FEDNEST: Federated Bilevel, Minimax, and Compositional Optimization

Davoud Ataee Tarzanagh [1] **Mingchen Li** [2] **Christos Thrampoulidis** [3] **Samet Oymak** [2]

Abstract

Standard federated optimization methods successfully apply to stochastic problems with *single-level* structure. However, many contemporary ML problems – including adversarial robustness, hyperparameter tuning, actor-critic – fall under nested bilevel programming that subsumes minimax and compositional optimization. In this work, we propose FEDNEST: A federated alternating stochastic gradient method to address general nested problems. We establish provable convergence rates for FEDNEST in the presence of heterogeneous data and introduce variations for bilevel, minimax, and compositional optimization. FEDNEST introduces multiple innovations including federated hypergradient computation and variance reduction to address inner-level heterogeneity. We complement our theory with experiments on hyperparameter & hyper-representation learning and minimax optimization that demonstrate the benefits of our method in practice.

1. Introduction

In the federated learning (FL) paradigm, multiple clients cooperate to learn a model under the orchestration of a central server (McMahan et al., 2017) without directly exchanging local client data with the server or other clients. The locality of data distinguishes FL from traditional distributed optimization and also motivates new methodologies to address heterogeneous data across clients. Additionally, cross-device FL across many edge devices presents additional challenges since only a small fraction of clients participate in each round, and clients cannot maintain *state* across rounds (Kairouz et al., 2019).

Traditional distributed SGD methods are often unsuitable in FL and incur high communication costs. To overcome this issue, popular FL methods, such as FEDAVG (McMahan et al., 2017), use local client updates, i.e. clients update their models multiple times before communicating with the server (aka, local SGD). Although FEDAVG has seen great success, recent works have exposed convergence issues in certain settings (Karimireddy et al., 2020; Hsu et al., 2019). This is due to a variety of factors, including *client drift*, where local models move away from globally optimal models due to objective and/or systems heterogeneity.

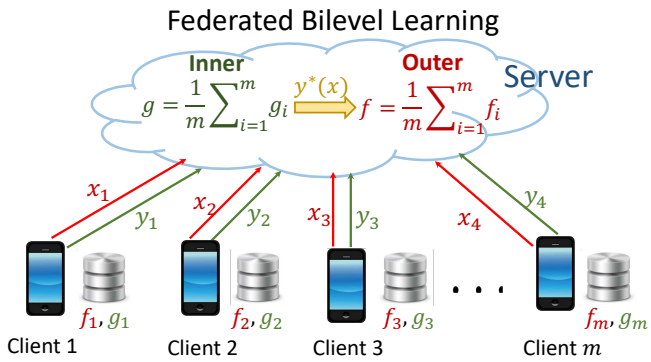
Existing FL methods, such as FEDAVG, are widely applied to stochastic problems with *single-level* structure. Instead, many machine learning tasks – such as adversarial learning (Madry et al., 2017), meta learning (Bertinetto et al., 2018), hyperparameter optimization (Franceschi et al., 2018), reinforcement/imitation learning (Wu et al., 2020; Arora et al., 2020), and neural architecture search (Liu et al., 2018) – admit *nested* formulations that go beyond the standard single-level structure. Towards addressing such nested problems, bilevel optimization has received significant attention in the recent literature (Ghadimi & Wang, 2018; Hong et al., 2020; Ji et al., 2021); albeit in non-FL settings. On the other hand, federated versions have been elusive perhaps due to the additional challenges surrounding heterogeneity, communication, and inverse Hessian approximation.

Contributions: This paper addresses these challenges and develops **FEDNEST**: A federated machinery for nested problems with provable convergence and lightweight communication. FEDNEST is composed of **FEDINN**: a federated stochastic variance reduction algorithm (FEDSVRG) to solve the inner problem while avoiding client drift, and **FEDOUT**: a communication-efficient federated hypergradient algorithm for solving the outer problem. Importantly, we allow both inner & outer objectives to be finite sums over heterogeneous client functions. FEDNEST runs a variant of FEDSVRG on inner & outer variables in an alternating fashion as outlined in Algorithm 1. We make multiple algorithmic and theoretical contributions summarized below.

- **The variance reduction** of FEDINN enables robustness in the sense that local models converge to the globally optimal inner model despite client drift/heterogeneity unlike FEDAVG. While FEDINN is similar to FEDSVRG (Konečnỳ et al., 2018) and FEDLIN (Mitra et al., 2021), we make two key contributions: (i) We

[1]Email: tarzanaq@umich.edu, University of Michigan [2]Emails: {mli176@,oymak@ece.}ucr.edu, University of California, Riverside [3]Email: cthrampo@ece.ubc.ca, University of British Columbia. Correspondence to: Davoud Ataee Tarzanagh <tarzanaq@umich.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



Communication-efficiency:
- ☑ FEDIHGP avoids explicit Hessian
- ☑ LFEDNEST for local hypergradients

Client heterogeneity:
- ☑ FEDINN avoids client drift

Finite sample bilevel theory:
- ☑ Stochastic inner & outer analysis

Specific nested optimization problems:
- ☑ Bilevel
- ☑ Minimax
- ☑ Compositional

Figure 1: Depiction of federated bilevel nested optimization and high-level summary of FEDNEST (Algorithm 1). At outer loop, FEDIHGP uses multiple rounds of matrix-vector products to facilitate hypergradient computation while only communicating vectors. At inner loop, FEDINN uses FEDSVRG to avoid client drift and find the unique global minima. Both are crucial for establishing provable convergence of FEDNEST.

leverage the global convergence of FEDINN to ensure accurate hypergradient computation which is crucial for our bilevel proof. (ii) We establish new convergence guarantees for single-level stochastic non-convex FEDSVRG, which are then integrated within our FEDOUT.

- **Communication efficient bilevel optimization**: Within FEDOUT, we develop an efficient federated method for hypergradient estimation that bypass Hessian computation. Our approach approximates the global Inverse Hessian-Gradient-Product (IHGP) via computation of matrix-vector products over few communication rounds.
- **LFEDNEST:** To further improve communication efficiency, we additionally propose a Light-FEDNEST algorithm, which computes hypergradients locally and only needs a single communication round for the outer update. Experiments reveal that LFEDNEST becomes very competitive as client functions become more homogeneous.
- **Unified federated nested theory:** We specialize our bilevel results to *minimax* and *compositional* optimization with emphasis on the former. For these, FEDNEST significantly simplifies and leads to faster convergence. Importantly, our results are on par with the state-of-the-art non-federated guarantees for nested optimization literature without additional assumptions (Table 1).
- We provide extensive **numerical experiments** [1] on bilevel and minimax optimization problems. These demonstrate the benefits of FEDNEST, efficiency of LFEDNEST, and shed light on tradeoffs surrounding communication, computation, and heterogeneity.

2. Federated Nested Problems & FEDNEST

We will first provide the background on bilevel nested problems and then introduce our general federated method.

[1] FEDNEST code is available at https://github.com/ucr-optml/FedNest.

Stochastic Bilevel Optimization

		Non-Federated		
	FEDNEST	**ALSET**	**BSA**	**TTSA**
batch size	$\mathcal{O}(1)$			
samples in ξ	$\mathcal{O}(\kappa_g^5\epsilon^{-2})$	$\mathcal{O}(\kappa_g^5\epsilon^{-2})$	$\mathcal{O}(\kappa_g^6\epsilon^{-2})$	$\mathcal{O}(\kappa_g^p\epsilon^{-2.5})$
samples in ζ	$\mathcal{O}(\kappa_g^9\epsilon^{-2})$	$\mathcal{O}(\kappa_g^9\epsilon^{-2})$	$\mathcal{O}(\kappa_g^9\epsilon^{-3})$	$\mathcal{O}(\kappa_g^p\epsilon^{-2.5})$

Stochastic Minimax Optimization

		Non-Federated		
	FEDNEST	**ALSET**	**SGDA**	**SMD**
batch size	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-1})$	N.A.
samples	$\mathcal{O}(\kappa_f^3\epsilon^{-2})$			

Stochastic Compositional Optimization

		Non-Federated		
	FEDNEST	**ALSET**	**SCGD**	**NASA**
batch size	$\mathcal{O}(1)$			
samples	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$

Table 1: Sample complexity of FEDNEST and comparable non-FL methods to find an ϵ-stationary point of f: $\kappa_g := \ell_{g,1}/\mu_g$ and $\kappa_f := \ell_{f,1}/\mu_f$. κ_g^p denotes a polynomial function of κ_g. ALSET (Chen et al., 2021a), BSA (Ghadimi & Wang, 2018), TTSA (Hong et al., 2020), SGDA (Lin et al., 2020), SMD (Rafique et al., 2021), SCGD (Wang et al., 2017), and NASA (Ghadimi et al., 2020).

Notation. For a differentiable function $h(\boldsymbol{x},\boldsymbol{y}) : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ in which $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}) : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, we denote $\nabla h \in \mathbb{R}^{d_1}$ the gradient of h as a function of $\boldsymbol{x}$ and $\nabla_{\boldsymbol{x}} h$, $\nabla_{\boldsymbol{y}} h$ the partial derivatives of h with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. We let $\nabla^2_{\boldsymbol{xy}} h$ and $\nabla^2_{\boldsymbol{y}} h$ denote the Jacobian and Hessian of h, respectively. We consider FL optimization over m clients and we denote $\mathcal{S} = \{1,\dots,m\}$. For vectors $\boldsymbol{v} \in \mathbb{R}^d$ and matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$, we denote $\|\boldsymbol{v}\|$ and $\|\boldsymbol{M}\|$ the respective Euclidean and spectral norms.

2.1. Preliminaries on Federated Nested Optimization

In **federated bilevel learning**, we consider the following nested optimization problem as depicted in Figure 1:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \quad & f(\boldsymbol{x}) = \tfrac{1}{m}\textstyle\sum_{i=1}^m f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) \\ \text{subj. to} \quad & \boldsymbol{y}^*(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \tfrac{1}{m}\textstyle\sum_{i=1}^m g_i\left(\boldsymbol{x},\boldsymbol{y}\right). \end{aligned} \tag{1a}$$

Algorithm 1 FEDNEST

Inputs: $K, T \in \mathbb{N}$; $(\boldsymbol{x}^0, \boldsymbol{y}^0) \in \mathbb{R}^{d_1+d_2}$; FEDINN, FEDOUT with stepsizes $\{(\alpha^k, \beta^k)\}_{k=0}^{K-1}$
for $k = 0, \cdots, K-1$ **do**
  $\boldsymbol{y}^{k,0} = \boldsymbol{y}^k$
  for $t = 0, \cdots, T-1$ **do**
    $\boldsymbol{y}^{k,t+1} = \text{FEDINN}\left(\boldsymbol{x}^k, \boldsymbol{y}^{k,t}, \beta^k\right)$
  end for
  $\boldsymbol{y}^{k+1} = \boldsymbol{y}^{k,T}$
  $\boldsymbol{x}^{k+1} = \text{FEDOUT}\left(\boldsymbol{x}^k, \boldsymbol{y}^{k+1}, \alpha^k\right)$
end for

Recall that m is the number of clients. Here, to model objective heterogeneity, each client i is allowed to have its own individual outer & inner functions (f_i, g_i). Moreover, we consider a general stochastic oracle model, access to local functions (f_i, g_i) is via stochastic sampling as follows:

$$f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) := \mathbb{E}_{\xi\sim\mathcal{C}_i}\left[f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}); \xi)\right], \\ g_i(\boldsymbol{x}, \boldsymbol{y}) := \mathbb{E}_{\zeta\sim\mathcal{D}_i}\left[g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta)\right], \tag{1b}$$

where $(\xi, \zeta) \sim (\mathcal{C}_i, \mathcal{D}_i)$ are outer/inner sampling distributions for the i^{th} client. We emphasize that for $i \neq j$, the tuples $(f_i, g_i, \mathcal{C}_i, \mathcal{D}_i)$ and $(f_j, g_j, \mathcal{C}_j, \mathcal{D}_j)$ can be different.

Example 1 (Hyperparameter tuning)**.** *Each client has local validation and training datasets associated with objectives $(f_i, g_i)_{i=1}^m$ corresponding to validation and training losses, respectively. The goal is finding hyper-parameters $\boldsymbol{x}$ that lead to learning model parameters $\boldsymbol{y}$ that minimize the (global) validation loss.*

The stochastic bilevel problem (1) subsumes two popular problem classes with the nested structure: *Stochastic Mini-Max* & *Stochastic Compositional*. Therefore, results on the general nested problem (1) also imply the results in these special cases. Below, we briefly describe them.

Minimax optimization. If $g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta) := -f_i(\boldsymbol{x}, \boldsymbol{y}; \xi)$ for all $i \in \mathcal{S}$, the stochastic bilevel problem (1) reduces to the stochastic minimax problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} f(\boldsymbol{x}) := \frac{1}{m} \max_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \sum_{i=1}^{m} \mathbb{E}[f_i(\boldsymbol{x}, \boldsymbol{y}; \xi)]. \tag{2}$$

Motivated by applications in fair beamforming, training generative-adversarial networks (GANs) and robust machine learning, significant efforts have been made for solving (2) including (Daskalakis & Panageas, 2018; Gidel et al., 2018; Mokhtari et al., 2020; Thekumparampil et al., 2019).

Example 2 (GANs)**.** *We train a generative model $g_{\boldsymbol{x}}(\cdot)$ and an adversarial model $a_{\boldsymbol{y}}(\cdot)$ using client datasets $\mathcal{C}_i$. The local functions may for example take the form $f_i(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}_{s\sim\mathcal{C}_i}\{\log a_{\boldsymbol{y}}(s)\} + \mathbb{E}_{z\sim\mathcal{D}_{noise}}\{\log[1 - a_{\boldsymbol{y}}(g_{\boldsymbol{x}}(z))]\}$.*

Compositional optimization. Suppose $f_i(\boldsymbol{x}, \boldsymbol{y}; \xi) := f_i(\boldsymbol{y}; \xi)$ and g_i is quadratic in $\boldsymbol{y}$ given as $g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta) := \|\boldsymbol{y} - \boldsymbol{r}_i(\boldsymbol{x}; \zeta)\|^2$. Then, the bilevel problem (1) reduces to

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \quad & f(\boldsymbol{x}) = \tfrac{1}{m}\textstyle\sum_{i=1}^m f_i\left(\boldsymbol{y}^*(\boldsymbol{x})\right) \\ \text{subj. to} \quad & \boldsymbol{y}^*(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^{d_2}} \tfrac{1}{m}\textstyle\sum_{i=1}^m g_i\left(\boldsymbol{x}, \boldsymbol{y}\right) \end{aligned} \tag{3}$$

with $f_i(\boldsymbol{y}^*(\boldsymbol{x})) := \mathbb{E}_{\xi\sim\mathcal{C}_i}[f_i(\boldsymbol{y}^*(\boldsymbol{x}); \xi)]$ and $g_i(\boldsymbol{x}, \boldsymbol{y}) := \mathbb{E}_{\zeta\sim\mathcal{D}_i}[g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta)]$. Optimization problems in the form of (3) occur for example in model agnostic meta-learning and policy evaluation in reinforcement learning (Finn et al., 2017; Ji et al., 2020b; Dai et al., 2017; Wang et al., 2017).

Assumptions. Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{d_1+d_2}$. Throughout, we make the following assumptions on inner/outer objectives.

Assumption A (Well-behaved objectives)**.** *For all $i \in [m]$:*

(A1) *$f_i(\boldsymbol{z}), \nabla f_i(\boldsymbol{z}), \nabla g_i(\boldsymbol{z}), \nabla^2 g_i(\boldsymbol{z})$ are $\ell_{f,0}, \ell_{f,1}, \ell_{g,1}, \ell_{g,2}$-Lipschitz continuous, respectively; and*

(A2) *$g_i(\boldsymbol{x}, \boldsymbol{y})$ is μ_g-strongly convex in $\boldsymbol{y}$ for all $\boldsymbol{x} \in \mathbb{R}^{d_1}$.*

Throughout, we use $\kappa_g = \ell_{g,1}/\mu_g$ to denote the condition number of the inner function g.

Assumption B (Stochastic samples)**.** *For all $i \in [m]$:*

(B1) *$\nabla f_i(\boldsymbol{z}; \xi), \nabla g_i(\boldsymbol{z}; \zeta), \nabla^2 g_i(\boldsymbol{z}; \zeta)$ are unbiased estimators of $\nabla f_i(\boldsymbol{z}), \nabla g_i(\boldsymbol{z}), \nabla^2 g_i(\boldsymbol{z})$, respectively; and*

(B2) *Their variances are bounded, i.e., $\mathbb{E}_\xi[\|\nabla f_i(\boldsymbol{z}; \xi) - \nabla f_i(\boldsymbol{z})\|^2] \leq \sigma_f^2$, $\mathbb{E}_\zeta[\|\nabla^2 g_i(\boldsymbol{z}; \zeta) - \nabla^2 g_i(\boldsymbol{z})\|^2] \leq \sigma_{g,1}^2$, and $\mathbb{E}_\zeta[\|\nabla^2 g_i(\boldsymbol{z}; \zeta) - \nabla^2 g_i(\boldsymbol{z})\|^2] \leq \sigma_{g,2}^2$ for some $\sigma_f^2, \sigma_{g,1}^2$, and $\sigma_{g,2}^2$.*

These assumptions are common in the bilevel optimization literature (Ghadimi & Wang, 2018; Chen et al., 2021a; Ji et al., 2021). Assumption A requires that the inner and outer functions are well-behaved. Specifically, strong-convexity of the inner objective is a recurring assumption in bilevel optimization theory implying a unique solution to the inner minimization in (1).

