmic design in Algorithm 2. Therefore, we have

$$\overline{\mathcal{H}}(x^k, y^{k+1}) := \mathbb{E}\left[\mathcal{H}_i(x^k, y^{k+1})|\mathcal{F}^k\right]$$
$$=\nabla_x f(x^k, y^{k+1}) - \nabla_{xy}^2 g(x^k, y^{k+1})\mathbb{E}\left[\frac{N}{l_{g,1}}\prod_{l=1}^{N_i}\left(I - \frac{1}{\ell_{g,1}}\nabla_{yy}^2 g_{C_i^l}(x^k, y^{k+1}; \zeta^{i,l})\right)\Big|\mathcal{F}^k\right]\nabla_y f(x^k, y^{k+1}).$$

From the definition of $\overline{\nabla} f$, we have

$$\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \overline{\nabla} f(x^k, y^{k+1}))\right\|$$

$$\leq \left\|\nabla^2_{xy} g(x^k, y^{k+1})\right\| \cdot \left\| \mathbb{E}\Big[\frac{N}{l_{g,1}} \prod_{l=1}^{N_i} \Big(I - \frac{1}{\ell_{g,1}} \nabla^2_{yy} g_{C_i^l}(x^k, y^{k+1}; \zeta^{i,l})\Big) \Big| \mathcal{F}^k\Big] - \left[\nabla^2_{yy} g(x,y)\right]^{-1} \right\| \cdot \left\|\nabla_y f(x^k, y^{k+1})\right\|$$
$$\leq \frac{\ell_{f,0}\ell_{g,1}}{\mu_g}\big(1 - \frac{\mu_g}{\ell_g}\big)^N,$$

where we have applied the Assumption 1 and Equation (5) to the last inequality.

Then, the proof is complete. □

Proof of Proposition 2. Following Hong et al. 2020, Lemma 1, we can derive

$$\mathbb{E}\left[||\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1})||^2\right] \leq \tilde{\sigma}_f, \tag{7}$$

where $\tilde{\sigma}_f^2 := \sigma_f^2 + \frac{3}{\mu_g^2}\Big[(\sigma_f^2 + \ell_{f,0}^2)(\sigma_{g,2}^2 + 2\ell_{g,1}^2) + \sigma_f^2 \ell_{g,1}^2\Big]$.

From the definition in Equation (3), we have

$$\begin{aligned}
\left\|\overline{\nabla} f(x,y)\right\| &= \left\|\nabla_x f(x,y) - \nabla^2_{xy} g(x,y)[\nabla^2_{yy} g(x,y)]^{-1} \nabla_y f(x,y)\right\| \\
&\leq \left\|\nabla_x f(x,y)\right\| + \left\|\nabla^2_{xy} g(x,y)\right\| \left\|[\nabla^2_{yy} g(x,y)]^{-1}\right\| \left\|\nabla_y f(x,y)\right\| \\
&\leq \ell_{f,0} + \frac{\ell_{f,0}\ell_{g,1}}{\mu_g},
\end{aligned} \tag{8}$$

where the last inequality comes from Assumptions 1 and 2.

Now we derive the bound of $\mathcal{H}_i(x^k, y^{k+1})$ as follows,

$$\begin{aligned}
\mathbb{E}\left[||\mathcal{H}_i(x^k, y^{k+1})||^2 | \mathcal{F}^k\right] &= \left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2 + \mathbb{E}\left[||\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1})||^2 | \mathcal{F}^k\right] \\
&\leq \left(\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \overline{\nabla} f(x,y)\right\| + \left\|\overline{\nabla} f(x,y)\right\|\right)^2 + \tilde{\sigma}_f^2 \\
&\leq \Big(\kappa_g \ell_{f,1} \left((\kappa_g - 1)/\kappa_g\right)^N + \ell_{f,0} + \ell_{g,1} \frac{1}{\mu_g} \ell_{f,0}\Big)^2 + \tilde{\sigma}_f^2 \\
&\leq \big(\ell_{f,0} + \frac{\ell_{f,0}\ell_{g,1}}{\mu_g} + \frac{\ell_{f,1}\ell_{g,1}}{\mu_g}\big)^2 + \tilde{\sigma}_f^2,
\end{aligned}$$

where the first inequality is based on the result of Equation (7) and the second inequality is based on Proposition 1 and Equation (8).