2.2. Proposed Algorithm: FEDNEST

In this section, we develop FEDNEST, which is formally presented in Algorithm 1. The algorithm operates in two nested loops. The outer loop operates in rounds $k \in \{1, \ldots, K\}$. Within each round, an inner loop operating for T iterations is executed. Given estimates $\boldsymbol{x}^k$ and $\boldsymbol{y}^k$, each iteration $t \in \{1, \ldots, T\}$ of the inner loop produces a new *global* model $\boldsymbol{y}^{k,t+1}$ of the inner optimization variable $\boldsymbol{y}^*(\boldsymbol{x}^k)$ as the output of an optimizer FEDINN. The final estimate $\boldsymbol{y}^{k+1} = \boldsymbol{y}^{k,T}$ of the inner variable is then used by an optimizer FEDOUT to update the outer *global* model $\boldsymbol{x}^{k+1}$.

The subroutines FEDINN and FEDOUT are gradient-based optimizers. Each subroutine involves a certain number of

local training steps indexed by $\nu \in \{0, \ldots, \tau_i - 1\}$ that are performed at the i^{th} client. The local steps of FEDINN iterate over *local* models $\boldsymbol{y}_{i,\nu}$ of the inner variable. Accordingly, FEDOUT iterates over *local* models $\boldsymbol{x}_{i,\nu}$ of the global variable. A critical component of FEDOUT is a communication-efficient federated hypergradient estimation routine, which we call FEDIHGP. The implementation of FEDINN, FEDOUT and FEDIHGP is critical to circumvent the algorithmic challenges of federated bilevel optimization. In the remaining of this section, we detail the challenges and motivate our proposed implementations. Later, in Section 3, we provide a formal convergence analysis of FEDNEST.

2.3. Key Challenge: Federated Hypergradient Estimation

FEDOUT is a gradient-based optimizer for the outer minimization in (1); thus each iteration involves computing $\nabla f(\boldsymbol{x}) = (1/m)\sum_{i=1}^m \nabla f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$. Unlike single-level FL, the fact that the outer objective f depends explicitly on the inner minimizer $\boldsymbol{y}^*(\boldsymbol{x})$ introduces a new challenge. A good starting point to understand the challenge is the following evaluation of $\nabla f(\boldsymbol{x})$ in terms of partial derivatives. The result is well-known from properties of implicit functions.

Lemma 2.1. *Under Assumption A, for all $i \in [m]$:*

$$\nabla f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) = \nabla^{\text{D}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) + \nabla^{\text{I}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right),$$

where the direct *and* indirect *gradient components are:*

$$\nabla^{\text{D}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) := \nabla_{\boldsymbol{x}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right), \tag{4a}$$

$$\nabla^{\text{I}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) := -\nabla^2_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) \cdot \left[\nabla^2_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))\right]^{-1} \nabla_{\boldsymbol{y}} f_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right). \tag{4b}$$

We now use the above formula to describe the two core challenges of bilevel FL optimization.

First, evaluation of any of the terms in (4) requires access to the minimizer $\boldsymbol{y}^*(\boldsymbol{x})$ of the inner problem. On the other hand, one may at best hope for a good *approximation* to $\boldsymbol{y}^*(\boldsymbol{x})$ produced by the inner optimization subroutine. Of course, this challenge is inherent in any bilevel optimization setting, but is exacerbated in the FL setting because of *client drift*. Specifically, when clients optimize their individual (possibly different) local inner objectives, the global estimate of the inner variable produced by SGD-type methods may drift far from (a good approximation to) $\boldsymbol{y}^*(\boldsymbol{x})$. We explain in Section 2.5 how FEDINN solves that issue.

The second challenge comes from the stochastic nature of the problem. Observe that the indirect component in (4b) is nonlinear in the Hessian $\nabla^2_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$, complicating an unbiased stochastic approximation of $\nabla f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$. As we expose here, solutions to this complication developed in the non-federated bilevel optimization literature, are *not* directly applicable in the FL setting. Indeed, existing stochastic bilevel algorithms, e.g. (Ghadimi & Wang, 2018), define $\bar{\nabla} f(\boldsymbol{x}, \boldsymbol{y}) := \bar{\nabla}^{\text{D}} f(\boldsymbol{x}, \boldsymbol{y}) + \bar{\nabla}^{\text{I}} f(\boldsymbol{x}, \boldsymbol{y})$ as a surrogate of $\nabla f(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$ by replacing $\boldsymbol{y}^*(\boldsymbol{x})$ in definition (4) with an approximation $\boldsymbol{y}$ and using the following stochastic approximations:

$$\bar{\nabla}^{\text{D}} f(\boldsymbol{x}, \boldsymbol{y}) \approx \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}; \dot{\xi}), \tag{5a}$$

$$\bar{\nabla}^{\text{I}} f(\boldsymbol{x}, \boldsymbol{y}) \approx -\nabla^2_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}; \zeta_{N'+1}) \left[\frac{N}{\ell_{g,1}} \prod_{n=1}^{N'} \left(\boldsymbol{I} - \frac{1}{\ell_{g,1}} \nabla^2_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y}; \zeta_n)\right)\right] \nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}; \dot{\xi}). \tag{5b}$$

Here, N' is drawn from $\{0, \ldots, N-1\}$ uniformly at random (UAR) and $\{\dot{\xi}, \zeta_1, \ldots, \zeta_{N'+1}\}$ are i.i.d. samples. Ghadimi & Wang (2018); Hong et al. (2020) have shown that using (5), the inverse Hessian estimation bias exponentially decreases with the number of samples N.

One might hope to directly leverage the above approach in a local computation fashion by replacing the global outer function f with the individual function f_i. However, note from (4b) and (5b) that the proposed stochastic approximation of the indirect gradient involves in a nonlinear way the *global* Hessian, which is *not* available at the client [2]. Communication efficiency is one of the core objectives of FL making the idea of communicating Hessians between clients and server prohibitive. *Is it then possible, in a FL setting, to obtain an accurate stochastic estimate of the indirect gradient while retaining communication efficiency?* In Section 2.4, we show how FEDOUT and its subroutine FEDIHGP, a matrix-vector products-based (thus, communication efficient) federated hypergradient estimator, answer this question affirmatively.

2.4. Outer Optimizer: FEDOUT

This section presents the outer optimizer FEDOUT, formally described in Algorithm 2. As a subroutine of FEDNEST (see Line 9, Algorithm 1), at each round $k = 0, \ldots, K-1$, FEDOUT takes the most recent global outer model $\boldsymbol{x}^k$ together the updated (by FEDINN) global inner model $\boldsymbol{y}^{k+1}$ and produces an update $\boldsymbol{x}^{k+1}$. To lighten notation, for a round k, denote the function's input as $(\boldsymbol{x}, \boldsymbol{y}^+)$ (instead of $(\boldsymbol{x}^k, \boldsymbol{y}^{k+1})$) and the output as $\boldsymbol{x}^+$ (instead of $\boldsymbol{x}^{k+1}$). For each client $i \in \mathcal{S}$, FEDOUT uses stochastic approximations of $\bar{\nabla}^{\text{I}} f_i(\boldsymbol{x}, \boldsymbol{y}^+)$ and $\bar{\nabla}^{\text{D}} f_i(\boldsymbol{x}, \boldsymbol{y}^+)$, which we call $\boldsymbol{h}_i^{\text{I}}(\boldsymbol{x}, \boldsymbol{y}^+)$ and $\boldsymbol{h}_i^{\text{D}}(\boldsymbol{x}, \boldsymbol{y}^+)$, respectively. The specific choice of these approximations (see Line 5) is critical and is discussed in detail later in this section. Before that, we explain

[2] We note that the approximation in (5) is not the only construction, and bilevel optimization can accommodate other forms of gradient surrogates (Ji et al., 2021). Yet, all these approximations require access (in a nonlinear fashion) to the *global* Hessian; thus, they suffer from the same challenge in FL setting.

Algorithm 2 $x^+ =$ **FEDOUT** (x, y^+, α) for stochastic bilevel and minimax problems

$\mathbb{F}_i(\cdot) \leftarrow \nabla_{\boldsymbol{x}} f_i(\cdot, \boldsymbol{y}^+; \cdot)$
$\boldsymbol{x}_{i,0} = \boldsymbol{x}$ and $\alpha_i \in (0, \alpha]$
Choose $N \geq 1$ and set $\boldsymbol{p}_{N'} =$ **FEDIHGP** $(\boldsymbol{x}, \boldsymbol{y}^+, N)$
for $i \in \mathcal{S}$ **in parallel do**
  $\boldsymbol{h}_i = \mathbb{F}_i(\boldsymbol{x}; \xi_i) - \nabla^2_{\boldsymbol{xy}} g_i(\boldsymbol{x}, \boldsymbol{y}^+; \zeta_i) \boldsymbol{p}_{N'}$
  $\boldsymbol{h}_i = \mathbb{F}_i(\boldsymbol{x}; \xi_i)$
end for
$\boldsymbol{h} = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \boldsymbol{h}_i$
for $i \in \mathcal{S}$ **in parallel do**
  for $\nu = 0, \ldots, \tau_i - 1$ **do**
    $\boldsymbol{h}_{i,\nu} = \mathbb{F}_i(\boldsymbol{x}_{i,\nu}; \xi_{i,\nu}) - \mathbb{F}_i(\boldsymbol{x}; \xi_{i,\nu}) + \boldsymbol{h}$
    $\boldsymbol{x}_{i,\nu+1} = \boldsymbol{x}_{i,\nu} - \alpha_i \boldsymbol{h}_{i,\nu}$
  end for
end for
$\boldsymbol{x}^+ = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \boldsymbol{x}_{i,\tau_i}$

Algorithm 3 $\boldsymbol{p}_{N'} =$ **FEDIHGP** $(\boldsymbol{x}, \boldsymbol{y}^+, N)$: Federated approximation of inverse-Hessian-gradient product

Select $N' \in \{0, \ldots, N-1\}$ UAR.
Select $\mathcal{S}_0 \in \mathcal{S}$ UAR.
for $i \in \mathcal{S}_0$ **in parallel do**
  $\boldsymbol{p}_{i,0} = \nabla_{\boldsymbol{y}} f_i(\boldsymbol{x}, \boldsymbol{y}^+; \xi_{i,0})$
end for
$\boldsymbol{p}_0 = \frac{N}{\ell_{g,1}} |\mathcal{S}_0|^{-1} \sum_{i \in \mathcal{S}_0} \boldsymbol{p}_{i,0}$
if $N' = 0$ **then**
  Return $\boldsymbol{p}_{N'}$
end if
Select $\mathcal{S}_1, \ldots, \mathcal{S}_{N'} \in \mathcal{S}$ UAR.
for $n = 1, \ldots, N'$ **do**
  for $i \in \mathcal{S}_n$ **in parallel do**
    $\boldsymbol{p}_{i,n} = \left(\boldsymbol{I} - \frac{1}{\ell_{g,1}} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}^+; \zeta_{i,n}) \right) \boldsymbol{p}_{n-1}$
  end for
  $\boldsymbol{p}_n = |\mathcal{S}_n|^{-1} \sum_{i \in \mathcal{S}_n} \boldsymbol{p}_{i,n}$
end for

how each client uses these proxies to form local updates of the outer variable. In each round, starting from a common global model $\boldsymbol{x}_{i,0} = \boldsymbol{x}$, each client i performs τ_i local steps (in parallel):

$$\boldsymbol{x}_{i,\nu+1} = \boldsymbol{x}_{i,\nu} - \alpha_i \boldsymbol{h}_{i,\nu}, \tag{6}$$

and then the server aggregates local models via $\boldsymbol{x}^+ = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \boldsymbol{x}_{i,\tau_i}$. Here, $\alpha_i \in (0, \alpha]$ is the local stepsize,

$$\begin{aligned} \boldsymbol{h}_{i,\nu} :=& \boldsymbol{h}^{\mathrm{I}}(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+) \\ & - \boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+), \end{aligned} \tag{7}$$

and, $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) := |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \boldsymbol{h}_i(\boldsymbol{x}, \boldsymbol{y}) = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} \left(\boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+) - \boldsymbol{h}_i^{\mathrm{I}}(\boldsymbol{x}, \boldsymbol{y}^+) \right)$.

The key features of updates (6)–(7) are exploiting past gradients (variance reduction) to account for objective heterogeneity. Indeed, the ideal update in FEDOUT would perform the update $\boldsymbol{x}_{i,\nu+1} = \boldsymbol{x}_{i,\nu} - \alpha_i \big(\boldsymbol{h}^{\mathrm{I}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) + \boldsymbol{h}^{\mathrm{D}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) \big)$ using the global gradient estimates. But this requires each client i to have access to both direct and indirect gradients of all other clients–which it does not, since clients do not communicate between rounds. To overcome this issue, each client i uses global gradient estimates, i.e., $\boldsymbol{h}^{\mathrm{I}}(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+)$ from the beginning of each round as a guiding direction in its local update rule. However, since both $\boldsymbol{h}^{\mathrm{D}}$ and $\boldsymbol{h}^{\mathrm{I}}$ are computed at a previous $(\boldsymbol{x}, \boldsymbol{y}^+)$, client i makes a correction by subtracting off the stale direct gradient estimate $\boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+)$ and adding its own *local* estimate $\boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)$. Our local update rule in Step 11 of Algorithm 2 is precisely of this form, i.e., $\boldsymbol{h}_{i,\nu}$ approximates $\boldsymbol{h}^{\mathrm{I}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) + \boldsymbol{h}^{\mathrm{D}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)$ via (7). Note here that the described local correction of FEDOUT only applies to the direct gradient component (the indirect component would require global Hessian information). An alterantive approach leading to LFEDNEST is discussed in Section 2.6.

FEDOUT applied to special nested problems. Algorithm 2 naturally allows the use of other optimizers for minimax & compositional optimization. For example, in the minimax problem (2), the bilevel gradient components are $\nabla^{\mathrm{D}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) = \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$ and $\nabla^{\mathrm{I}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})) = 0$ for all $i \in \mathcal{S}$. Hence, the hypergradient estimate (7) reduces to

$$\boldsymbol{h}_{i,\nu} = \boldsymbol{h}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+) - \boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}_i^{\mathrm{D}}(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+). \tag{8}$$

For the compositional problem (3), Hessian becomes the identity matrix, the direct gradient is the zero vector, and $\nabla_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}) = -(1/m) \sum_{i=1}^m \nabla \boldsymbol{r}_i(\boldsymbol{x})^\top$. Hence, $\boldsymbol{h}_i = \ell_{g,1} \nabla \boldsymbol{r}_i(\boldsymbol{x})^\top \boldsymbol{p}_0$ for all $i \in \mathcal{S}$.

More details on these special cases are provided in Appendices D and E.

Indirect gradient estimation & FEDIHGP. Here, we aim to address one of the key challenges in nested FL: inverse Hessian gradient product. Note from (5b) that the proposed stochastic approximation of the indirect gradient involves in a nonlinear way the *global* Hessian, which is *not* available at the client. To get around this, we use a client sampling strategy and recursive reformulation of (5b) so that $\bar{\nabla}^{\mathrm{I}} f_i(\boldsymbol{x}, \boldsymbol{y})$ can be estimated in an efficient federated manner. In particular, given $N \in \mathbb{N}$, we select $N' \in \{0 \ldots, N-1\}$ and $\mathcal{S}_0, \ldots, \mathcal{S}_{N'} \in \mathcal{S}$ UAR. For all $i \in \mathcal{S}$, we then define

$$\boldsymbol{h}_i^{\mathrm{I}}(\boldsymbol{x}, \boldsymbol{y}) = -\nabla^2_{\boldsymbol{xy}} g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta_i) \boldsymbol{p}_{N'}, \tag{9a}$$

where $\boldsymbol{p}_{N'} = |\mathcal{S}_0|^{-1} \widehat{\boldsymbol{H}}_{\boldsymbol{y}} \sum_{i \in \mathcal{S}_0} \nabla_{\boldsymbol{y}} f_i(\boldsymbol{x}, \boldsymbol{y}; \xi_{i,0})$ and $\widehat{\boldsymbol{H}}_{\boldsymbol{y}}$

is the approximate inverse Hessian:

$$\frac{N}{\ell_{g,1}}\prod_{n=1}^{N'}\Big(\boldsymbol{I}-\frac{1}{\ell_{g,1}|\mathcal{S}_n|}\sum_{i=1}^{|\mathcal{S}_n|}\nabla^2_{\boldsymbol{y}}g_i(\boldsymbol{x},\boldsymbol{y};\zeta_{i,n})\Big). \tag{9b}$$

The subroutine FEDIHGP provides a recursive strategy to compute $\boldsymbol{p}_{N'}$ and FEDOUT multiplies $\boldsymbol{p}_{N'}$ with the global Jacobian to drive an indirect gradient estimate. Importantly, these approximations require only matrix-vector products and vector communications.

Lemma 2.2. *Under Assumptions A and B, the approximate inverse Hessian $\widehat{\boldsymbol{H}}_{\boldsymbol{y}}$ defined in (9b) satisfies the following for any $\boldsymbol{x}$ and $\boldsymbol{y}$:*

$$\Big\|\big[\nabla^2_{\boldsymbol{y}}g(\boldsymbol{x},\boldsymbol{y})\big]^{-1}-\mathbb{E}_{\mathcal{W}}[\widehat{\boldsymbol{H}}_{\boldsymbol{y}}]\Big\|\le\frac{1}{\mu_g}\left(\frac{\kappa_g-1}{\kappa_g}\right)^N,$$
$$\mathbb{E}_{\mathcal{W}}\left[\Big\|\big[\nabla^2_{\boldsymbol{y}}g(\boldsymbol{x},\boldsymbol{y})\big]^{-1}-\widehat{\boldsymbol{H}}_{\boldsymbol{y}}\Big\|\right]\le\frac{2}{\mu_g}. \tag{10}$$

Here, $\mathcal{W}:=\{\mathcal{S}_n,\xi_i,\zeta_i,\xi_{i,0},\zeta_{i,n}\mid i\in\mathcal{S}_n,\quad 0\le n\le N'\}$. Further, for all $i\in\mathcal{S}$, $\boldsymbol{h}_i^{\mathcal{I}}(\boldsymbol{x},\boldsymbol{y})$ defined in (9a) satisfies

$$\big\|\mathbb{E}_{\mathcal{W}}\left[\boldsymbol{h}_i^{\mathcal{I}}(\boldsymbol{x},\boldsymbol{y})\right]-\bar{\nabla}^{\mathcal{I}}f_i(\boldsymbol{x},\boldsymbol{y})\big\|\le b, \tag{11}$$

where $b:=\kappa_g\ell_{f,1}\big((\kappa_g-1)/\kappa_g\big)^N$.

2.5. Inner Optimizer: FEDINN

In FL, each client performs multiple local training steps in isolation on its own data (using for example SGD) before communicating with the server. Due to such local steps, FEDAVG suffers from a *client-drift* effect under objective heterogeneity; that is, the local iterates of each client drift-off towards the minimum of their own local function. In turn, this can lead to convergence to a point different from the global optimum $\boldsymbol{y}^*(\boldsymbol{x})$ of the inner problem; e.g., see (Mitra et al., 2021). This behavior is particularly undesirable in a nested optimization setting since it directly affects the outer optimization; see, e.g. (Liu et al., 2021, Section 7).

In light of this observation, we build on the recently proposed FEDLIN (Mitra et al., 2021) which improves FEDSVRG (Konečnỳ et al., 2018) to solve the inner problem; see Algorithm 4. For each $i\in\mathcal{S}$, let $\boldsymbol{q}_i(\boldsymbol{x},\boldsymbol{y})$ denote an unbiased estimate of the gradient $\nabla_{\boldsymbol{y}}g_i(\boldsymbol{x},\boldsymbol{y})$. In each round, starting from a common global model $\boldsymbol{y}$, each client i performs τ_i local SVRG-type training steps in parallel: $\boldsymbol{y}_{i,\nu+1}=\boldsymbol{y}_{i,\nu}-\beta_i\boldsymbol{q}_{i,\nu}$, where $\boldsymbol{q}_{i,\nu}:=\boldsymbol{q}_i(\boldsymbol{x},\boldsymbol{y}_{i,\nu})-\boldsymbol{q}_i(\boldsymbol{x},\boldsymbol{y})+\boldsymbol{q}(\boldsymbol{x},\boldsymbol{y})$, $\beta_i\in(0,\beta]$ is the local inner stepsize, and $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{y}):=|\mathcal{S}|^{-1}\sum_{i\in\mathcal{S}}\boldsymbol{q}_i(\boldsymbol{x},\boldsymbol{y})$. We note that for the optimization problems (1), (2), and (3), $\boldsymbol{q}_i(\boldsymbol{x},\boldsymbol{y}_{i,\nu})$ is equal to $\nabla_{\boldsymbol{y}}g_i(\boldsymbol{x},\boldsymbol{y}_{i,\nu};\zeta_{i,\nu})$, $-\nabla_{\boldsymbol{y}}f_i(\boldsymbol{x},\boldsymbol{y}_{i,\nu};\xi_{i,\nu})$, and $\boldsymbol{y}_{i,\nu}-\boldsymbol{r}_i(\boldsymbol{x};\zeta_{i,\nu})$, respectively; see Appendices C–E.

Algorithm 4 $\boldsymbol{y}^+=$ **FEDINN** $(\boldsymbol{x},\boldsymbol{y},\beta)$

$\mathbb{G}_i(\cdot)\leftarrow$ $\nabla_{\boldsymbol{y}}g_i(\boldsymbol{x},\cdot)$ (bilevel), $-\nabla_{\boldsymbol{y}}f_i(\boldsymbol{x},\cdot)$ (minimax)
$\boldsymbol{y}_{i,0}=\boldsymbol{y}$ and $\beta_i\in(0,\beta]$
for $i\in\mathcal{S}$ **in parallel do**
  $\boldsymbol{q}_i=\mathbb{G}_i(\boldsymbol{y};\zeta_i)$
end for
$\boldsymbol{q}=|\mathcal{S}|^{-1}\sum_{i\in\mathcal{S}}\boldsymbol{q}_i$
for $i\in\mathcal{S}$ **in parallel do**
  for $\nu=0,\dots,\tau_i-1$ **do**
    $\boldsymbol{q}_{i,\nu}=\mathbb{G}_i(\boldsymbol{y}_{i,\nu};\zeta_{i,\nu})-\mathbb{G}_i(\boldsymbol{y};\zeta_{i,\nu})+\boldsymbol{q}$
    $\boldsymbol{y}_{i,\nu+1}=\boldsymbol{y}_{i,\nu}-\beta_i\boldsymbol{q}_{i,\nu}$
  end for
end for
$\boldsymbol{y}^+=|\mathcal{S}|^{-1}\sum_{i\in\mathcal{S}}\boldsymbol{y}_{i,\tau_i}$

2.6. Light-FEDNEST: Communication Efficiency via Local Hypergradients

Each FEDNEST epoch k requires $2T+N+3$ communication rounds as follows: $2T$ rounds for SVRG of FEDINN, N iterations for inverse Hessian approximation within FEDIHGP and 3 additional aggregations. Note that, these are vector communications and we fully avoid Hessian communication. In Appendix A, we also propose simplified variants of FEDOUT and FEDIHGP, which are tailored to homogeneous or high-dimensional FL settings. These algorithms can then either use *local* Jacobian / inverse Hessian or their approximation, and can use either SVRG or SGD.

Light-FEDNEST: Specifically, we propose LFEDNEST where each client runs IHGP locally. This reduces the number of rounds to $T+1$, saving $T+N+2$ rounds (see experiments in Section 4 for performance comparison and Appendix A for further discussion.)

3. Convergence Analysis for FEDNEST

In this section, we present convergence results for FEDNEST. All proofs are relegated to Appendices C–E.

Theorem 3.1. *Suppose Assumptions A and B hold. Further, assume $\alpha_i^k=\alpha_k/\tau_i$ and $\beta_i^k=\beta_k/\tau_i$ for all $i\in\mathcal{S}$, where*

$$\beta_k=\frac{\bar{\beta}\alpha_k}{T},\quad \alpha_k=\min\left\{\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3,\frac{\bar{\alpha}}{\sqrt{K}}\right\} \tag{12}$$

for some positive constants $\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3,\bar{\alpha}$, and $\bar{\beta}$ independent of K. Then, for any $T\ge 1$, the iterates $\{(\boldsymbol{x}^k,\boldsymbol{y}^k)\}_{k\ge0}$ generated by FEDNEST *satisfy*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\big\|\nabla f(\boldsymbol{x}^k)\big\|^2\right]=\mathcal{O}\Big(\frac{\bar{\alpha}\max(\sigma_{g,1}^2,\sigma_{g,2}^2,\sigma_f^2)}{\sqrt{K}}+\frac{1}{\min(\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3)K}+b^2\Big),$$

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

Corollary 3.1 (Bilevel). *Under the same conditions as in Theorem 3.1, if* $N = \mathcal{O}(\kappa_g \log K)$ *and* $T = \mathcal{O}(\kappa_g^4)$*, then*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}^k)\right\|^2\right] = \mathcal{O}\left(\frac{\kappa_g^4}{K} + \frac{\kappa_g^{2.5}}{\sqrt{K}}\right).$$

For ϵ*-accurate stationary point, we need* $K = \mathcal{O}(\kappa_g^5 \epsilon^{-2})$.

Above, we choose $N \propto \kappa_g \log K$ to guarantee $b^2 \lesssim 1/\sqrt{K}$. In contrast, we use $T \gtrsim \kappa_g^4$ inner SVRG epochs. From Section 2.6, this would imply the communication cost is dominated by SVRG epochs N and $\mathcal{O}(\kappa_g^4)$ rounds.

From Corollary 3.1, we remark that FEDNEST matches the guarantees of centralized alternating SGD methods, such as ALSET (Chen et al., 2021a) and BSA (Ghadimi & Wang, 2018), despite federated setting, i.e. communication challenge, heterogeneity in the client objectives, and device heterogeneity.

3.1. Minimax Federated Learning

We focus on special features of federated minimax problems and customize the general results to yield improved convergence results for this special case. Recall from (2) that $g_i(\boldsymbol{x}, \boldsymbol{y}) = -f_i(\boldsymbol{x}, \boldsymbol{y})$ which implies that $b = 0$ and following Assumption A, $f_i(\boldsymbol{x}, \boldsymbol{y})$ is μ_f–strongly concave in $\boldsymbol{y}$ for all $\boldsymbol{x}$.

Corollary 3.2 (Minimax). *Denote* $\kappa_f = \ell_{f,1}/\mu_f$*. Assume same conditions as in Theorem 3.1 and* $T = \mathcal{O}(\kappa_f)$*. Then,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}^k)\right\|^2\right] = \mathcal{O}\left(\frac{\kappa_f^2}{K} + \frac{\kappa_f}{\sqrt{K}}\right).$$

Corollary 3.2 implies that for the minimax problem, the convergence rate of FEDNEST to the stationary point of f is $\mathcal{O}(1/\sqrt{K})$. Again, we note this matches the convergence rate of non-FL algorithms (see also Table 1) such as SGDA (Lin et al., 2020) and SMD (Rafique et al., 2021).

3.2. Compositional Federated Learning

Observe that in the compositional problem (3), the outer function is $f_i(\boldsymbol{x}, \boldsymbol{y}; \xi) = f_i(\boldsymbol{y}; \xi)$ and the inner function is $g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{r}_i(\boldsymbol{x}; \zeta)\|^2$, for all $i \in \mathcal{S}$. Hence, $b = 0$ and $\kappa_g = 1$.

Corollary 3.3 (Compositional). *Under the same conditions as in Theorem 3.1, if we select* $T = 1$ *in* (12)*. Then,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}^k)\right\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

Corollary 3.3 implies that for the compositional problem (3), the convergence rate of FEDNEST to the stationary point of f is $\mathcal{O}(1/\sqrt{K})$. This matches the convergence rate of non-federated stochastic algorithms such as SCGD (Wang et al., 2017) and NASA (Ghadimi et al., 2020) (Table 1).

3.3. Single-Level Federated Learning

Building upon the general results for stochastic nonconvex nested problems, we establish new convergence guarantees for *single-level* stochastic non-convex federated SVRG which is integrated within our FEDOUT. Note that in the single-level setting, the optimization problem (1) reduces to

$$\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \; f(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(\boldsymbol{x}) \tag{13}$$

with $f_i(\boldsymbol{x}) := \mathbb{E}_{\xi\sim\mathcal{C}_i}[f_i(\boldsymbol{x}; \xi)]$, where $\xi \sim \mathcal{C}_i$ is sampling distribution for the i^{th} client.

We make the following assumptions on (13) that are counterparts of Assumptions A and B.

Assumption C (Lipschitz continuity)**.** *For all* $i \in [m]$*,* $\nabla f_i(\boldsymbol{x})$ *is* L_f*-Lipschitz continuous.*

Assumption D (Stochastic samples)**.** *For all* $i \in [m]$*,* $\nabla f_i(\boldsymbol{x}; \xi)$ *is an unbiased estimator of* $\nabla f_i(\boldsymbol{x})$ *and its variance is bounded, i.e.,* $\mathbb{E}_\xi[\|\nabla f_i(\boldsymbol{x}; \xi) - \nabla f_i(\boldsymbol{x})\|^2] \le \sigma_f^2$*.*

Theorem 3.2 (Single-Level). *Suppose Assumptions C and D hold. Further, assume* $\alpha_i^k = \alpha_k/\tau_i$ *for all* $i \in \mathcal{S}$*, where*

$$\alpha_k = \min\left\{\bar{\alpha}_1, \frac{\bar{\alpha}}{\sqrt{K}}\right\} \tag{14}$$

for some $\bar{\alpha}_1, \bar{\alpha} > 0$*. Then,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla f(\boldsymbol{x}^k)\right\|^2\right] = \mathcal{O}\left(\frac{\Delta_f}{\bar{\alpha}_1 K} + \frac{\frac{\Delta_f}{\bar{\alpha}} + \bar{\alpha}\sigma_f^2}{\sqrt{K}}\right),$$

where $\Delta_f := f(\boldsymbol{x}^0) - \mathbb{E}[f(\boldsymbol{x}^K)]$.

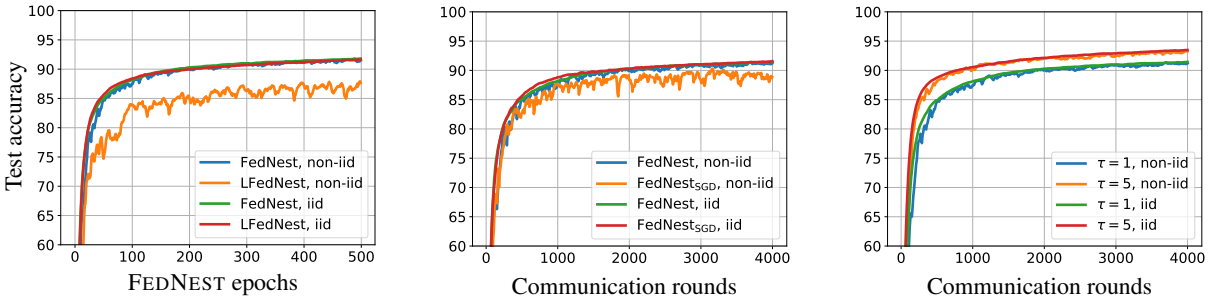
Theorem 3.2 extends recent results by (Mitra et al., 2021) from the stochastic strongly convex to the stochastic nonconvex setting. The above rate is also consistent with existing single-level non-FL guarantees (Ghadimi & Lan, 2013).

4. Numerical Experiments

In this section, we numerically investigate the impact of several attributes of our algorithms on a hyper-representation problem (Franceschi et al., 2018), a hyper-parameter optimization problem for loss function tuning (Li et al., 2021), and a federated minimax optimization problem.

4.1. Hyper-Representation Learning

Modern approaches in meta learning such as MAML (Finn et al., 2017) and reptile (Nichol & Schulman, 2018) learn



(a) Comparison between FEDNEST and LFEDNEST.

(b) SVRG in FEDINN provides better convergence and stability.

(c) Larger τ in FEDOUT provides better performance.

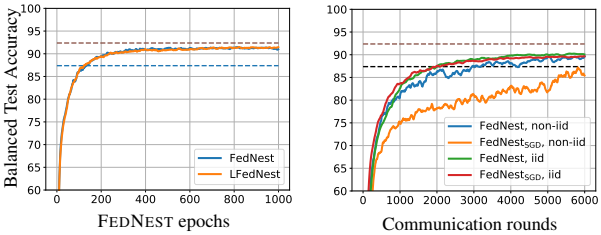
Figure 2: Hyper-representation experiments on a 2-layer MLP and MNIST dataset.

representations (that are shared across all tasks) in a bilevel manner. Similarly, the hyper-representation problem optimizes a classification model in a two-phased process. The outer objective optimizes the model backbone to obtain better feature representation on validation data. The inner problem optimizes a header for downstream classification tasks on training data. In this experiment, we use a 2-layer multilayer perceptron (MLP) with 200 hidden units. The outer problem optimizes the hidden layer with 157,000 parameters, and the inner problem optimizes the output layer with 2,010 parameters. We study both i.i.d and non-i.i.d. ways of partitioning the MNIST data exactly following FEDAVG (McMahan et al., 2017), and split each client's data evenly to train and validation datasets. Thus, each client has 300 train and 300 validation samples.

Figure 2 demonstrates the impact on test accuracy of several important components of FEDNEST. Figure 2a compares FEDNEST and LFEDNEST. Both algorithms perform well on the i.i.d. setup, while on the non-i.i.d. setup, FEDNEST achieves i.i.d. performance, significantly outperforming LFEDNEST. These findings are in line with our discussions in Section 2.6. LFEDNEST saves on communication rounds compared to FEDNEST and performs well on homogeneous clients. However, for heterogeneous clients, the isolation of local Hessian in LFEDNEST (see Algorithm 5 in Appendix A) degrades the test performance. Next, Figure 2b demonstrates the importance of SVRG in FEDINN algorithm for heterogeneous data (as predicted by our theoretical considerations in Section 2.5). To further clarify the algorithm difference in Figures 2b and 3a, we use FEDNEST$_{\text{SGD}}$ to denote the FEDNEST algorithm where SGD is used in FEDINN. Finally, Figure 2c elucidates the role of local epoch τ in FEDOUT: larger τ saves on communication and improves test performance by enabling faster convergence.

4.2. Loss Function Tuning on Imbalanced Dataset

We use bilevel optimization to tune a loss function for learning an imbalanced MNIST dataset. We aim to maximize the class-balanced validation accuracy (which helps minority/tail classes). Following the problem formulation in (Li et al., 2021) we tune the so-called VS-loss function (Kini et al., 2021) in a federated setting. In particular, we first create a long-tail imbalanced MNIST dataset by exponentially decreasing the number of examples per class (e.g. class 0 has 6,000 samples, class 1 has 3,597 samples and finally, class 9 has only 60 samples). We partition the dataset to 100 clients following again FEDAVG (McMahan et al., 2017) on both i.i.d. and non-i.i.d. setups. Different from the hyper-representation experiment, we employ 80%-20% train-validation on each client and use a 3-layer MLP model with 200, 100 hidden units, respectively. It is worth noting that, in this problem, the outer objective f (aka validation cost) only depends on the hyperparameter $\boldsymbol{x}$ through the optimal model parameters $\boldsymbol{y}^*(\boldsymbol{x})$; thus, the direct gradient $\nabla^{\text{D}} f_i(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))$ is zero for all $i \in \mathcal{S}$.