Then, the proof is complete. □

Proof of Proposition 3. Note that if we choose to sample the clients *with replacement* in Algorithm 2, then apparently $\{\mathcal{H}_i\}$ are pairwise independent random variables (refer to Figure 1). From Proposition 2, we have the variances of $\{\mathcal{H}_i\}_{i=1,\ldots,n}$ are bounded by a constant $\tilde{\sigma}_f^2$. Therefore, we have

$$\begin{aligned}
&\mathbb{E}\Big\|\frac{1}{n}\sum_{i=1}^n (\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1}))\Big\|^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left\|\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2 \\
&+ \frac{1}{n^2} \sum_{1 \leq i \neq j \leq n} \mathbb{E}\Big[\mathbb{E}\Big\langle \mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1}), \mathcal{H}_j(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1})\Big\rangle | \mathcal{F}^k\Big] \\
&\leq \frac{\tilde{\sigma}_f^2}{n},
\end{aligned}$$

where the last inequality follows because $\mathcal{H}_i - \overline{\mathcal{H}}$ and $\mathcal{H}_j - \overline{\mathcal{H}}$ are independent conditioning on $\mathcal{F}^k$.

Then, the proof is complete. □

C. Convergence Proofs

Proof of Lemma 1. This result has been well-known in the literature on bilevel optimization. See, e.g., Ghadimi & Wang 2018 for its proof. □

Lemma 4. *Suppose Assumptions 1 to 4 hold, Algorithm 1 guarantees:*

$$\mathbb{E}\left[f(x^{k+1})\right] - \mathbb{E}[f(x^k)] \le \alpha_k M_f^2 \mathbb{E}\left[\left\|y^{k+1} - y^*(x^k)\right\|^2\right] + (\alpha_k^2 L_f - \frac{\alpha_k}{2})\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] + \alpha_k b^2 + \frac{\alpha_k^2 L_f \tilde{\sigma}_f^2}{n}.$$

Proof of Lemma 4. Based on the smoothness property of the objective function Φ established in Lemma 2, we have

$$\begin{aligned}
&\mathbb{E}\left[f(x^{k+1})\right] - \mathbb{E}\left[f(x^k)\right] \\
\le&\mathbb{E}\left[\left\langle x^{k+1} - x^k, \nabla f(x^k)\right\rangle\right] + \frac{L_f}{2}\mathbb{E}\left[\left\|x^{k+1} - x^k\right\|^2\right] \\
=&-\mathbb{E}\left[\left\langle \frac{1}{n}\sum_{i=1}^n \alpha_k \mathcal{H}_i(x^k, y^{k+1}), \nabla f(x^k)\right\rangle\right] + \frac{L_f}{2}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \alpha_k \mathcal{H}_i(x^k, y^{k+1})\right\|^2\right]. \qquad (10)
\end{aligned}$$

To bound the first term of Equation (10), we have

$$\begin{aligned}
&-\mathbb{E}\left[\left\langle \frac{1}{n}\sum_{i=1}^n \alpha_k \mathcal{H}_i(x^k, y^{k+1}), \nabla f(x^k)\right\rangle\right] \\
=&-\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \alpha_k \mathbb{E}\left[\langle \mathcal{H}_i(x^k, y^{k+1}), \nabla f(x^k)\rangle | \mathcal{F}^k\right]\right] \\
=&-\mathbb{E}\left[\left\langle \alpha_k \overline{\mathcal{H}}(x^k, y^{k+1}), \nabla f(x^k)\right\rangle\right] \\
=&-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] + \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \nabla f(x^k)\right\|^2\right] \\
=&-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] \\
&+\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \overline{\nabla} f(x^k, y^{k+1}) + \overline{\nabla} f(x, y^{k+1}) - \nabla f(x^k)\right\|^2\right] \\
\le&-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] \\
&+\alpha_k\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1}) - \overline{\nabla} f(x^k, y^{k+1}))\right\|^2\right] + \alpha_k\mathbb{E}\left[\left\|\overline{\nabla} f(x^k, y^{k+1}) - \nabla f(x^k)\right\|^2\right] \\
\le&-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] + \alpha_k b^2 + \alpha_k M_f^2 \mathbb{E}\left[\left\|y^{k+1} - y^*(x^k)\right\|^2\right],
\end{aligned}$$

where the last inequality is due to Lemma 2 and Proposition 1. The second term of Equation (10) can be bounded as

$$\begin{aligned}
&\frac{L_f}{2}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n \alpha_k \mathcal{H}_i(x^k, y^{k+1})\right\|^2\right] \\
=&\frac{\alpha_k^2 L_f}{2}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n (\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1}) + \overline{\mathcal{H}}(x^k, y^{k+1}))\right\|^2\right] \\
\le&\alpha_k^2 L_f \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^n (\mathcal{H}_i(x^k, y^{k+1}) - \overline{\mathcal{H}}(x^k, y^{k+1}))\right\|^2\right] + \alpha_k^2 L_f \mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] \\
=&\frac{\alpha_k^2 L_f \tilde{\sigma}_f^2}{n} + \alpha_k^2 L_f \mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right],
\end{aligned}$$

where we use Proposition 3 in the last equality. Combining the above inequalities yields

$$\mathbb{E}[f(x^{k+1})] - \mathbb{E}[f(x^k)] \leq \alpha_k M_f^2 \mathbb{E}\left[\left\|y^{k+1} - y^*(x^k)\right\|^2\right] + (\alpha_k^2 L_f - \frac{\alpha_k}{2})\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k, y^{k+1})\right\|^2\right] - \frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] + \alpha_k b^2 + \frac{\alpha_k^2 L_f \tilde{\sigma}_f^2}{n}.$$