(a) FEDNEST achieves similar performance as centralized bilevel loss function tuning.

(b) SVRG in FEDINN provides better convergence and stability especially in non-iid setup.

Figure 3: Loss function tuning on a 3-layer MLP and imbalanced MNIST dataset to maximize *class-balanced test accuracy*. The *brown dashed* line is the accuracy on non-federated bilevel optimization (Li et al., 2021), and the *black dashed* line is the accuracy without tuning the loss function.

Figure 3 displays test accuracy vs epochs/rounds for our federated bilevel algorithms. The horizontal dashed lines serve as centralized baselines: brown depicts accuracy reached by bilevel optimization in non-FL setting, and, black depicts accuracy without any loss tuning. Compared to these, Figure 3a shows that FEDNEST achieves near non-federated performance. In Figure 3b, we investigate the key role of SVRG in FEDINN by comparing it with possible alternative

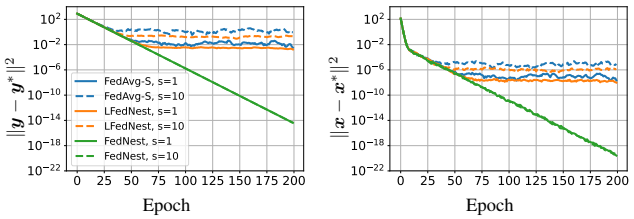


Figure 4: FEDNEST converges linearly despite heterogeneity. LFEDNEST slightly outperforms FEDAVG-S.

implementation that uses SGD-type updates. The figure confirms our discussion in Section 2.5: SVRG offers significant performance gains that are pronounced by client heterogeneity.

4.3. Federated Minimax Problem

We conduct experiments on the minimax problem (2) with

$$f_i(\boldsymbol{x},\boldsymbol{y}) := -\left[\frac{1}{2}\|\boldsymbol{y}\|^2 - \boldsymbol{b}_i^\top \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{A}_i \boldsymbol{x}\right] + \frac{\lambda}{2}\|\boldsymbol{x}\|^2,$$

to compare standard FEDAVG saddle-point (FEDAVG-S) method updating $(\boldsymbol{x},\boldsymbol{y})$ simultaneously (Hou et al., 2021) and our alternative approaches (LFEDNEST and FEDNEST). This is a saddle-point formulation of $\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}} \frac{1}{2}\|\frac{1}{m}\sum_{i=1}^m \boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i\|^2$. We set $\lambda = 10$, $\boldsymbol{b}_i = \boldsymbol{b}_i' - \frac{1}{m}\sum_{i=1}^m \boldsymbol{b}_i'$ and $\boldsymbol{A}_i = t_i \boldsymbol{I}$, where $\boldsymbol{b}_i' \sim \mathcal{N}(0, s^2 I_d)$, and t_i is drawn UAR over $(0, 0.1)$. Figure 4 shows that LFEDNEST and FEDNEST outperform FEDAVG-S thanks to their alternating nature. FEDNEST significantly improves the convergence of LFEDNEST due to controlling client-drift. To our knowledge, FEDNEST is the only *alternating* federated SVRG for minimax problems.

5. Related Work

Federated learning. FEDAVG was first introduced by McMahan et al. (2017), who showed it can dramatically reduce communication costs. For identical clients, FEDAVG coincides with local SGD (Zinkevich et al., 2010) which has been analyzed by many works (Stich, 2019; Yu et al., 2019; Wang & Joshi, 2018). Recently, many variants of FEDAVG have been proposed to tackle issues such as convergence and client drift. Examples include FEDPROX (Li et al., 2020b), SCAFFOLD (Karimireddy et al., 2020), FEDSPLIT (Pathak & Wainwright, 2020), FEDNOVA (Wang et al., 2020), and, the most closely relevant to us FEDLIN (Mitra et al., 2021). A few recent studies are also devoted to the extension of FEDAVG to the minimax optimization (Rasouli et al., 2020; Deng et al., 2020) and compositional optimization (Huang et al., 2021). In contrast to these methods, FEDNEST makes alternating SVRG updates between the global variables $\boldsymbol{x}$ and $\boldsymbol{y}$, and yields sample complexity bounds and batch size choices that are on par with the non-FL guarantees (Table 1). Evaluations in the Appendix H.1 reveal that both alternating updates and SVRG provides a performance boost over these prior approaches.

Bilevel optimization. This class of problems was first introduced by (Bracken & McGill, 1973), and since then, different types of approaches have been proposed. See (Sinha et al., 2017; Liu et al., 2021) for surveys. Earlier works in (Aiyoshi & Shimizu, 1984; Lv et al., 2007) reduced the bilevel problem to a single-level optimization problem. However, the reduced problem is still difficult to solve due to for example a large number of constraints. Recently, more efficient gradient-based algorithms have been proposed by estimating the hypergradient of $\nabla f(\boldsymbol{x})$ through iterative updates (Maclaurin et al., 2015; Franceschi et al., 2017; Domke, 2012; Pedregosa, 2016). The asymptotic and non-asymptotic analysis of bilevel optimization has been provided in (Franceschi et al., 2018; Shaban et al., 2019; Liu et al., 2020) and (Ghadimi & Wang, 2018; Hong et al., 2020), respectively. There is also a line of work focusing on minimax optimization (Nemirovski, 2004; Daskalakis & Panageas, 2018) and compositional optimization (Wang et al., 2017). Closely related to our work are (Lin et al., 2020; Rafique et al., 2021; Chen et al., 2021a) and (Ghadimi et al., 2020; Chen et al., 2021a) which provide non-asymptotic analysis of SGD-type methods for minimax and compositional problems with outer nonconvex objective, respectively.

A more in-depth discussion of related work is given in Appendix B. We summarize the complexities of different methods for FL/non-FL bilevel optimization in Table 1.

6. Conclusions

We presented a new class of federated algorithms for solving general nested stochastic optimization spanning bilevel and minimax problems. FEDNEST runs a variant of federated SVRG on inner & outer variables in an alternating fashion. We established provable convergence rates for FEDNEST under arbitrary client heterogeneity and introduced variations for min-max and compositional problems and for improved communication efficiency (LFEDNEST). We showed that, to achieve an ϵ-stationary point of the nested problem, FEDNEST requires $O(\epsilon^{-2})$ samples in total, which matches the complexity of the non-federated nested algorithms in the literature.

Acknowledgements

Davoud Ataee Tarzanagh was supported by ARO YIP award W911NF1910027 and NSF CAREER award CCF-1845076. Christos Thrampoulidis was supported by NSF Grant Numbers CCF-2009030 and HDR-1934641, and an NSERC Discovery Grant. Mingchen Li and Samet Oymak were supported by the NSF CAREER award CCF-2046816, Google Research Scholar award, and ARO grant W911NF2110312.

References

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.

Al-Khayyal, F. A., Horst, R., and Pardalos, P. M. Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. *Annals of Operations Research*, 34(1):125–147, 1992.

Arora, S., Du, S., Kakade, S., Luo, Y., and Saunshi, N. Provable representation learning for imitation learning via bi-level optimization. In *International Conference on Machine Learning*, pp. 367–376. PMLR, 2020.

Barazandeh, B., Huang, T., and Michailidis, G. A decentralized adaptive momentum method for solving a class of min-max optimization problems. *Signal Processing*, 189: 108245, 2021a.

Barazandeh, B., Tarzanagh, D. A., and Michailidis, G. Solving a class of non-convex min-max games using adaptive momentum methods. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3625–3629. IEEE, 2021b.

Basu, D., Data, D., Karakus, C., and Diggavi, S. Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, pp. 14668–14679, 2019.

Bertinetto, L., Henriques, J. F., Torr, P. H., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

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

Brown, G. W. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

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, 2021a.

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

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.

Dai, B., He, N., Pan, Y., Boots, B., and Song, L. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pp. 1458–1467. PMLR, 2017.

Daskalakis, C. and Panageas, I. The limit points of (optimistic) gradient descent in min-max optimization. *arXiv preprint arXiv:1807.03907*, 2018.

Deng, Y. and Mahdavi, M. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. In *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1395. PMLR, 2021.

Deng, Y., Kamani, M. M., and Mahdavi, M. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33:15111–15122, 2020.

Diakonikolas, J., Daskalakis, C., and Jordan, M. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2746–2754. PMLR, 2021.

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

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.

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.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

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

Ghadimi, S., Ruszczynski, A., and Wang, M. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

Gidel, G., Berard, H., Vignoud, G., Vincent, P., and Lacoste-Julien, S. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

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

Guo, Z., Xu, Y., Yin, W., Jin, R., and Yang, T. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*, 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.

Hou, C., Thekumparampil, K. K., Fanti, G., and Oh, S. Efficient algorithms for federated saddle point optimization. *arXiv preprint arXiv:2102.06333*, 2021.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

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

Huang, F., Li, J., and Huang, H. Compositional federated learning: Applications in distributionally robust averaging and meta learning. *arXiv preprint arXiv:2106.11264*, 2021.

Ji, K. and Liang, Y. Lower bounds and accelerated algorithms for bilevel optimization. *ArXiv*, abs/2102.03926, 2021.

Ji, K., Yang, J., and Liang, Y. Provably faster algorithms for bilevel optimization and applications to meta-learning. *ArXiv*, abs/2010.07962, 2020a.

Ji, K., Yang, J., and Liang, Y. Theoretical convergence of multi-step model-agnostic meta-learning. *arXiv preprint arXiv:2002.07836*, 2020b.

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.

Ji, S. A pytorch implementation of federated learning. Mar 2018. doi: 10.5281/zenodo.4321561.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khaled, A., Mishchenko, K., and Richtárik, P. First analysis of local GD on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 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. *arXiv preprint arXiv:2102.07367*, 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, 2021.

Kini, G. R., Paraskevas, O., Oymak, S., and Thrampoulidis, C. Label-imbalanced and group-sensitive classification under overparameterization. *arXiv preprint arXiv:2103.01550*, 2021.

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *International Conference on Learning Representations*, 2018.

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

Li, M., Zhang, X., Thrampoulidis, C., Chen, J., and Oymak, S. Autobalance: Optimized loss functions for imbalanced data. *Advances in Neural Information Processing Systems*, 34, 2021.

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, 2020b.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of FedAvg on non-IID data. *arXiv preprint arXiv:1907.02189*, 2019.

Lin, T., Jin, C., and Jordan, M. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pp. 6083–6093. PMLR, 2020.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, M., Mroueh, Y., Ross, J., Zhang, W., Cui, X., Das, P., and Yang, T. Towards better understanding of adaptive gradient algorithms in generative adversarial nets. *arXiv preprint arXiv:1912.11940*, 2019.

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

Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv preprint arXiv:2101.11517*, 2021.

Luo, L., Ye, H., Huang, Z., and Zhang, T. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33:20566–20577, 2020.

Lv, Y., Hu, T., Wang, G., and Wan, Z. A penalty function method based on kuhn–tucker condition for solving linear bilevel programming. *Applied Mathematics and Computation*, 188(1):808–813, 2007.

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.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

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

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Mokhtari, A., Ozdaglar, A., and Pattathil, S. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pp. 1497–1507. PMLR, 2020.

Moore, G. M. *Bilevel programming algorithms for machine learning model selection*. Rensselaer Polytechnic Institute, 2010.

Nazari, P., Tarzanagh, D. A., and Michailidis, G. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

Nedić, A. and Ozdaglar, A. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

Nemirovski, A. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Nouiehed, M., Sanjabi, M., Huang, T., Lee, J. D., and Razaviyayn, M. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Pathak, R. and Wainwright, M. J. Fedsplit: an algorithmic framework for fast federated optimization. *Advances in Neural Information Processing Systems*, 33:7057–7066, 2020.

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

Rafique, H., Liu, M., Lin, Q., and Yang, T. Weakly-convex–concave min–max optimization: provable algorithms and applications in machine learning. *Optimization Methods and Software*, pp. 1–35, 2021.

Rasouli, M., Sun, T., and Rajagopal, R. Fedgan: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*, 2020.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive

federated optimization. In *International Conference on Learning Representations*, 2020.

Reisizadeh, A., Farnia, F., Pedarsani, R., and Jadbabaie, A. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33:21554–21565, 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.

Shen, Y., Du, J., Zhao, H., Zhang, B., Ji, Z., and Gao, M. Fedmm: Saddle point optimization for federated adversarial domain adaptation. *arXiv preprint arXiv:2110.08477*, 2021.

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.

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

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1g2JnRcFX.

Stich, S. U. and Karimireddy, S. P. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Thekumparampil, K. K., Jain, P., Netrapalli, P., and Oh, S. Efficient algorithms for smooth minimax optimization. *arXiv preprint arXiv:1907.01543*, 2019.

Tran Dinh, Q., Liu, D., and Nguyen, L. Hybrid variance-reduced sgd algorithms for minimax problems with nonconvex-linear function. *Advances in Neural Information Processing Systems*, 33:11096–11107, 2020.

Wang, J. and Joshi, G. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 2020.

Wang, M., Liu, J., and Fang, E. Accelerating stochastic composition optimization. *Advances in Neural Information Processing Systems*, 29, 2016.

Wang, M., Fang, E. X., and Liu, H. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., and Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6): 1205–1221, 2019.

Wu, Y. F., Zhang, W., Xu, P., and Gu, Q. A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33:17617–17628, 2020.

Xie, J., Zhang, C., Zhang, Y., Shen, Z., and Qian, H. A federated learning framework for nonconvex-pl minimax problems. *arXiv preprint arXiv:2105.14216*, 2021.

Yan, Y., Xu, Y., Lin, Q., Liu, W., and Yang, T. Optimal epoch stochastic gradient descent ascent methods for min-max optimization. *Advances in Neural Information Processing Systems*, 33:5789–5800, 2020.

Yang, J., Kiyavash, N., and He, N. Global convergence and variance reduction for a class of nonconvex-nonconcave minimax problems. *Advances in Neural Information Processing Systems*, 33:1153–1165, 2020.

Yoon, T. and Ryu, E. K. Accelerated algorithms for smooth convex-concave minimax problems with o (1/kˆ 2) rate on squared gradient norm. In *International Conference on Machine Learning*, pp. 12098–12109. PMLR, 2021.

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.

Zinkevich, M., Weimer, M., Li, L., and Smola, A. J. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pp. 2595–2603, 2010.

APPENDIX
FEDNEST: Federated Bilevel, Minimax, and Compositional Optimization

The appendix is organized as follows: Section A introduces the LFEDNEST algorithm. Section B discusses the related work. We provide all details for the proof of the main theorems in Sections C, D, E, and F for federated bilevel, minimax, compositional, and single-level optimization, respectively. In Section G, we state a few auxiliary technical lemmas. Finally, in Section H, we provide the detailed parameters of our numerical experiments (Section 4) and then introduce further experiments.

A. LFEDNEST

Implementing FEDINN and FEDOUT naively by using the global direct and indirect gradients and sending the local information to the server that would then calculate the global gradients leads to a communication and space complexity of which can be prohibitive for large-sized d_1 and d_2. One can consider possible local variants of FEDINN and FEDOUT tailore to such scenarios. Each of the possible algorithms (See Table 2) can then either use the global gradient or only the local gradient, either use a SVRG or SGD.

Algorithm 5 $x^+ =$ **LFEDOUT** (x, y, α) for stochastic bilevel, minimax, and compositional problems

$x_{i,0} = x$ and $\alpha_i \in (0, \alpha]$ for each $i \in \mathcal{S}$.
Choose $N \in \{1, 2, \ldots\}$ (the number of terms of Neumann series).
for $i \in \mathcal{S}$ **in parallel do**
  for $\nu = 0, \ldots, \tau_i - 1$ **do**
    Select $N' \in \{0, \ldots, N-1\}$ UAR.
    $h_{i,\nu} = \nabla_x f_i(x_{i,\nu}, y; \xi_{i,\nu}) - \frac{N}{\ell_{g,1}} \nabla^2_{xy} g_i(x_{i,\nu}, y; \zeta_{i,\nu}) \prod_{n=1}^{N'} \big(I - \frac{1}{\ell_{g,1}} \nabla^2_y g_i(x_{i,\nu}, y; \zeta_{i,n})\big) \nabla_x f_i(y_{i,\nu}, y, \xi_{i,\nu})$
    $h_{i,\nu} = \nabla_x f_i(x_{i,\nu}, y; \xi_{i,\nu})$
    $h_{i,\nu} = \nabla r_i(x_{i,\nu}; \zeta_{i,\nu})^\top \nabla f_i(y_{i,\nu}; \xi_{i,\nu})$
    $x_{i,\nu+1} = x_{i,\nu} - \alpha_i h_{i,\nu}$
  end for
end for
$x^+ = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} x_{i,\tau_i}$

Algorithm 6 $y^+ =$ **LFEDINN** (x, y, β) for stochastic bilevel, minimax, and compositional problems

$y_{i,0} = y$ and $\beta_i \in (0, \beta]$ for each $i \in \mathcal{S}$.
for $i \in \mathcal{S}$ **in parallel do**
  for $\nu = 0, \ldots, \tau_i - 1$ **do**
    $q_{i,\nu} = \nabla_y g_i(x, y_{i,\nu}; \zeta_{i,\nu})$ $q_{i,\nu} = -\nabla_y f_i(x, y_{i,\nu}; \xi_{i,\nu})$ $q_{i,\nu} = y_{i,\nu} - r_i(x; \zeta_{i,\nu})$
    $y_{i,\nu+1} = y_{i,\nu} - \beta_i q_{i,\nu}$
  end for
end for
$y^+ = |\mathcal{S}|^{-1} \sum_{i \in \mathcal{S}} y_{i,\tau_i}$

	definition		properties			
	outer optimization	inner optimization	global outer gradient	global IHGP	global inner gradient	# communication rounds
FEDNEST	Algorithm 2 (SVRG on $\boldsymbol{x}$)	Algorithm 4 (SVRG on $\boldsymbol{y}$)	yes	yes	yes	$2T+N+3$
LFEDNEST	Algorithm 5 (SGD on $\boldsymbol{x}$)	Algorithm 6 (SGD on $\boldsymbol{y}$)	no	no	no	$T+1$
FEDNEST$_{\text{SGD}}$	Algorithm 2 (SVRG on $\boldsymbol{x}$)	Algorithm 6 (SGD on $\boldsymbol{y}$)	yes	yes	no	$T+N+3$
LFEDNEST$_{\text{SVRG}}$	Algorithm 5 (SGD on $\boldsymbol{x}$)	Algorithm 4 (SVRG on $\boldsymbol{y}$)	no	no	yes	$2T+1$

Table 2: Definition of studied algorithms by using inner/outer optimization algorithms and server updates and resulting properties of these algorithms. T and N denote the number of inner iterations and terms of Neumann series, respectively.

B. Related Work

We provide an overview of the current literature on non-federated nested (bilevel, minmimax, and compositional) optimization and federated learning.

B.1. Bilevel Optimization

A broad collection of algorithms have been proposed to solve bilevel nonlinear programming problems. Aiyoshi & Shimizu (1984); Edmunds & Bard (1991); Al-Khayyal et al. (1992); Hansen et al. (1992); Shi et al. (2005); Lv et al. (2007); Moore (2010) reduce the bilevel problem to a single-level optimization problem using for example the Karush-Kuhn-Tucker (KKT) conditions or penalty function methods. A similar idea was also explored in Khodak et al. (2021) where the authors provide a reformulation of the hyperparameter optimization (bilevel objective) into a single-level objective and develop a federated online method to solve it. However, the reduced single-level problem is usually difficult to solve (Sinha et al., 2017).

In comparison, alternating gradient-based approaches designed for the bilevel problems are more attractive due to their simplicity and effectiveness. This type of approaches estimate the hypergradient $\nabla f(\boldsymbol{x})$ for iterative updates, and are generally divided to approximate implicit differentiation (AID) and iterative differentiation (ITD) categories. ITD-based approaches (Maclaurin et al., 2015; Franceschi et al., 2017; Finn et al., 2017; Grazzi et al., 2020) estimate the hypergradient $\nabla f(\boldsymbol{x})$ in either a reverse (automatic differentiation) or forward manner. AID-based approaches (Pedregosa, 2016; Grazzi et al., 2020; Ghadimi & Wang, 2018) estimate the hypergradient via implicit differentiation which involves solving a linear system. Our algorithms follow the latter approach.

Theoretically, bilevel optimization has been studied via both asymptotic and non-asymptotic analysis (Franceschi et al., 2018; Liu et al., 2020; Li et al., 2020a; Shaban et al., 2019; Ghadimi & Wang, 2018; Ji et al., 2021; Hong et al., 2020). In particular, (Franceschi et al., 2018) provided the asymptotic convergence of a backpropagation-based approach as one of ITD-based algorithms by assuming the inner problem is strongly convex. (Shaban et al., 2019) gave a similar analysis for a *truncated* backpropagation approach. Non-asymptotic complexity analysis for bilevel optimization has also been explored. Ghadimi & Wang (2018) provided a finite-time convergence analysis for an AID-based algorithm under three different loss geometries, where $f(\cdot)$ is either strongly convex, convex or nonconvex, and $g(\boldsymbol{x},\cdot)$ is strongly convex. (Ji et al., 2021) provided an improved non-asymptotic analysis for AID- and ITD-based algorithms under the nonconvex-strongly-convex geometry. (Ji & Liang, 2021) provided the first-known lower bounds on complexity as well as tighter upper bounds. When the objective functions can be expressed in an expected or finite-time form, (Ghadimi & Wang, 2018; Ji et al., 2021; Hong et al., 2020) developed stochastic bilevel algorithms and provided the non-asymptotic analysis. (Chen et al., 2021a) provided a tighter analysis of SGD for stochastic bilevel problems. (Chen et al., 2021b; Guo et al., 2021; Khanduri et al., 2021; Ji et al., 2020a; Huang & Huang, 2021; Dagréou et al., 2022) studied accelerated SGD, SAGA, momentum, and adaptive-type bilevel optimization methods. More results can be found in the recent review paper (Liu et al., 2021) and references therein.

B.1.1. MINIMAX OPTIMIZATION

Minimax optimization has a long history dating back to (Brown, 1951). Earlier works focused on the deterministic convex-concave regime (Nemirovski, 2004; Nedić & Ozdaglar, 2009). Recently, there has emerged a surge of studies of stochastic minimax problems. The alternating version of the gradient descent ascent (SGDA) has been studied by incorporating the idea of optimism (Daskalakis & Panageas, 2018; Gidel et al., 2018; Mokhtari et al., 2020; Yoon & Ryu, 2021). (Rafique et al., 2021; Thekumparampil et al., 2019; Nouiehed et al., 2019; Lin et al., 2020) studied SGDA in the nonconvex-strongly concave setting. Specifically, the $\mathcal{O}(\epsilon^{-2})$ sample complexity has been established in (Lin et al., 2020) under an increasing batch size $\mathcal{O}(\epsilon^{-1})$. Chen et al. (2021a) provided the $\mathcal{O}(\epsilon^{-2})$ sample complexity under an $\mathcal{O}(1)$ constant batch size. In the same setting, accelerated GDA algorithms have been developed in (Luo et al., 2020; Yan et al., 2020; Tran Dinh et al., 2020). Going beyond the one-side concave settings, algorithms and their convergence analysis have been studied for nonconvex-nonconcave minimax problems with certain benign structure; see e.g., (Gidel et al., 2018; Liu et al., 2019; Yang et al., 2020; Diakonikolas et al., 2021; Barazandeh et al., 2021b). A comparison of our results with prior work can be found in Table 1.

B.1.2. COMPOSITIONAL OPTIMIZATION

Stochastic compositional gradient algorithms (Wang et al., 2017; 2016) can be viewed as an alternating SGD for the special compositional problem. However, to ensure convergence, the algorithms in (Wang et al., 2017; 2016) use two sequences of variables being updated in two different time scales, and thus the iteration complexity of (Wang et al., 2017) and (Wang et al., 2016) is worse than $\mathcal{O}(\epsilon^{-2})$ of the standard SGD. Our work is closely related to ALSET (Chen et al., 2021a), where an $\mathcal{O}(\epsilon^{-2})$ sample complexity has been established in a non-FL setting.

B.2. Federated Learning

FL involves learning a centralized model from distributed client data. Although this centralized model benefits from all client data, it raises several types of issues such as generalization, fairness, communication efficiency, and privacy (Mohri et al., 2019; Stich, 2019; Yu et al., 2019; Wang & Joshi, 2018; Stich & Karimireddy, 2019; Basu et al., 2019; Nazari et al., 2019; Barazandeh et al., 2021a). FEDAVG (McMahan et al., 2017) can tackle some of these issues such as high communication costs. Many variants of FEDAVG have been proposed to tackle other emerging issues such as convergence and *client drift*. Examples include adding a regularization term in the client objectives towards the broadcast model (Li et al., 2020b), proximal splitting (Pathak & Wainwright, 2020; Mitra et al., 2021), variance reduction (Karimireddy et al., 2020; Mitra et al., 2021) and adaptive updates (Reddi et al., 2020). When clients are homogeneous, FEDAVG is closely related to local SGD (Zinkevich et al., 2010), which has been analyzed by many works (Stich, 2019; Yu et al., 2019; Wang & Joshi, 2018; Stich & Karimireddy, 2019; Basu et al., 2019).

In order to analyze FEDAVG in heterogeneous settings, (Li et al., 2020b; Wang et al., 2019; Khaled et al., 2019; Li et al., 2019) derive convergence rates depending on the amount of heterogeneity. They showed that the convergence rate of FEDAVG gets worse with client heterogeneity. By using control variates to reduce client drift, the SCAFFOLD method (Karimireddy et al., 2020) achieves convergence rates that are independent of the amount of heterogeneity. Relatedly, FEDNOVA (Wang et al., 2020) and FEDLIN (Mitra et al., 2021) provided the convegence of their methods despite arbitrary local objective and systems heterogeneity. In particular, (Mitra et al., 2021) showed that FEDLIN guarantees linear convergence to the global minimum of deterministic objective, despite arbitrary objective and systems heterogeneity. As explained in the main body, our algorithms critically leverage these ideas after identifying the additional challenges that client drift brings to federated bilevel settings.

B.2.1. FEDERATED MINIMAX LEARNING

A few recent studies are devoted to federated minimax optimization (Rasouli et al., 2020; Reisizadeh et al., 2020; Deng et al., 2020; Hou et al., 2021). In particular, (Reisizadeh et al., 2020) consider minimax problem with inner problem satisfying PL condition and the outer one being either nonconvex or satisfying PL. However, the proposed algorithm only communicates $\boldsymbol{x}$ to the server. Xie et al. (2021) consider a general class of nonconvex-PL minimax problems in the cross-device federated learning setting. Their algorithm performs multiple local update steps on a subset of active clients in each round and leverages global gradient estimates to correct the bias in local update directions. Deng & Mahdavi (2021) studied federated optimization for a family of smooth nonconvex minimax functions. Shen et al. (2021) proposed a distributed minimax optimizer called FEDMM, designed specifically for the federated adversary domain adaptation problem.

Hou et al. (2021) proposed a SCAFFOLD saddle point algorithm (SCAFFOLD-S) for solving strongly convex-concave minimax problems in the federated setting. To the best of our knowledge, all the aforementioned developments require a bound on the *heterogeneity* of the local functions, and do not account for the effects of systems heterogeneity which is also a key challenge in FL. In addition, our work proposes the first *alternating* federated SVRG-type algorithm for minimax problems with iteration complexity that matches to the non-federated setting (see, Table 1).

C. Proof for Federated Bilevel Optimization

Throughout the proof, we will use $\mathcal{F}_{i,\nu}^{k,t}$ to denote the filtration that captures all the randomness up to the ν-th local step of client i in inner round t and outer round k. With a slight abuse of notation, $\mathcal{F}_{i,-1}^{k,t}$ is to be interpreted as $\mathcal{F}^{k,t}, \forall i \in \mathcal{S}$. For simplicity, we remove subscripts k and t from the definition of stepsize and model parameters. For example, $\boldsymbol{x}$ and $\boldsymbol{x}^+$ denote $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k+1}$, respectively. We further set

$$\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) := \mathbb{E}\Big[\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)|\mathcal{F}_{i,\nu-1}\Big]. \tag{15}$$

Proof of Lemma 2.1

Proof. Given $\boldsymbol{x} \in \mathbb{R}^{d_1}$, the optimality condition of the inner problem in (1) is $\nabla_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y}) = 0$. Now, since $\nabla_{\boldsymbol{x}} \left(\nabla_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y})\right) = 0$, we obtain

$$0 = \sum_{j=1}^{m} \left(\nabla^2_{\boldsymbol{xy}} g_j\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) + \nabla \boldsymbol{y}^*(\boldsymbol{x}) \nabla^2_{\boldsymbol{y}} g_j\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right)\right),$$

which implies

$$\nabla \boldsymbol{y}^*(\boldsymbol{x}) = -\left(\sum_{i=1}^{m} \nabla^2_{\boldsymbol{xy}} g_i\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right)\right)\left(\sum_{i=1}^{m} \nabla^2_{\boldsymbol{y}} g_j(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))\right)^{-1}.$$

The results follows from a simple application of the chain rule to f as follows:

$$\nabla f\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) = \nabla_{\boldsymbol{x}} f\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right) + \nabla \boldsymbol{y}^*(\boldsymbol{x}) \nabla_{\boldsymbol{y}} f\left(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x})\right).$$

□

Proof of Lemma 2.2

Proof. By independency of N', $\zeta_{i,n}$, and $\mathcal{S}_n$, and under Assumption B, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{W}}\left[\widehat{\boldsymbol{H}}_{\boldsymbol{y}}\right] &= \mathbb{E}_{\mathcal{W}}\left[\frac{N}{\ell_{g,1}} \prod_{n=1}^{N'} \left(\boldsymbol{I} - \frac{1}{\ell_{g,1}|\mathcal{S}_n|} \sum_{i=1}^{|\mathcal{S}_n|} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta_{i,n})\right)\right] \\
&= \mathbb{E}_{N'}\left[\mathbb{E}_{\mathcal{S}_{1:N'}}\left[\mathbb{E}_{\zeta}\left[\frac{N}{\ell_{g,1}} \prod_{n=1}^{N'} \left(\boldsymbol{I} - \frac{1}{\ell_{g,1}|\mathcal{S}_n|} \sum_{i=1}^{|\mathcal{S}_n|} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta_{i,n})\right)\right]\right]\right] \\
&= \tfrac{1}{\ell_{g,1}} \sum_{n=0}^{N-1} \left[\boldsymbol{I} - \tfrac{1}{\ell_{g,1}} \nabla^2_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y})\right]^n,
\end{aligned} \tag{16}$$

where the last equality follows from the uniform distribution of N'.

Note that since $\boldsymbol{I} \succeq \frac{1}{\ell_{g,1}} \nabla^2_{\boldsymbol{y}} g_i \succeq \frac{\mu_g}{\ell_{g,1}}$ for all $i \in [m]$ due to Assumption A, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{W}}\left[\|\widehat{\boldsymbol{H}}_{\boldsymbol{y}}\|\right] &\leq \frac{N}{\ell_{g,1}} \mathbb{E}_{\mathcal{W}}\left[\prod_{n=1}^{N'} \left\|\boldsymbol{I} - \frac{1}{\ell_{g,1}|\mathcal{S}_n|} \sum_{i=1}^{|\mathcal{S}_n|} \nabla^2_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta_{i,n})\right\|\right] \\
&\leq \frac{N}{\ell_{g,1}} \mathbb{E}_{N'}\left[1 - \frac{\mu_g}{\ell_{g,1}}\right]^{N'} = \frac{1}{\ell_{g,1}} \sum_{n=0}^{N-1} \left[1 - \frac{\mu_g}{\ell_{g,1}}\right]^n \leq \frac{1}{\mu_g}.
\end{aligned}$$

The reminder of the proof is similar to (Ghadimi & Wang, 2018). □

The following lemma extends (Ghadimi & Wang, 2018, Lemma 2.2) and (Chen et al., 2021a, Lemma 2) to the finite-sum problem (1). Proofs follow similarly by applying their analysis to the inner & outer functions (f_i, g_i), $\forall i \in \mathcal{S}$.

Lemma C.1. *Under Assumptions A and B, for all $\boldsymbol{x}_1$, $\boldsymbol{x}_2$:*

$$\|\nabla f(\boldsymbol{x}_1) - \nabla f(\boldsymbol{x}_2)\| \leq L_f \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \tag{17a}$$
$$\|\boldsymbol{y}^*(\boldsymbol{x}_1) - \boldsymbol{y}^*(\boldsymbol{x}_2)\| \leq L_{\boldsymbol{y}} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \tag{17b}$$
$$\|\nabla \boldsymbol{y}^*(\boldsymbol{x}_1) - \nabla \boldsymbol{y}^*(\boldsymbol{x}_2)\| \leq L_{\boldsymbol{yx}} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \tag{17c}$$

Also, for all $i \in \mathcal{S}$, $\nu \in \{0, \ldots, \tau_i - 1\}$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{y}$, we have:

$$\|\bar{\nabla} f_i(\boldsymbol{x}_1, \boldsymbol{y}) - \bar{\nabla} f_i(\boldsymbol{x}_1, \boldsymbol{y}^*(\boldsymbol{x}_1))\| \leq M_f \|\boldsymbol{y}^*(\boldsymbol{x}_1) - \boldsymbol{y}\|, \tag{17d}$$
$$\|\bar{\nabla} f_i(\boldsymbol{x}_2, \boldsymbol{y}) - \bar{\nabla} f_i(\boldsymbol{x}_1, \boldsymbol{y})\| \leq M_f \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|, \tag{17e}$$
$$\mathbb{E}\left[\|\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}) - \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y})\|^2\right] \leq \tilde{\sigma}_f^2, \tag{17f}$$
$$\mathbb{E}\left[\|\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\|^2 | \mathcal{F}_{i,\nu-1}\right] \leq \tilde{D}_f^2. \tag{17g}$$

Here,

$$\begin{aligned}
L_{\boldsymbol{y}} &:= \frac{\ell_{g,1}}{\mu_g} = \mathcal{O}(\kappa_g), \\
L_{\boldsymbol{yx}} &:= \frac{\ell_{g,2} + \ell_{g,2} L_{\boldsymbol{y}}}{\mu_g} + \frac{\ell_{g,1}}{\mu_g^2}\Big(\ell_{g,2} + \ell_{g,2} L_{\boldsymbol{y}}\Big) = \mathcal{O}(\kappa_g^3), \\
M_f &:= \ell_{f,1} + \frac{\ell_{g,1}\ell_{f,1}}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right) = \mathcal{O}(\kappa_g^2), \\
L_f &:= \ell_{f,1} + \frac{\ell_{g,1}(\ell_{f,1} + M_f)}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right) = \mathcal{O}(\kappa_g^3), \\
\tilde{\sigma}_f^2 &:= \sigma_f^2 + \frac{3}{\mu_g^2}\Big((\sigma_f^2 + \ell_{f,0}^2)(\sigma_{g,2}^2 + 2\ell_{g,1}^2) + \sigma_f^2 \ell_{g,1}^2\Big), \\
\tilde{D}_f^2 &:= \left(\ell_{f,0} + \frac{\ell_{g,1}}{\mu_g}\ell_{f,1} + \ell_{g,1}\ell_{f,1}\frac{1}{\mu_g}\right)^2 + \tilde{\sigma}_f^2 = \mathcal{O}(\kappa_g^2),
\end{aligned} \tag{18}$$

where the other constants are provided in Assumptions A and B.