Then, the proof is complete. □

Lemma 5. *Suppose Assumptions 1 to 5 hold and* $0 < \beta_{k,t} \leq \frac{1}{2l_{g,1}}$*, the iterates of Algorithm 1 guarantees:*

$$\mathbb{E}\left[\left\|y^{k+1} - y^*(x^k)\right\|^2\right] \leq \left(\prod_{t=0}^{T-1}(1 - \beta_{k,t}\mu_g)\right)\mathbb{E}\left[\left\|y^k - y^*(x^k)\right\|^2\right] + \frac{4(\sigma_{g,1}^2 + \sigma_g^2)}{nS}\sum_{t=0}^{T-1}\beta_{k,t}^2.$$

Proof of Lemma 5. We first show

$$\mathbb{E}\left\|\frac{1}{nS}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right)\right\|^2 \leq \frac{2(\sigma_{g,1}^2 + \sigma_g^2)}{nS}. \tag{11}$$

By the algorithm update, we have

$$\begin{aligned}
&\mathbb{E}\left\|\frac{1}{nS}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right)\right\|^2 \\
=&\frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right)\right\|^2 \\
&+\frac{1}{n^2}\sum_{1\leq i\neq j\leq n}\mathbb{E}\left\langle\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right), \frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k); \xi_{j,s}^{k,t}\right)\right\rangle \\
=&\frac{1}{n^2S^2}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right)\right\|^2 \\
&+\frac{1}{n^2}\sum_{1\leq i\neq j\leq n}\mathbb{E}\left\langle\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right), \frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k); \xi_{j,s}^{k,t}\right)\right\rangle \\
\leq&\frac{2}{n^2S^2}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right) - \nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k)\right)\right\|^2 \\
&+\frac{2}{n^2S^2}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k)\right) - \nabla_y g\left(x^k, y^*(x^k)\right)\right\|^2 \\
&+\frac{1}{n^2}\sum_{1\leq i\neq j\leq n}\mathbb{E}\left\langle\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right), \frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k); \xi_{j,s}^{k,t}\right)\right\rangle \\
\leq&\frac{2(\sigma_{g,1}^2 + \sigma_g^2)}{nS} \\
&+\frac{1}{n^2}\sum_{1\leq i\neq j\leq n}\mathbb{E}\left\langle\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right), \frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k); \xi_{j,s}^{k,t}\right)\right\rangle,
\end{aligned} \tag{12}$$

where the second equality comes from the pairwise independence between ξ, and the last inequality is due to Assumptions 4 and 5. We next show the second term in Equation (12) equal to zero:

$$\sum_{1\leq i\neq j\leq n}\mathbb{E}\left\langle\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k); \xi_{i,s}^{k,t}\right), \frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k); \xi_{j,s}^{k,t}\right)\right\rangle$$

$$
\begin{aligned}
&= \sum_{1\le i\neq j\le n} \mathbb{E}\left\langle \nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k)\right), \nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k)\right)\right\rangle \\
&= \sum_{1\le i\neq j\le n}\sum_{1\le p\neq q\le m} \mathbb{E}\left[\left\langle \nabla_y g_{c_i^{k,t}}\left(x^k, y^*(x^k)\right), \nabla_y g_{c_j^{k,t}}\left(x^k, y^*(x^k)\right)\right\rangle | c_i^{k,t}=p, c_j^{k,t}=q\right]\cdot \mathbb{P}(c_i^{k,t}=p, c_j^{k,t}=q) \\
&= \sum_{1\le p\neq q\le m} \mathbb{E}\left[\left\langle \nabla_y g_p\left(x^k, y^*(x^k)\right), \nabla_y g_q\left(x^k, y^*(x^k)\right)\right\rangle\right] \sum_{1\le i\neq j\le n} \mathbb{P}(c_i^{k,t}=p, c_j^{k,t}=q) \\
&= \sum_{1\le i\neq j\le n} \mathbb{P}(c_i^{k,t}=1, c_j^{k,t}=2) \sum_{1\le p\neq q\le m} \mathbb{E}\left[\left\langle \nabla_y g_p\left(x^k, y^*(x^k)\right), \nabla_y g_q\left(x^k, y^*(x^k)\right)\right\rangle\right] \\
&\le \sum_{1\le i\neq j\le n} \mathbb{P}(c_i^{k,t}=1, c_j^{k,t}=2)\mathbb{E}\left\|\sum_{p=1}^m \nabla g_p(x^k, y^*(x^k))\right\|^2 = 0,
\end{aligned}
$$