C.1. Descent of Outer Objective

The following lemma characterizes the descent of the outer objective.

Lemma C.2 (Descent Lemma)**.** *Suppose Assumptions A and B hold. Further, assume $\tau_i \geq 1$ and $\alpha_i = \alpha/\tau_i, \forall i \in \mathcal{S}$ for some positive constant α. Then,* FEDOUT *guarantees:*

$$\begin{aligned}
\mathbb{E}\left[f(\boldsymbol{x}^+)\right] - \mathbb{E}\left[f(\boldsymbol{x})\right] \leq & -\frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + \frac{\alpha^2 L_f}{2}\tilde{\sigma}_f^2 \\
& - \frac{\alpha}{2}(1 - \alpha L_f)\,\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i - 1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right] \\
& + \frac{3\alpha}{2}\left(b^2 + M_f^2 \mathbb{E}\left[\|\boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x})\|^2\right] + \frac{M_f^2}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i - 1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right]\right).
\end{aligned} \tag{19}$$

Proof. It follows from Algorithm 2 that $\boldsymbol{x}_{i,0} = \boldsymbol{x}, \forall i \in \mathcal{S}$, and

$$\boldsymbol{x}^+ = \boldsymbol{x} - \frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i - 1} \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+). \tag{20}$$

Now, using the Lipschitz property of ∇f in Lemma C.1, we have

$$\begin{aligned}\mathbb{E}\left[f(\boldsymbol{x}^+)\right]-\mathbb{E}\left[f(\boldsymbol{x})\right] &\le \mathbb{E}\left[\langle \boldsymbol{x}^+-\boldsymbol{x}, \nabla f(\boldsymbol{x})\rangle\right]+\frac{L_f}{2}\mathbb{E}\left[\|\boldsymbol{x}^+-\boldsymbol{x}\|^2\right]\\ &= -\mathbb{E}\left[\left\langle \frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+), \nabla f(\boldsymbol{x})\right\rangle\right]\\ &+\frac{L_f}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right].\end{aligned} \tag{21}$$

In the following, we bound each term on the right hand side (RHS) of (21). For the first term, we have

$$\begin{aligned}-\mathbb{E}\left[\left\langle \frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+), \nabla f(\boldsymbol{x})\right\rangle\right] &= -\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\left\langle \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+), \nabla f(\boldsymbol{x})\right\rangle \mid \mathcal{F}_{i,\nu-1}\right]\right]\\ &= -\mathbb{E}\left[\left\langle \frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+), \nabla f(\boldsymbol{x})\right\rangle\right]\\ &= -\frac{\alpha}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right]-\frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]\\ &+\frac{\alpha}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\nabla f(\boldsymbol{x})\right\|^2\right],\end{aligned} \tag{22}$$

where the first equality follows from the law of total expectation; the second equality uses the fact that $\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+) = \mathbb{E}\left[\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)|\mathcal{F}_{i,\nu-1}\right]$; and the last equality is obtained from our assumption $\alpha_i = \alpha/\tau_i, \forall i \in \mathcal{S}$.

Next, we bound the last term in (22). Note that

$$\begin{aligned}\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\nabla f(\boldsymbol{x})\right\|^2 &= \Big\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}(\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\bar{\nabla} f_i(\boldsymbol{x},\boldsymbol{y}^+))\\ &+\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\nabla} f_i(\boldsymbol{x},\boldsymbol{y}^+)-\nabla f(\boldsymbol{x})\Big\|^2\\ &\le 3\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}(\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\bar{\nabla} f_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+))\right\|^2\\ &+3\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}(\bar{\nabla} f_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\bar{\nabla} f_i(\boldsymbol{x},\boldsymbol{y}^+))\right\|^2\\ &+3\left\|\bar{\nabla} f(\boldsymbol{x},\boldsymbol{y}^+)-\nabla f(\boldsymbol{x})\right\|^2,\end{aligned}$$

where the inequality uses Lemma G.1.

Hence,

$$\begin{aligned}&\mathbb{E}\left[\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\nabla f(\boldsymbol{x})\|^2\right]\\ &\le 3b^2+\frac{3M_f^2}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right]+3M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^*(\boldsymbol{x})\|^2\right],\end{aligned} \tag{23}$$

where the inequality uses Lemmas 2.2 and C.1.

Substituting (23) into (22) yields

$$
\begin{aligned}
&-\mathbb{E}\left[\left\langle \frac{1}{m}\sum_{i=1}^{m}\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+),\nabla f(\boldsymbol{x})\right\rangle\right]\\
&\le -\frac{\alpha}{2}\mathbb{E}\left[\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\|^2\right]-\frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] \qquad (24)\\
&+\frac{3\alpha}{2}\Big(b^2+M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^*(\boldsymbol{x})\|^2\right]+\frac{M_f^2}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right]\Big).
\end{aligned}
$$

Next, we bound the second term on the RHS of (21). Observe that

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right]\\
&=\alpha^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\left(\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)-\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)+\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right)\right\|\right] \qquad (25)\\
&\le\alpha^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right]+\alpha^2\tilde{\sigma}_f^2,
\end{aligned}
$$

where the inequality follows from Lemmas G.3 and C.1.

Plugging (25) and (24) into (21) completes the proof. □

C.2. Error of FEDINN

The following lemma establishes the progress of FEDINN. It should be mentioned that the assumption on $\beta_i, \forall i \in \mathcal{S}$ is identical to the one listed in (Mitra et al., 2021, Theorem 4).

Lemma C.3 (Error of FEDINN)**.** *Suppose Assumptions A and B hold. Further, assume*

$$
\tau_i \ge 1, \quad \alpha_i = \frac{\alpha}{\tau_i}, \quad \beta_i = \frac{\beta}{\tau_i}, \quad \forall i \in \mathcal{S},
$$

where $0 < \beta < \min\left(1/(6\ell_{g,1}), 1\right)$ *and* α *is some positive constant. Then,* FEDINN *guarantees:*

$$
\mathbb{E}\left[\left\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] \le \left(1-\frac{\beta\mu_g}{2}\right)^T\mathbb{E}\left[\left\|\boldsymbol{y}-\boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right]+25T\beta^2\sigma_{g,1}^2, \quad \text{and} \tag{26a}
$$

$$
\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x}^+)\right\|^2\right] &\le a_1(\alpha)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right]\\
&\quad+a_2(\alpha)\mathbb{E}\left[\left\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right]+a_3(\alpha)\tilde{\sigma}_f^2. \qquad (26b)
\end{aligned}
$$

Here,

$$
\begin{aligned}
a_1(\alpha) &:= L_{\boldsymbol{y}}^2\alpha^2+\frac{L_{\boldsymbol{y}}\alpha}{4M_f}+\frac{L_{\boldsymbol{yx}}\alpha^2}{2\eta},\\
a_2(\alpha) &:= 1+4M_fL_{\boldsymbol{y}}\alpha+\frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\alpha^2}{2}, \qquad (27)\\
a_3(\alpha) &:= \alpha^2L_{\boldsymbol{y}}^2+\frac{L_{\boldsymbol{yx}}\alpha^2}{2\eta},
\end{aligned}
$$

for any $\eta > 0$.

Proof. Note that

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x}^+)\|^2\right] =& \mathbb{E}\left[\|\boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x})\|^2\right] + \mathbb{E}\left[\|\boldsymbol{y}^*(\boldsymbol{x}^+) - \boldsymbol{y}^*(\boldsymbol{x})\|^2\right] \\ &+2\mathbb{E}\left[\langle \boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x}), \boldsymbol{y}^*(\boldsymbol{x}) - \boldsymbol{y}^*(\boldsymbol{x}^+)\rangle\right]. \end{aligned} \tag{28}$$

Next, we upper bound each term on the RHS of (28).

Bounding the first term in (28):

From (Mitra et al., 2021, Theorem 4), for all $t \in \{0, \ldots, T-1\}$, we obtain

$$\mathbb{E}[\|\boldsymbol{y}^{t+1} - \boldsymbol{y}^\star(\boldsymbol{x})\|^2] \leq \left(1 - \frac{\beta\mu_g}{2}\right)\mathbb{E}[\|\boldsymbol{y}^t - \boldsymbol{y}^\star(\boldsymbol{x})\|^2] + 25\beta^2\sigma_{g,1}^2,$$

which together with our setting $\boldsymbol{y}^+ = \boldsymbol{y}^T$ implies

$$\mathbb{E}[\|\boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x})\|^2] \leq \left(1 - \frac{\beta\mu_g}{2}\right)^T \mathbb{E}[\|\boldsymbol{y} - \boldsymbol{y}(\boldsymbol{x}^*)\|^2] + 25T\beta^2\sigma_{g,1}^2. \tag{29}$$

Bounding the second term in (28):

By similar steps as in (25), we have

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{y}^*(\boldsymbol{x}^+) - \boldsymbol{y}^*(\boldsymbol{x})\|^2\right] &\leq L_{\boldsymbol{y}}^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1} \boldsymbol{h}_i(\boldsymbol{x}, \boldsymbol{y}^+)\right\|^2\right] \\ &\leq L_{\boldsymbol{y}}^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right] + \alpha^2 L_{\boldsymbol{y}}^2\tilde{\sigma}_f^2, \end{aligned} \tag{30}$$

where the inequalities are obtained from Lemmas C.1 and G.3.

Bounding the third term in (28):

Observe that

$$\begin{aligned}\mathbb{E}\left[\langle \boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x}), \boldsymbol{y}^*(\boldsymbol{x}) - \boldsymbol{y}^*(\boldsymbol{x}^+)\rangle\right] =& -\mathbb{E}\left[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\rangle\right] \\ &- \mathbb{E}\left[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \boldsymbol{y}^\star(\boldsymbol{x}^+) - \boldsymbol{y}^\star(\boldsymbol{x}) - \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\rangle\right]. \end{aligned} \tag{31}$$

For the first term on the R.H.S. of the above equality, we have

$$\begin{aligned}-\mathbb{E}[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\rangle] =& -\mathbb{E}\left[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \frac{1}{m}\nabla\boldsymbol{y}^\star(\boldsymbol{x})\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\rangle\right] \\ \leq& \mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\frac{1}{m}\nabla\boldsymbol{y}^\star(\boldsymbol{x})\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|\right] \\ \leq& L_{\boldsymbol{y}}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\frac{1}{m}\sum_{i=1}^m \alpha_i \sum_{\nu=0}^{\tau_i-1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|\right] \\ \leq& 2\gamma\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] + \frac{L_{\boldsymbol{y}}^2\alpha^2}{8\gamma}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m \frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1} \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right], \end{aligned} \tag{32}$$

where the first equality uses the fact that $\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) = \mathbb{E}\left[\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)|\mathcal{F}_{i,\nu-1}\right]$; the second inequality follows from Lemma C.1; and the last inequality is obtained from the Young's inequality such that $ab \leq 2\gamma a^2 + \frac{b^2}{8\gamma}$.

Further, using Lemma C.1, we have

$$\begin{aligned}&-\mathbb{E}[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \boldsymbol{y}^\star(\boldsymbol{x}^+) - \boldsymbol{y}^\star(\boldsymbol{x}) - \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\rangle] \\ &\leq \mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\boldsymbol{y}^\star(\boldsymbol{x}^+) - \boldsymbol{y}^\star(\boldsymbol{x}) - \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\right\|\right] \\ &\leq \frac{L_{\boldsymbol{yx}}}{2}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right], \end{aligned} \tag{33}$$

where the inequality follows from Lemma C.1.

From Algorithm 2, we have

$$\boldsymbol{x}_{i,0} = \boldsymbol{x}, \forall i \in \mathcal{S}, \quad \text{and} \quad \boldsymbol{x}^+ = \boldsymbol{x} - \frac{1}{m}\sum_{i=1}^{m}\frac{\alpha}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+). \tag{34}$$

Note that $\mathcal{F}_0 = \mathcal{F}_{i,0}$ for all $i \in \mathcal{S}$. Hence,

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2 \left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right] \\
&\le \frac{1}{m}\sum_{i=1}^{m}\frac{\alpha^2}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2 \mathbb{E}\left[\left\|\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2 \mid \mathcal{F}_{i,\tau_i-1}\right]\right] \\
&\le \alpha^2\tilde{D}_f^2\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right],
\end{aligned} \tag{35}$$

where the last inequality uses Lemma C.1.

Note also that for any $\eta > 0$, we have $1 \le \frac{\eta}{2} + \frac{1}{2\eta}$. Combining this inequality with (33) and using (35) give

$$\begin{aligned}
&-\mathbb{E}[\langle \boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x}), \boldsymbol{y}^\star(\boldsymbol{x}^+) - \boldsymbol{y}^\star(\boldsymbol{x}) - \nabla\boldsymbol{y}^\star(\boldsymbol{x})(\boldsymbol{x}^+ - \boldsymbol{x})\rangle] \\
&\le \frac{L_{\boldsymbol{yx}}}{2}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right] \\
&\le \frac{\eta L_{\boldsymbol{yx}}}{4}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|\left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right] + \frac{L_{\boldsymbol{yx}}}{4\eta}\mathbb{E}\left[\left\|\boldsymbol{x}^+ - \boldsymbol{x}\right\|^2\right] \\
&\le \frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\alpha^2}{4}\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right] + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\tilde{\sigma}_f^2,
\end{aligned} \tag{36}$$

where the last inequality uses (35) and Lemma 2.2.

Let $\gamma = M_f L_{\boldsymbol{y}}\alpha$. Plugging (36) and (32) into (31), we have

$$\begin{aligned}
\mathbb{E}[\langle \boldsymbol{y}^+ - \boldsymbol{y}^*(\boldsymbol{x}), \boldsymbol{y}^*(\boldsymbol{x}) - \boldsymbol{y}^*(\boldsymbol{x}^+)\rangle] &\le \left(2\gamma + \frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2}{4}\alpha^2\right)\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] \\
&\quad + \left(\frac{L_{\boldsymbol{y}}^2\alpha^2}{8\gamma} + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\right)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right] + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\tilde{\sigma}_f^2 \\
&= \left(2M_f L_{\boldsymbol{y}}\alpha + \frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2}{4}\alpha^2\right)\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] \\
&\quad + \left(\frac{L_{\boldsymbol{y}}\alpha}{8M_f} + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\right)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2\right] + \frac{L_{\boldsymbol{yx}}\alpha^2}{4\eta}\tilde{\sigma}_f^2.
\end{aligned} \tag{37}$$

Substituting (37), (30), and (29) into (28) completes the proof. □

C.3. Drifting Errors of FEDOUT

The following lemma provides a bound on the *drift* of each $\boldsymbol{x}_{i,\nu}$ from $\boldsymbol{x}$ for stochastic nonconvex bilevel problems. It should be mentioned that similar drifting bounds for *single-level* problems are provided under either strong convexity (Mitra et al., 2021) and/or bounded dissimilarity assumptions (Wang et al., 2020; Reddi et al., 2020; Li et al., 2020b).

Lemma C.4 (Drifting Error of FEDOUT)**.** *Suppose Assumptions A and B hold. Further, assume $\tau_i \ge 1$ and $\alpha_i \le 1/(5M_f\tau_i), \forall i \in \mathcal{S}$. Then, for each $i \in \mathcal{S}$ and $\forall \nu \in \{0, \ldots, \tau_i - 1\}$,* FEDOUT *guarantees:*

$$\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\right\|^2\right] &\le 36\tau_i^2\alpha_i^2\left(M_f^2\mathbb{E}\left[\left\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\right\|^2\right] + \mathbb{E}\left[\left\|\nabla f(\boldsymbol{x})\right\|^2\right] + 3b^2\right) \\
&\quad + 27\tau_i\alpha_i^2\tilde{\sigma}_f^2.
\end{aligned} \tag{38}$$

Proof. The result trivially holds for $\tau_i = 1$. Let $\tau_i > 1$ and define

$$\begin{aligned}
\boldsymbol{v}_{i,\nu} &:= \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\nabla} f_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}^+) \\
&\quad + \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y}^+) + \bar{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{y}^+) - \bar{\nabla} f(\boldsymbol{x}, \boldsymbol{y}^+), \\
\boldsymbol{w}_{i,\nu} &:= \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) + \bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}^+) \\
&\quad - \boldsymbol{h}_i(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{y}^+), \\
\boldsymbol{z}_{i,\nu} &:= \bar{\nabla} f_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y}^+) + \bar{\nabla} f(\boldsymbol{x}, \boldsymbol{y}^+) - \nabla f(\boldsymbol{x}) + \nabla f(\boldsymbol{x}).
\end{aligned} \tag{39}$$

One will notice that

$$\boldsymbol{v}_{i,\nu} + \boldsymbol{w}_{i,\nu} + \boldsymbol{z}_{i,\nu} = \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \boldsymbol{h}_i(\boldsymbol{x}, \boldsymbol{y}^+) + \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^+).$$

Hence, from Algorithm 2, for each $i \in \mathcal{S}$, and $\forall \nu \in \{0, \dots, \tau_i - 1\}$, we have

$$\boldsymbol{x}_{i,\nu+1} - \boldsymbol{x} = \boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{w}_{i,\nu} + \boldsymbol{z}_{i,\nu}), \tag{40}$$

which implies that

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu+1} - \boldsymbol{x}\|^2\right] &= \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{z}_{i,\nu})\|^2\right] + \alpha_i^2 \mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right] \\
&\quad - 2\mathbb{E}\left[\mathbb{E}\left[\langle \boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{z}_{i,\nu}), \alpha_i \boldsymbol{w}_{i,\nu} \rangle \mid \mathcal{F}_{i,\nu-1}\right]\right] \\
&= \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{z}_{i,\nu})\|^2\right] + \alpha_i^2 \mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right].
\end{aligned} \tag{41}$$

Here, the last equality uses Lemma G.3 since $\mathbb{E}[\boldsymbol{w}_{i,\nu} | \mathcal{F}_{i,\nu-1}] = 0$, by definition.