where the third equality is based on the fact that $\mathbb{P}(c_i^{k,t}=p, c_j^{k,t}=q)$ is constant cross different combination (p,q) and the last equality is due the optimality condition of the lower level problem. Next we show that for any $t\in\{0,...,T-1\}$,

$$
\mathbb{E}\left[\left\|y^{k,t+1}-y^*(x^k)\right\|^2\right] \le (1-\beta_{k,t}\mu_g)\mathbb{E}\left[\left\|y^{k,t}-y^*(x^k)\right\|^2\right] + \frac{4\beta_{k,t}^2(\sigma_{g,1}^2+\sigma_g^2)}{nS}.
$$

Note that

$$
\begin{aligned}
&\mathbb{E}\left\|y^{k,t+1}-y^*(x^k)\right\|^2 \\
=&\mathbb{E}\left\|y^{k,t}-\frac{\beta_{k,t}}{n}\sum_{i=1}^n G_i^{k,t}-y^*(x^k)\right\|^2 \\
=&\mathbb{E}\left\|y^{k,t}-y^*(x^k)\right\|^2 - 2\beta_{k,t}\mathbb{E}\left\langle \frac{1}{n}\sum_{i=1}^n G_i^{k,t}, y^{k,t}-y^*(x^k)\right\rangle + \beta_{k,t}^2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_i^{k,t}\right\|^2 \\
=&\mathbb{E}\left\|y^{k,t}-y^*(x^k)\right\|^2 - 2\beta_{k,t}\mathbb{E}\left\langle \nabla_y g(x^k, y^{k,t}), y^{k,t}-y^*(x^k)\right\rangle + \beta_{k,t}^2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_i^{k,t}\right\|^2 \\
\le&(1-\beta_{k,t}\mu_g)\left\|y^{k,t}-y^*(x^k)\right\|^2 - 2\beta_{k,t}\mathbb{E}\left[g(x^k, y^{k,t})-g(x^k, y^*(x^k))\right] + \beta_{k,t}^2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_i^{k,t}\right\|^2,
\end{aligned} \tag{13}
$$

where $G_i^{k,t}=\frac{1}{S}\sum_{s=0}^{S-1}\nabla_y g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,s}^{k,t})$ as defined in Algorithm 1. We use the fact that $G_i^{k,t}$ is an unbiased gradient estimator in the third equality and employ the μ_g-strong convexity of $g(x,y)$ with respect to y in the last inequality. To bound the last term in Equation (13), we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_i^{k,t}\right\|^2 \\
=&\mathbb{E}\left\|\frac{1}{nS}\sum_{i=1}^n\sum_{s=0}^{S-1}\left[\nabla_y g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,s}^{k,t}) - \nabla_y g_{c_i^{k,t}}(x^k, y^*(x^k); \xi_{i,s}^{k,t}) + \nabla_y g_{c_i^{k,t}}(x^k, y^*(x^k); \xi_{i,s}^{k,t})\right]\right\|^2 \\
\le&2\mathbb{E}\left\|\frac{1}{nS}\sum_{i=1}^n\sum_{s=0}^{S-1}\left[\nabla_y g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,s}^{k,t}) - \nabla_y g_{c_i^{k,t}}(x^k, y^*(x^k); \xi_{i,s}^{k,t})\right]\right\|^2 \\
&+ 2\mathbb{E}\left\|\frac{1}{nS}\sum_{i=1}^n\sum_{s=0}^{S-1}\left[\nabla_y g_{c_i^{k,t}}(x^k, y^*(x^k); \xi_{i,s}^{k,t})\right]\right\|^2 \\
\le&\frac{2}{nS}\sum_{i=1}^n\sum_{s=0}^{S-1}\mathbb{E}\left\|\nabla_y g_{c_i^{k,t}}(x^k, y^{k,t}; \xi_{i,s}^{k,t}) - \nabla_y g_{c_i^{k,t}}(x^k, y^*(x^k); \xi_{i,s}^{k,t})\right\|^2 + \frac{4(\sigma_{g,1}^2+\sigma_g^2)}{nS}
\end{aligned}
$$

$$
\begin{aligned}
\leq & \frac{4l_{g,1}}{nS}\sum_{i=1}^{n}\sum_{s=0}^{S-1}\mathbb{E}\left[g_{c_i^{k,t}}(x^k,y^{k,t};\xi_{i,s}^{k,t})-g_{c_i^{k,t}}(x^k,y^*(x^k);\xi_{i,s}^{k,t})-\left\langle\nabla_y g_{c_i^{k,t}}(x^k,y^*(x^k);\xi_{i,s}^{k,t}),y^{k,t}-y^*(x^k)\right\rangle\right] \\
& + \frac{4(\sigma_{g,1}^2+\sigma_g^2)}{nS} \\
= & 4l_{g,1}\mathbb{E}\left[g\left(x^k,y^{k,t}\right)-g\left(x^k,y^*(x^k)\right)\right]+\frac{4(\sigma_{g,1}^2+\sigma_g^2)}{nS},
\end{aligned}
$$