From Lemmas G.1, 2.2, and C.1, for $\boldsymbol{v}_{i,\nu}$, $\boldsymbol{w}_{i,\nu}$, and $\boldsymbol{z}_{i,\nu}$ defined in (39), we have

$$\begin{aligned}
\mathbb{E}\Big[\|\boldsymbol{v}_{i,\nu}\|^2\Big] &\le 3\mathbb{E}\Big[\left\|\bar{\nabla} f_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\right\|^2 \\
&\quad + \left\|\bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}^+) - \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y}^+)\right\|^2 + \left\|\bar{\nabla} f(\boldsymbol{x}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{y}^+)\right\|^2\Big] \\
&\le 9b^2,
\end{aligned} \tag{42a}$$

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right] &\le 3\mathbb{E}\Big[\|\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+)\|^2 \\
&\quad + \|\bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}^+) - \boldsymbol{h}_i(\boldsymbol{x}, \boldsymbol{y}^+)\|^2 + \|\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}^+) - \bar{\boldsymbol{h}}(\boldsymbol{x}, \boldsymbol{y}^+)\|^2\Big] \\
&\le 9\tilde{\sigma}_f^2,
\end{aligned} \tag{42b}$$

and

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{z}_{i,\nu}\|^2\right] &\le 3\mathbb{E}\Big[\|\bar{\nabla} f_i(\boldsymbol{x}_{i,\nu}, \boldsymbol{y}^+) - \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y}^+)\|^2 \\
&\quad + \|\bar{\nabla} f(\boldsymbol{x}, \boldsymbol{y}^+) - \nabla f(\boldsymbol{x})\|^2 + \|\nabla f(\boldsymbol{x})\|^2\Big] \\
&\le 3\left(M_f^2 \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + M_f^2 \mathbb{E}\left[\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\|^2\right] + \mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]\right).
\end{aligned} \tag{42c}$$

Now, the first term in the RHS of (41) can be bounded as follows:

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{z}_{i,\nu})\|^2\right] &\le \left(1 + \frac{1}{2\tau_i - 1}\right) \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 2\tau_i \mathbb{E}\left[\|\alpha_i(\boldsymbol{v}_{i,\nu} + \boldsymbol{z}_{i,\nu})\|^2\right] \\
&\le \left(1 + \frac{1}{2\tau_i - 1}\right) \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 4\tau_i \alpha_i^2 \left(\mathbb{E}\left[\|\boldsymbol{z}_{i,\nu}\|^2\right] + \mathbb{E}\left[\|\boldsymbol{v}_{i,\nu}\|^2\right]\right) \\
&\le \left(1 + \frac{1}{2\tau_i - 1}\right) \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 4\tau_i \alpha_i^2 \left(\mathbb{E}\left[\|\boldsymbol{z}_{i,\nu}\|^2\right] + 9b^2\right) \\
&\le \left(1 + \frac{1}{2\tau_i - 1} + 12\tau_i \alpha_i^2 M_f^2\right) \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] \\
&\quad + 12\tau_i \alpha_i^2 \left(M_f^2 \mathbb{E}\left[\|\boldsymbol{y}^+ - \boldsymbol{y}^\star(\boldsymbol{x})\|^2\right] + \mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 3b^2\right).
\end{aligned} \tag{43}$$

Here, the first inequality follows from Lemma G.2; the second inequality uses Lemma G.1; and the third and last inequalities follow from (42a) and (42c).

Substituting (43) into (41) gives

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu+1}-\boldsymbol{x}\|^2\right] &\leq \left(1+\frac{1}{2\tau_i-1}+12\tau_i\alpha_i^2M_f^2\right)\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right]\\ &\quad+12\tau_i\alpha_i^2\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\|^2\right]+\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]+3b^2\right)+9\alpha_i^2\tilde{\sigma}_f^2\\ &\leq \left(1+\frac{1}{\tau_i-1}\right)\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right]\\ &\quad+12\tau_i\alpha_i^2\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\|^2\right]+\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]+3b^2\right)+9\alpha_i^2\tilde{\sigma}_f^2.\end{aligned} \tag{44}$$

Here, the first inequality uses (42b) and the last inequality follows by noting $\alpha_i \leq 1/(5M_f\tau_i)$.

For all $\tau_i > 1$, we have

$$\begin{aligned}\sum_{j=0}^{\nu-1}\left(1+\frac{1}{\tau_i-1}\right)^j &= \frac{\left(1+\frac{1}{\tau_i-1}\right)^\nu-1}{\left(1+\frac{1}{\tau_i-1}\right)-1}\\ &\leq \tau_i\left(1+\frac{1}{\tau_i}\right)^\nu \leq \tau_i\left(1+\frac{1}{\tau_i}\right)^{\tau_i} \leq \exp(1)\tau_i < 3\tau_i.\end{aligned} \tag{45}$$

Now, iterating equation (44) and using $\boldsymbol{x}_{i,0}=\boldsymbol{x}, \forall i \in \mathcal{S}$, we obtain

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right] &\leq \left(12\tau_i\alpha_i^2\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\|^2\right]+\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]+3b^2\right)+9\alpha_i^2\tilde{\sigma}_f^2\right)\sum_{j=0}^{\nu-1}\left(1+\frac{1}{\tau_i-1}\right)^j\\ &\leq 3\tau_i\left(12\tau_i\alpha_i^2\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\|^2\right]+\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right]+3b^2\right)+9\alpha_i^2\tilde{\sigma}_f^2\right),\end{aligned}$$

where the second inequality uses (45). This completes the proof. □

Remark C.1. *Lemma C.4 shows that the bound on the client-drift scales linearly with τ_i and the inner error $\|\boldsymbol{y}^+-\boldsymbol{y}^\star(\boldsymbol{x})\|^2$ in general nested FL. We aim to control such a drift by selecting $\alpha_i = \mathcal{O}(1/\tau_i)$ for all $i \in \mathcal{S}$ and using the inner error bound provided in Lemma C.3.*

Next, we provide the proof of our main result which can be adapted to general nested problems (bilevel, min-max, compositional).

C.4. Proof of Theorem 3.1

Proof. We define the following Lyapunov function

$$\mathbb{W}^k := f(\boldsymbol{x}^k)+\frac{M_f}{L_{\boldsymbol{y}}}\|\boldsymbol{y}^k-\boldsymbol{y}^\star(\boldsymbol{x}^k)\|^2. \tag{46}$$

Motivated by (Chen et al., 2021a), we bound the difference between two Lyapunov functions. That is,

$$\mathbb{W}^{k+1}-\mathbb{W}^k = f(\boldsymbol{x}^{k+1})-f(\boldsymbol{x}^k)+\frac{M_f}{L_{\boldsymbol{y}}}\left(\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^*(\boldsymbol{x}^{k+1})\|^2-\|\boldsymbol{y}^k-\boldsymbol{y}^\star(\boldsymbol{x}^k)\|^2\right). \tag{47}$$

The first two terms on the RHS of (47) quantifies the descent of outer objective f and the reminding terms measure the descent of the inner errors.

From our assumption, we have $\alpha_i^k=\alpha_k/\tau_i$, $\beta_i^k=\beta_k/\tau_i$, $\forall i \in \mathcal{S}$. Substituting these stepsizes into the bounds provided in

Lemmas C.2 and C.3, and using (47), we get

$$\begin{aligned}\mathbb{E}[\mathbb{W}^{k+1}]-\mathbb{E}[\mathbb{W}^k] \le& \left(\frac{L_f\alpha_k^2}{2}+\frac{M_f}{L_{\boldsymbol{y}}}a_3(\alpha_k)\right)\tilde{\sigma}_f^2+\frac{3\alpha_k}{2}b^2,\\ &-\frac{\alpha_k}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right]+\frac{3M_f^2\alpha_k}{2m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}^k-\boldsymbol{x}^k\|^2\right] &\text{(48a)}\\ &-\left(\frac{\alpha_k}{2}-\frac{L_f\alpha_k^2}{2}-\frac{M_f}{L_{\boldsymbol{y}}}a_1(\alpha_k)\right)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\bar{\boldsymbol{h}}_i(\boldsymbol{x}_{i,\nu}^k,\boldsymbol{y}^{k+1})\right\|^2\right] &\text{(48b)}\\ &+\frac{M_f}{L_{\boldsymbol{y}}}\left(\frac{3M_fL_{\boldsymbol{y}}\alpha_k}{2}+a_2(\alpha_k)\right)\mathbb{E}\left[\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^*(\boldsymbol{x}^k)\|^2\right]-\frac{M_f}{L_{\boldsymbol{y}}}\mathbb{E}\left[\|\boldsymbol{y}^k-\boldsymbol{y}^*(\boldsymbol{x}^k)\|^2\right], &\text{(48c)}\end{aligned}$$

where $a_1(\alpha)-a_3(\alpha)$ are defined in (27).

Let

$$\alpha_i^k=\frac{\alpha_k}{\tau_i},\ \ \forall i\in\mathcal{S}\ \ \text{where}\ \ \alpha_k\le\frac{1}{216M_f^2+5M_f}. \tag{49}$$

The above choice of α_k satisfies the condition of Lemma C.4 and we have $54M_f^2\alpha_k^3\le\alpha_k^2/4$. Hence, from Lemma C.4, we get

$$\begin{aligned}\text{(48a)}\le&-\frac{\alpha_k}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right]+\frac{81}{2}\alpha_k^3M_f^2\tilde{\sigma}_f^2\\ &+54M_f^2\alpha_k^3\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^\star(\boldsymbol{x}^k)\|^2\right]+\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right]+3b^2\right)\\ \le&-\frac{\alpha_k}{4}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right]+\frac{\alpha_k^2}{4}\tilde{\sigma}_f^2+\frac{\alpha_k^2}{4}\left(M_f^2\mathbb{E}\left[\|\boldsymbol{y}^{k+1}-\boldsymbol{y}^\star(\boldsymbol{x}^k)\|^2\right]+3b^2\right)\\ \le&-\frac{\alpha_k}{4}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right]+\frac{\alpha_k^2}{4}\tilde{\sigma}_f^2+\frac{3\alpha_k^2}{4}b^2\\ &+\frac{25M_f}{L_{\boldsymbol{y}}}\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2\right)T\beta_k^2\sigma_{g,1}^2+\frac{M_f}{L_{\boldsymbol{y}}}\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2\right)\left(1-\frac{\beta_k\mu_g}{2}\right)^T,\end{aligned} \tag{50}$$

where the first inequlaity uses (49) and the last inequality follows from (26a).

To guarantee the descent of $\mathbb{W}^k$, the following constraints need to be satisfied

$$\begin{aligned}&\text{(48b)}\le 0,\\ &\implies \frac{\alpha_k}{2}-\frac{L_f\alpha_k^2}{2}-\frac{M_f}{L_{\boldsymbol{y}}}\left(L_{\boldsymbol{y}}^2\alpha_k^2+\frac{L_{\boldsymbol{y}}\alpha_k}{4M_f}+\frac{L_{\boldsymbol{yx}}\alpha_k^2}{2\eta}\right)\ge 0,\\ &\implies \alpha_k\le\frac{1}{2L_f+4M_fL_{\boldsymbol{y}}+\frac{2M_fL_{\boldsymbol{yx}}}{L_{\boldsymbol{y}}\eta}},\end{aligned} \tag{51}$$

where the second line uses (27).

Further, substituting (26) in (48c) gives

$$\begin{aligned}\text{(48c)}\le&\frac{25M_f}{L_{\boldsymbol{y}}}\left(\frac{3M_fL_{\boldsymbol{y}}\alpha_k}{2}+a_2(\alpha_k)\right)T\beta_k^2\sigma_{g,1}^2\\ &+\frac{M_f}{L_{\boldsymbol{y}}}\left(\left(\frac{3M_fL_{\boldsymbol{y}}\alpha_k}{2}+a_2(\alpha_k)\right)\left(1-\frac{\beta_k\mu_g}{2}\right)^T-1\right)\mathbb{E}[\|\boldsymbol{y}^k-\boldsymbol{y}^*(\boldsymbol{x}^k)\|^2].\end{aligned} \tag{52}$$

Substituting (50)–(52) into (48) gives

$$\begin{aligned}\mathbb{E}[\mathbb{W}^{k+1}]-\mathbb{E}[\mathbb{W}^k] \le & -\frac{\alpha_k}{4}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] \\ & + \left(\frac{3}{2}\alpha_k+\frac{3}{4}\alpha_k^2\right) b^2 \\ & + \left(\left(\frac{L_f}{2}+\frac{1}{4}\right)\alpha_k^2+\frac{M_f}{L_{\boldsymbol{y}}}a_3(\alpha_k)\right)\tilde{\sigma}_f^2 \\ & + \frac{25M_f}{L_{\boldsymbol{y}}}\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2+\frac{3M_fL_{\boldsymbol{y}}\alpha_k}{2}+a_2(\alpha_k)\right)T\beta_k^2\sigma_{g,1}^2 \\ & + \frac{M_f}{L_{\boldsymbol{y}}}\left(\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2+\frac{3M_fL_{\boldsymbol{y}}\alpha_k}{2}+a_2(\alpha_k)\right)\left(1-\frac{\beta_k\mu_g}{2}\right)^T-1\right)\mathbb{E}[\|\boldsymbol{y}^k-\boldsymbol{y}^*(\boldsymbol{x}^k)\|^2]. \end{aligned} \tag{53a}$$

Let $\beta_k < \min\left(1/(6\ell_{g,1}),1\right)$. Then, we have $\beta_k\mu_g/2<1$. This together with (27) implies that for any $\alpha_k>0$

$$\begin{aligned} & (53a)\le 0, \\ \implies & \left(1+\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2+\frac{11M_fL_{\boldsymbol{y}}\alpha_k}{2}+\frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\alpha_k^2}{2}\right)\left(1-\frac{\beta_k\mu_g}{2}\right)^T-1\le 0, \\ \implies & \exp\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2+\frac{11M_fL_{\boldsymbol{y}}\alpha_k}{2}+\frac{\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\alpha_k^2}{2}\right)\exp\left(-\frac{T\beta_k\mu_g}{2}\right)-1\le 0, \\ \implies & \beta_k \ge \frac{11M_fL_{\boldsymbol{y}}+\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\alpha_k+\frac{M_fL_{\boldsymbol{y}}\alpha_k}{2}}{\mu_g}\cdot\frac{\alpha_k}{T}. \end{aligned} \tag{54}$$

From (49), (51) and (54), we select

$$\alpha_k=\min\{\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3,\frac{\bar{\alpha}}{\sqrt{K}}\},\qquad \beta_k=\frac{\bar{\beta}\alpha_k}{T}, \tag{55}$$

where

$$\begin{aligned} \bar{\beta} &:= \frac{1}{\mu_g}\left(11M_fL_{\boldsymbol{y}}+\eta L_{\boldsymbol{yx}}\tilde{D}_f^2\bar{\alpha}_1+\frac{M_fL_{\boldsymbol{y}}\bar{\alpha}_1}{2}\right), \\ \bar{\alpha}_1 &:= \frac{1}{2L_f+4M_fL_{\boldsymbol{y}}+\frac{2M_fL_{\boldsymbol{yx}}}{L_{\boldsymbol{y}}\eta}},\quad \bar{\alpha}_2:=\frac{T}{8\ell_{g,1}\bar{\beta}},\quad \bar{\alpha}_3:=\frac{1}{216M_f^2+5M_f}, \end{aligned} \tag{56}$$

With the above choice of stepsizes, (53) can be simplified as

$$\begin{aligned}\mathbb{E}[\mathbb{W}^{k+1}]-\mathbb{E}[\mathbb{W}^k] \le & -\frac{\alpha_k}{4}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] \\ & + \left(\frac{3}{2}\alpha_k+\frac{3}{4}\alpha_k^2\right) b^2 \\ & + \left(\frac{L_f+\frac{1}{2}}{2}\alpha_k^2+\frac{M_f}{L_{\boldsymbol{y}}}a_3(\alpha_k)\right)\tilde{\sigma}_f^2 \\ & + \frac{25M_f}{L_{\boldsymbol{y}}}\left(\frac{M_fL_{\boldsymbol{y}}}{4}\alpha_k^2+\frac{3M_fL_{\boldsymbol{y}}}{2}\alpha_k+a_2(\alpha_k)\right)T\beta_k^2\sigma_{g,1}^2 \\ & \le -\frac{\alpha_k}{4}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2]+c_1\alpha_k^2\sigma_{g,1}^2+\left(\frac{3}{2}\alpha_k+\frac{3}{4}\alpha_k^2\right)b^2+c_2\alpha_k^2\tilde{\sigma}_f^2, \end{aligned} \tag{57}$$

where the constants c_1 and c_2 are defined as

$$\begin{aligned} c_1 &= \frac{25M_f}{L_{\boldsymbol{y}}}\left(1+\frac{11M_fL_{\boldsymbol{y}}}{2}\bar{\alpha}_1+\left(\frac{M_fL_{\boldsymbol{y}}+2\eta L_{\boldsymbol{yx}}\tilde{D}_f^2}{4}\right)\bar{\alpha}_1^2\right)\bar{\beta}^2\frac{1}{T}, \\ c_2 &= \frac{L_f+\frac{1}{2}}{2}+M_fL_{\boldsymbol{y}}+\frac{L_{\boldsymbol{yx}}M_f}{4\eta L_{\boldsymbol{y}}}. \end{aligned} \tag{58}$$