where the second inequality uses the previous result of Equation (11) and the third inequality uses Lemma 1 in (Woodworth et al., 2020b). Plugging this into Equation (13) and enforcing $\beta_{k,t}\leq\frac{1}{2l_{g,1}}$ yield

$$
\begin{aligned}
& \mathbb{E}\left\|y^{k,t+1}-y^*(x^k)\right\|^2 \\
\leq & (1-\beta_{k,t}\mu_g)\left\|y^{k,t}-y^*(x^k)\right\|^2+2\beta_{k,t}(2\beta_{k,t}l_{g,1}-1)\mathbb{E}\left[g(x^k,y^{k,t})-g(x^k,y^*(x^k))\right]+\frac{4\beta_{k,t}^2(\sigma_{g,1}^2+\sigma_g^2)}{nS} \\
\leq & (1-\beta_{k,t}\mu_g)\left\|y^{k,t}-y^*(x^k)\right\|^2+\frac{4\beta_{k,t}^2(\sigma_{g,1}^2+\sigma_g^2)}{nS}. \qquad (14)
\end{aligned}
$$

Applying recursion on Equation (14), we obtain

$$
\mathbb{E}\left[\left\|y^{k,T}-y^*(x^k)\right\|^2\right]\leq\left(\prod_{t=0}^{T-1}(1-\beta_{k,t}\mu_g)\right)\mathbb{E}\left[\left\|y^{k,0}-y^*(x^k)\right\|^2\right]+\frac{4(\sigma_{g,1}^2+\sigma_g^2)}{nS}\sum_{t=0}^{T-1}\beta_{k,t}^2,
$$

which completes the proof. □

Remark 5. *In the case of full client participation and the clients are sampled without replacement, from the similar analysis above, we have*

$$
\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]\leq\left(\prod_{t=0}^{T-1}(1-\beta_{k,t}\mu_g)\right)\mathbb{E}\left[\left\|y^{k}-y^*(x^k)\right\|^2\right]+\frac{4\sigma_{g,1}^2}{nS}\sum_{t=0}^{T-1}\beta_{k,t}^2.
$$

Especially Assumption 5 is released for this scenario.

Lemma 6. *Suppose Assumptions 1 to 4 hold, Algorithm 1 guarantees:*

$$
\begin{aligned}
\mathbb{E}\left[\left\|y^{k+1}-y^*(x^{k+1})\right\|^2\right]\leq & a_1(\alpha_k)\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right] \\
& + a_2(\alpha_k,n)\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+a_3(\alpha_k,n)\tilde{\sigma}_f^2,
\end{aligned}
$$

where

$$
\begin{aligned}
a_1(\alpha_k) &:= L_y^2\alpha_k^2+\frac{L_y\alpha_k}{4M_f}+\frac{L_{yx}\alpha_k^2}{2\eta}, \\
a_2(\alpha_k,n) &:= 1+4M_fL_y\alpha_k+\frac{\eta L_{yx}\tilde{D}_f^2\alpha_k^2}{2}, \\
a_3(\alpha_k,n) &:= \frac{\alpha_k^2L_y^2}{n}+\frac{L_{yx}\alpha_k^2}{2\eta n},
\end{aligned}
$$

for any $\eta>0$.

Proof of Lemma 6. Note that

$$
\begin{aligned}
\mathbb{E}\left[\left\|y^{k+1}-y^*(x^{k+1})\right\|^2\right] = & \mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+\mathbb{E}\left[\left\|y^*(x^{k+1})-y^*(x^k)\right\|^2\right] & (15a) \\
& +2\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),y^*(x^k)-y^*(x^{k+1})\rangle\right]. & (15b)
\end{aligned}
$$

To bound the second term of Equation (15a), we have

$$\begin{aligned}\mathbb{E}\left[\left\|y^*(x^{k+1})-y^*(x^k)\right\|^2\right] \leq & L_y^2\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2\right]\\ \leq & L_y^2\mathbb{E}\left[\left\|\alpha_k\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right]+\frac{\alpha_k^2L_y^2\tilde{\sigma}_f^2}{n}.\end{aligned}$$

To bound Equation (15b), we have

$$\begin{aligned}&\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),y^*(x^k)-y^*(x^{k+1})\rangle\right]\\ &=-\mathbb{E}[\langle y^{k+1}-y^*(x^k),\nabla y^*(x^k)(x^{k+1}-x^k)\rangle] \qquad (16a)\\ &-\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),y^*(x^{k+1})-y^*(x^k)-\nabla y^*(x^k)(x^{k+1}-x^k)\rangle\right]. \qquad (16b)\end{aligned}$$

After plugging the updating step of x^k into Equation (16a), we have

$$\begin{aligned}&-\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),\nabla y^*(x^k)(x^{k+1}-x^k)\rangle\right]\\ &=-\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),\alpha_k\nabla y^*(x^k)\overline{\mathcal{H}}(x^k,y^{k+1})\rangle\right]\\ &\leq\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|\left\|\alpha_k\nabla y^*(x^k)\overline{\mathcal{H}}(x^k,y^{k+1})\right\|\right]\\ &\leq L_y\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|\left\|\alpha_k\overline{\mathcal{H}}(x^k,y^{k+1})\right\|\right]\\ &\overset{(a)}{\leq}2\gamma\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+\frac{L_y^2\alpha_k^2}{8\gamma}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right]\\ &\leq 2M_fL_y\alpha_k\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+\frac{L_y\alpha_k}{8M_f}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right],\end{aligned}$$

where Young's inequality is applied in the inequality (a), and the last inequality comes from setting $\gamma = M_fL_y\alpha_k$. Equation (16b) can be further bounded as follows,

$$\begin{aligned}&-\mathbb{E}\left[\langle y^{k+1}-y^*(x^k),y^*(x^{k+1})-y^*(x^k)-\nabla y^*(x^k)(x^{k+1}-x^k)\rangle\right]\\ &\leq\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|\left\|y^*(x^{k+1})-y^*(x^k)-\nabla y^*(x^k)(x^{k+1}-x^k)\right\|\right]\\ &\leq\frac{L_{yx}}{2}\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|\left\|x^{k+1}-x^k\right\|^2\right]\\ &\leq\frac{\eta L_{yx}}{4}\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\left\|x^{k+1}-x^k\right\|^2\right]+\frac{L_{yx}}{4\eta}\mathbb{E}\left[\left\|x^{k+1}-x^k\right\|^2\right]\\ &\leq\frac{\eta L_{yx}\tilde{D}_f^2\alpha_k^2}{4}\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+\frac{L_{yx}\alpha_k^2}{4\eta}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right]+\frac{L_{yx}\alpha_k^2\tilde{\sigma}_f^2}{4\eta n},\end{aligned}$$

where Proposition 2 is applied in the last inequality.

Combining and rearranging the above inequalities complete the proof. □

Proof of Theorem 1. Motivated by Chen et al. 2021a; Tarzanagh et al. 2022, we define the following Lyapunov function

$$W^k := f(x^k)+\frac{M_f}{L_y}\left\|y^k-y^*(x^k)\right\|^2.$$

The difference between the two Lyapunov functions is bounded as

$$W^{k+1}-W^k = f(x^{k+1})-f(x^k)+\frac{M_f}{L_y}(\left\|y^{k+1}-y^*(x^{k+1})\right\|^2-\left\|y^k-y^*(x^k)\right\|^2).$$

From Lemma 4 and Lemma 6, we obtain

$$\mathbb{E}\left[W^{k+1}\right]-\mathbb{E}\left[W^k\right]\leq\alpha_kb^2+\frac{\alpha_k^2L_f\tilde{\sigma}_f^2}{n}+\alpha_kM_f^2\mathbb{E}[\left\|y^{k+1}-y^*(x^k)\right\|^2]$$

$$
\begin{aligned}
&+(\alpha_k^2 L_f-\frac{\alpha_k}{2})\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right]-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]\\
&+\frac{a_1(\alpha_k)M_f}{L_y}\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right]\\
&+\frac{a_2(\alpha_k,n)M_f}{L_y}\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]+\frac{a_3(\alpha_k,n)M_f\tilde{\sigma}_f^2}{L_y}\\
&-\frac{M_f}{L_y}\mathbb{E}\left[\left\|y^k-y^*(x^k)\right\|\right]\\
=&\alpha_k b^2+(\frac{\alpha_k^2 L_f}{n}+\frac{a_3(\alpha_k,n)M_f}{L_y})\tilde{\sigma}_f^2-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]\\
&+(\alpha_k^2 L_f-\frac{\alpha_k}{2}+\frac{a_1(\alpha_k)M_f}{L_y})\mathbb{E}\left[\left\|\overline{\mathcal{H}}(x^k,y^{k+1})\right\|^2\right] \qquad (17a)\\
&+(\alpha_k M_f^2+\frac{a_2(\alpha_k,n)M_f}{L_y})\mathbb{E}\left[\left\|y^{k+1}-y^*(x^k)\right\|^2\right]-\frac{M_f}{L_y}\mathbb{E}\left[\left\|y^k-y^*(x^k)\right\|\right]. \qquad (17b)
\end{aligned}
$$

Note that $Equation$ (17a) ≤ 0 if

$$
\alpha_k \leq \hat{\alpha}_1 := \frac{1}{2L_f+4M_f L_y+\frac{2M_f L_{yx}}{L_y\eta}} \tag{18}
$$

We enforce $\alpha_k \leq \hat{\alpha}_1$ in the following context.

Based on Lemma 5, Equation (17b) can be further bounded as

$$
\begin{aligned}
Equation\ (17b) \leq& 4(\alpha_k M_f^2+\frac{a_2(\alpha_k,n)M_f}{L_y})\frac{T\beta_{k,t}^2(\sigma_{g,1}^2+\sigma_g^2)}{nS}\\
&+\frac{M_f}{L_y}\left((M_f L_y\alpha_k+a_2(\alpha_k,n))\left(1-\beta_{k,t}\mu_g\right)^T-1\right)\mathbb{E}\left[\left\|y^k-y^*(x^k)\right\|\right]. \qquad (19)
\end{aligned}
$$

If $\beta_{k,t} \leq \frac{1}{\mu_g}$, Equation (19) is nonpositive if

$$
\begin{aligned}
&(1+5M_f L_y\alpha_k+\frac{\eta L_{yx}\tilde{D}_f^2\alpha_k^2}{2})(1-\beta_{k,t}\mu_g)^T-1\leq 0\\
\Leftarrow\ &5M_f L_y\alpha_k+\frac{\eta L_{yx}\tilde{D}_f^2\alpha_k^2}{2}\leq T\beta_{k,t}\mu_g\\
\Leftarrow\ &\beta_{k,t}\geq\left(\frac{5M_f L_y}{\mu_g}+\frac{\eta L_{yx}\tilde{D}_f^2\alpha_k}{2\mu_g}\right)\frac{\alpha_k}{T}
\end{aligned}
$$

For simplicity, we remove the subscript t from $\beta_{k,t}$ and enforce

$$
\beta_k=\bar{\beta}\frac{\alpha_k}{T}
$$

where

$$
\bar{\beta}:=\frac{5M_f L_y}{\mu_g}+\frac{\eta L_{yx}\tilde{D}_f^2\hat{\alpha}_1}{2\mu_g}, \tag{20}
$$

which will imply another requirement on α_k since β_k should be less than $\frac{1}{2l_{g,1}}$ as a condition of Lemma 5, i.e.,

$$
\alpha_k\leq\hat{\alpha}_2:=\frac{T}{2l_{g,1}\bar{\beta}}. \tag{21}
$$

After rearranging, we obtain

$$
\mathbb{E}\left[W^{k+1}\right]-\mathbb{E}\left[W^k\right]\leq\alpha_k b^2+\left(\frac{\alpha_k^2 L_f}{n}+\frac{a_3(\alpha_k,n)M_f}{L_y}\right)\tilde{\sigma}_f^2-\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right]
$$

$$+ 4\left(\alpha_k M_f^2 + \frac{a_2(\alpha_k, n) M_f}{L_y}\right) \frac{T\beta_k^2(\sigma_{g,1}^2 + \sigma_g^2)}{nS}.$$

Then telescoping gives

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] \leq & \frac{2}{\sum_{k=0}^{K-1}\alpha_k}(\triangle_w) + 2b^2 + \frac{2}{\sum_{k=0}^{K-1}\alpha_k}\sum_{k=0}^{K-1}\left(\frac{\alpha_k^2 L_f}{n} + \frac{a_3(\alpha_k, n)M_f}{L_y}\right)\tilde{\sigma}_f^2 \\ & + \frac{8}{\sum_{k=0}^{K-1}\alpha_k}\sum_{k=0}^{K-1}\left(\alpha_k M_f^2 + \frac{a_2(\alpha_k, n)M_f}{L_y}\right)\frac{T\beta_k^2(\sigma_{g,1}^2 + \sigma_g^2)}{nS},\end{aligned}$$

where $\triangle_w := W^0 - \mathbb{E}\left[W^K\right]$. We enforce $\alpha_k \leq \sqrt{\frac{n}{K}}\hat{\alpha}_3$ for some positive constant $\hat{\alpha}_3$, which implies

$$\frac{2}{\sum_{k=0}^{K-1}\alpha_k}(\triangle_w) = \mathcal{O}\left(\frac{1}{\min(\hat{\alpha}_1, \hat{\alpha}_2)K} + \frac{1}{\hat{\alpha}_3\sqrt{nK}}\right) \tag{22a}$$

$$\begin{aligned}\frac{2}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\left(\frac{\alpha_k^2 L_f}{n} + \frac{a_3(\alpha_k, n)M_f}{L_y}\right)\tilde{\sigma}_f^2 &= \mathcal{O}\left(\frac{2}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\frac{\alpha_k^2}{n}\right) \\ &= \mathcal{O}\left(\frac{\hat{\alpha}_3}{\sqrt{nK}}\right)\end{aligned} \tag{22b}$$

$$\begin{aligned}\frac{8}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\left(\alpha_k M_f^2 + \frac{a_2(\alpha_k, n)M_f}{L_y}\right)\frac{T\beta_k^2(\sigma_{g,1}^2 + \sigma_g^2)}{nS} &= \mathcal{O}\left(\frac{4}{\sum_{k=0}^{K-1}\alpha_k}\sum_{k=0}^{K-1}\frac{\alpha_k^2}{STn} + \frac{\alpha_k^3}{STn} + \frac{\alpha_k^4}{STn}\right) \\ &= \mathcal{O}\left(\frac{\hat{\alpha}_3}{ST\sqrt{nK}} + \frac{\hat{\alpha}_3^2}{STK} + \frac{\sqrt{n}\hat{\alpha}_3^3}{STK^{3/2}}\right)\end{aligned} \tag{22c}$$

Therefore, we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] = \mathcal{O}\left(\frac{\hat{\alpha}_3 + \hat{\alpha}_3^{-1}}{\sqrt{nK}} + \frac{1}{\min(\hat{\alpha}_1, \hat{\alpha}_2)K} + b^2\right).$$

Then, the proof is complete. □

Proof of Corollary 1. Enforcing $\eta = \frac{M_f}{L_y}$ in Equation (18), Equation (21) and Equation (20), yields $\hat{\alpha}_1 = \mathcal{O}(\kappa_g^{-3})$, $\hat{\alpha}_2 = \mathcal{O}(T\kappa_g^{-3})$ and $\bar{\beta} = \mathcal{O}(\kappa^4)$.

Expanding Equation (22), we have

$$\begin{aligned}&\frac{2}{\sum_{k=0}^{K-1}\alpha_k}(\triangle_w) = \mathcal{O}(\frac{1}{\min(\hat{\alpha}_1, \hat{\alpha}_2)K} + \frac{1}{\hat{\alpha}_3\sqrt{nK}}) \\ &\frac{2}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}(\frac{\alpha_k^2 L_f}{n} + \frac{a_3(\alpha_k, n)M_f}{L_y})\tilde{\sigma}_f^2 \\ &= \mathcal{O}\left(\frac{2}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\frac{\kappa_g^3\alpha_k^2}{n} + \frac{\kappa_g^5\alpha_k^2}{n}\right) = \mathcal{O}\left(\frac{\kappa_g^5\hat{\alpha}_3}{\sqrt{nK}}\right) \\ &\frac{8}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}(\alpha_k M_f^2 + \frac{a_2(\alpha_k, n)M_f}{L_y})\frac{T\beta_k^2(\sigma_{g,1}^2 + \sigma_g^2)}{nS} \\ &= \mathcal{O}\left(\frac{4}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\frac{\eta\bar{\beta}^2}{nST}\alpha_k^2 + \left(\frac{M_f^2\bar{\beta}^2}{nST} + \frac{\eta M_f L_y\bar{\beta}^2}{nST}\right)\alpha_k^3 + \frac{\eta^2\bar{\beta}^2 L_{yz}\tilde{D}_f^2}{nST}\alpha_k^4\right) \\ &= \mathcal{O}\left(\frac{4}{\sum_{k=0}^{K-1}\alpha_k}\cdot\sum_{k=0}^{K-1}\frac{\kappa_g^9}{nST}\alpha_k^2 + \frac{\kappa_g^{12}}{nST}\alpha_k^3 + \frac{\kappa_g^{15}}{nST}\alpha_k^4\right)\end{aligned}$$

$$= \mathcal{O}\left(\frac{\kappa_g^9}{ST\sqrt{nK}}\hat{\alpha}_3 + \frac{\kappa_g^{12}}{STK}\hat{\alpha}_3^2 + \frac{\kappa_g^{15}\sqrt{n}}{STK^{3/2}}\hat{\alpha}_3^3\right)$$

After enforcing $ST = \Omega(\kappa^4)$, $\hat{\alpha}_3 = \mathcal{O}\left(\kappa_g^{-5/2}\right)$, and $N = \Omega(\kappa_g \log K)$, which implies $b = \frac{1}{K^{1/4}}$, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\|\nabla f(x^k)\right\|^2\right] = \mathcal{O}\left(\frac{\kappa_g^{5/2}}{\sqrt{nK}} + \frac{\kappa_g^3}{K} + \frac{\kappa_g^{7/2}\sqrt{n}}{K^{3/2}}\right).$$

Then, the proof is complete. □