Then telescoping gives

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] &\le \frac{4}{\sum_{k=0}^{K-1}\alpha_k}\left(\Delta_{\mathbb{W}}+\sum_{k=0}^{K-1}\frac{3}{2}\left(\alpha_k+\frac{\alpha_k^2}{2}\right)b^2+c_1\alpha_k^2\sigma_{g,1}^2+c_2\alpha_k^2\tilde{\sigma}_f^2\right)\\ &\le \frac{4\Delta_{\mathbb{W}}}{\min\{\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3\}K}+\frac{4\Delta_{\mathbb{W}}}{\bar{\alpha}\sqrt{K}}+6\left(1+\frac{\bar{\alpha}}{2\sqrt{K}}\right)b^2+\frac{4c_1\bar{\alpha}}{\sqrt{K}}\sigma_{g,1}^2+\frac{4c_2\bar{\alpha}}{\sqrt{K}}\tilde{\sigma}_f^2 \qquad (59)\\ &= \mathcal{O}\left(\frac{1}{\min\{\bar{\alpha}_1,\bar{\alpha}_2,\bar{\alpha}_3\}K}+\frac{\bar{\alpha}\max(\sigma_{g,1}^2,\sigma_{g,2}^2,\sigma_f^2)}{\sqrt{K}}+b^2\right),\end{aligned}$$

where $\Delta_{\mathbb{W}} := \mathbb{W}^0 - \mathbb{E}[\mathbb{W}^K]$. □

C.5. Proof of Corollary 3.1

Proof. Let $\eta = \frac{M_f}{L_{\boldsymbol{y}}} = \mathcal{O}(\kappa_g)$ in (58). It follows from (18), (56), and (58) that

$$\bar{\alpha}_1 = \mathcal{O}(\kappa_g^{-3}),\ \ \bar{\alpha}_2 = \mathcal{O}(T\kappa_g^{-3}),\ \ \bar{\alpha}_3 = \mathcal{O}(\kappa_g^{-4}),\ \ c_1 = \mathcal{O}(\kappa_g^9/T),\ \ c_2 = \mathcal{O}(\kappa_g^3). \qquad (60)$$

Further, $N = \mathcal{O}(\kappa_g \log K)$ gives $b = \frac{1}{K^{1/4}}$. Now, if we select $\bar{\alpha} = \mathcal{O}(\kappa_g^{-2.5})$ and $T = \mathcal{O}(\kappa_g^4)$, Eq. (59) gives

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] = \mathcal{O}\left(\frac{\kappa_g^4}{K}+\frac{\kappa_g^{2.5}}{\sqrt{K}}\right).$$

To achieve ε-optimal solution, we need $K = \mathcal{O}(\kappa_g^5\varepsilon^{-2})$, and the samples in ξ and ζ are $\mathcal{O}(\kappa_g^5\varepsilon^{-2})$ and $\mathcal{O}(\kappa_g^9\varepsilon^{-2})$, respectively. □

D. Proof for Federated Minimax Optimization

Note that the minimax optimization problem (2) has the following bilevel form

$$\begin{aligned}\min_{\boldsymbol{x}\in\mathbb{R}^{d_1}}\quad & f(\boldsymbol{x}) = \tfrac{1}{m}\textstyle\sum_{i=1}^m f_i\left(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})\right)\\ \text{subj. to}\quad & \boldsymbol{y}^*(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^{d_2}}\ -\tfrac{1}{m}\textstyle\sum_{i=1}^m f_i\left(\boldsymbol{x},\boldsymbol{y}\right).\end{aligned} \qquad (61a)$$

Here,

$$f_i(\boldsymbol{x},\boldsymbol{y}) = \mathbb{E}_{\xi\sim\mathcal{C}_i}[f_i(\boldsymbol{x},\boldsymbol{y};\xi)] \qquad (61b)$$

is the loss functions of the i^{th} client.

In this case, the hypergradient of (61) is

$$\nabla f_i(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f_i\big(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})\big) + \nabla_{\boldsymbol{x}}\boldsymbol{y}^*(\boldsymbol{x})^\top\nabla_{\boldsymbol{y}} f_i\big(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})\big) = \nabla_{\boldsymbol{x}} f_i\big(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})\big), \qquad (62)$$

where the second equality follows from the optimality condition of the inner problem, i.e., $\nabla_{\boldsymbol{y}} f(\boldsymbol{x},\boldsymbol{y}^*(\boldsymbol{x})) = 0$.

For each $i \in \mathcal{S}$, we can approximate $\nabla f_i(\boldsymbol{x})$ on a vector $\boldsymbol{y}$ in place of $\boldsymbol{y}^*(\boldsymbol{x})$, denoted as $\overline{\nabla} f_i(\boldsymbol{x},\boldsymbol{y}) := \nabla_{\boldsymbol{x}} f_i\big(\boldsymbol{x},\boldsymbol{y}\big)$. We also note that in the minimax case $\boldsymbol{h}_i$ is an unbiased estimator of $\bar{\nabla} f_i(\boldsymbol{x},\boldsymbol{y})$. Thus, $b = 0$. Therefore, we can apply FEDNEST using

$$\begin{aligned}\boldsymbol{q}_{i,\nu} &= -\nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y}_{i,\nu};\xi_{i,\nu}) + \nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y};\xi_{i,\nu}) - \frac{1}{m}\sum_{i=1}^m \nabla_{\boldsymbol{y}} f_i(\boldsymbol{x},\boldsymbol{y};\xi_i),\\ \boldsymbol{h}_{i,\nu} &= \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+;\xi_{i,\nu}) - \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x},\boldsymbol{y}^+;\xi_{i,\nu}) + \frac{1}{m}\sum_{i=1}^m \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x},\boldsymbol{y}^+;\xi_i).\end{aligned} \qquad (63)$$

D.1. Supporting Lemmas

Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{d_1+d_2}$. We make the following assumptions that are counterparts of Assumptions A and B.

Assumption E. *For all $i \in [m]$:*

(E1) *$f_i(\boldsymbol{z}), \nabla f_i(\boldsymbol{z}), \nabla^2 f_i(\boldsymbol{z})$ are respectively $\ell_{f,0}$, $\ell_{f,1}, \ell_{f,2}$-Lipschitz continuous; and*

(E2) *$f_i(\boldsymbol{x}, \boldsymbol{y})$ is μ_f-strongly convex in $\boldsymbol{y}$ for any fixed $\boldsymbol{x} \in \mathbb{R}^{d_1}$.*

We use $\kappa_f = \ell_{f,1}/\mu_f$ to denote the condition number of the inner objective with respect to $\boldsymbol{y}$.

Assumption F. *For all $i \in [m]$:*

(F1) *$\nabla f_i(\boldsymbol{z}; \xi)$ is unbiased estimators of $\nabla f_i(\boldsymbol{z})$; and*

(F2) *Its variance is bounded, i.e., $\mathbb{E}_\xi[\|\nabla f_i(\boldsymbol{z}; \xi) - \nabla f_i(\boldsymbol{z})\|^2] \le \sigma_f^2$, for some σ_f^2.*

In the following, we re-derive Lemma C.1 for the finite-sum minimax problem (61).

Lemma D.1. *Under Assumptions E and F, we have $\bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}) = \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y})$ for all $i \in \mathcal{S}$ and (17a)–(17g) hold with*

$$\begin{aligned} L_{yx} &= \frac{\ell_{f,2} + \ell_{f,2} L_{\boldsymbol{y}}}{\mu_f} + \frac{\ell_{f,1}(\ell_{f,2} + \ell_{f,2} L_{\boldsymbol{y}})}{\mu_f^2} = \mathcal{O}(\kappa_f^3), \\ M_f &= \ell_{f,1} = \mathcal{O}(1), \quad L_f = (\ell_{f,1} + \frac{\ell_{f,1}^2}{\mu_f}) = \mathcal{O}(\kappa_f), \\ L_{\boldsymbol{y}} &= \frac{\ell_{f,1}}{\mu_f} = \mathcal{O}(\kappa_f), \quad \tilde{\sigma}_f^2 = \sigma_f^2, \quad \tilde{D}_f^2 = \ell_{l,0}^2 + \sigma_f^2, \end{aligned} \tag{64}$$

where $\ell_{f,0}$, $\ell_{f,1}, \ell_{f,2}$, μ_f, and σ_f are given in Assumptions E and F.

D.2. Proof of Corollary 3.2

Proof. Let $\eta = 1$. From (55) and (56), we have

$$\alpha_k = \min\left\{\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \frac{\bar{\alpha}}{\sqrt{K}}\right\}, \qquad \beta_k = \frac{\bar{\beta}\alpha_k}{T}, \tag{65a}$$

where

$$\begin{aligned} \bar{\beta} &= \frac{1}{\mu_g}\left(11\ell_{f,1}L_{\boldsymbol{y}} + L_{yx}\tilde{D}_f^2\bar{\alpha}_1 + \frac{\ell_{f,1}L_{\boldsymbol{y}}\bar{\alpha}_1}{2}\right), \\ \bar{\alpha}_1 &= \frac{1}{2L_f + 4\ell_{f,1}L_{\boldsymbol{y}} + \frac{2\ell_{f,1}L_{yx}}{L_{\boldsymbol{y}}}}, \quad \bar{\alpha}_2 = \frac{T}{8\ell_{g,1}\bar{\beta}}, \quad \bar{\alpha}_3 = \frac{1}{216\ell_{f,1}^2 + 5\ell_{f,1}}. \end{aligned} \tag{65b}$$

Using the above choice of stepsizes, (59) reduces to

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] \le \frac{4\Delta_{\mathbb{W}}}{K\min\{\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3\}} + \frac{4\Delta_{\mathbb{W}}}{\bar{\alpha}\sqrt{K}} + \frac{4(c_1+c_2)\bar{\alpha}}{\sqrt{K}}\sigma_f^2, \tag{66}$$

where $\Delta_{\mathbb{W}} = \mathbb{W}^0 - \mathbb{E}[\mathbb{W}^K]$,

$$\begin{aligned} c_1 &= \frac{25\ell_{f,1}}{L_{\boldsymbol{y}}}\left(1 + \frac{11\ell_{f,1}L_{\boldsymbol{y}}}{2}\bar{\alpha}_1 + \left(\frac{\ell_{f,1}L_{\boldsymbol{y}} + 2L_{\boldsymbol{yx}}\tilde{D}_f^2}{4}\right)\bar{\alpha}_1^2\right)\bar{\beta}^2\frac{1}{T}, \\ c_2 &= \frac{L_f + \frac{1}{2}}{2} + \ell_{f,1}L_{\boldsymbol{y}} + \frac{L_{\boldsymbol{yx}}\ell_{f,1}}{4L_{\boldsymbol{y}}}. \end{aligned} \tag{67}$$

Let $\bar{\alpha} = \mathcal{O}(\kappa_f^{-1})$. Since by our assumption, $T = \mathcal{O}(\kappa_f)$, it follows from (64) and (73) that

$$\bar{\alpha}_1 = \mathcal{O}(\kappa_f^{-2}),\ \ \bar{\alpha}_2 = \mathcal{O}(\kappa_f^{-1}),\ \ \bar{\alpha}_3 = \mathcal{O}(1),\ \ c_1 = \mathcal{O}(\kappa_f^2),\ \ c_2 = \mathcal{O}(\kappa_f^2). \tag{68}$$

Substituting (68) in (66) and (67) gives

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] = \mathcal{O}\left(\frac{\kappa_f^2}{K} + \frac{\kappa_f}{\sqrt{K}}\right). \tag{69}$$

To achieve ε-accuracy, we need $K = \mathcal{O}(\kappa_f^2\varepsilon^{-2})$. □

E. Proof for Federated Compositional Optimization

Note that in the stochastic compositional problem (3), the inner function $f_i(\boldsymbol{x}, \boldsymbol{y}; \xi) = f_i(\boldsymbol{y}; \xi)$ for all $i \in \mathcal{S}$, and the outer function is $g_i(\boldsymbol{x}, \boldsymbol{y}; \zeta) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{r}_i(\boldsymbol{x}; \zeta)\|^2$, for all $i \in \mathcal{S}$. In this case, we have

$$\nabla_{\boldsymbol{y}} g_i(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}_i - \boldsymbol{r}_i(\boldsymbol{x}; \zeta),\ \ \nabla_{\boldsymbol{yy}} g(\boldsymbol{x}, \boldsymbol{y}; \zeta) = \mathbf{I}_{d_2\times d_2},\ \text{ and }\ \nabla_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}; \zeta) = -\frac{1}{m}\sum_{i=1}^m \nabla \boldsymbol{r}_i(\boldsymbol{x}; \zeta)^\top. \tag{70}$$

Hence, the hypergradient of (3) has the following form

$$\begin{aligned}\nabla f_i(\boldsymbol{x}) &= \nabla_{\boldsymbol{x}} f_i\left(\boldsymbol{y}^*(\boldsymbol{x})\right)\\ &\quad - \nabla^2_{\boldsymbol{xy}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))[\nabla^2_{\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y}^*(\boldsymbol{x}))]^{-1}\nabla_{\boldsymbol{y}} f_i\left(\boldsymbol{y}^*(\boldsymbol{x})\right)\\ &= (\frac{1}{m}\sum_{i=1}^m \nabla \boldsymbol{r}_i(\boldsymbol{x}))^\top \nabla_{\boldsymbol{y}} f_i(\boldsymbol{y}^*(\boldsymbol{x})).\end{aligned} \tag{71}$$

We can obtain an approximate gradient $\nabla f_i(\boldsymbol{x})$ by replacing $\boldsymbol{y}^\star(\boldsymbol{x})$ with $\boldsymbol{y}$; that is $\bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y}) = (\frac{1}{m}\sum_{i=1}^m \nabla \boldsymbol{r}_i(\boldsymbol{x}))^\top \nabla_{\boldsymbol{y}} f_i(\boldsymbol{y})$. It should be mentioned that in the compositional case $b = 0$. Thus, we can apply FEDNEST and LFEDNEST using the above gradient approximations.

E.1. Supporting Lemmas

Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^{d_1+d_2}$. We make the following assumptions that are counterparts of Assumptions A and B.

Assumption G. *For all $i \in [m]$, $f_i(\boldsymbol{z}), \nabla f_i(\boldsymbol{z}), \boldsymbol{r}_i(\boldsymbol{z}), \nabla \boldsymbol{r}_i(\boldsymbol{z})$ are respectively $\ell_{f,0}$, $\ell_{f,1}, \ell_{\boldsymbol{r},0}, \ell_{\boldsymbol{r},1}$-Lipschitz continuous.*

Assumption H. *For all $i \in [m]$:*

(H1) *$\nabla f_i(\boldsymbol{z}; \xi)$, $\boldsymbol{r}_i(\boldsymbol{x}; \zeta)$, $\nabla \boldsymbol{r}_i(\boldsymbol{x}; \zeta)$ are unbiased estimators of $\nabla f_i(\boldsymbol{z})$, $\boldsymbol{r}_i(\boldsymbol{x})$, and $\nabla \boldsymbol{r}_i(\boldsymbol{x})$.*

(H2) *Their variances are bounded, i.e., $\mathbb{E}_\xi[\|\nabla f_i(\boldsymbol{z}; \xi) - \nabla f_i(\boldsymbol{z})\|^2] \le \sigma_f^2$, $\mathbb{E}_\zeta[\|\boldsymbol{r}_i(\boldsymbol{z}; \zeta) - \boldsymbol{r}_i(\boldsymbol{z})\|^2] \le \sigma^2_{\boldsymbol{r},0}$, and $\mathbb{E}_\zeta[\|\nabla \boldsymbol{r}_i(\boldsymbol{z}; \zeta) - \nabla \boldsymbol{r}_i(\boldsymbol{z})\|^2] \le \sigma^2_{\boldsymbol{r},1}$ for some $\sigma_f^2, \sigma^2_{\boldsymbol{r},0}$, and $\sigma^2_{\boldsymbol{r},1}$.*

The following lemma is the counterpart of Lemma C.1. The proof is similar to (Chen et al., 2021a, Lemma 7).

Lemma E.1. *Under Assumptions G and H, we have $\bar{\boldsymbol{h}}_i(\boldsymbol{x}, \boldsymbol{y}) = \bar{\nabla} f_i(\boldsymbol{x}, \boldsymbol{y})$ for all $i \in \mathcal{S}$, and (17a)–(17g) hold with*

$$\begin{aligned}&M_f = \ell_{\boldsymbol{r},0}\ell_{f,1},\ \ \ L_{\boldsymbol{y}} = \ell_{\boldsymbol{r},0},\ \ \ L_f = \ell^2_{\boldsymbol{r},0}\ell_{f,1} + \ell_{f,0}\ell_{\boldsymbol{r},1},\ \ L_{\boldsymbol{yx}} = \ell_{\boldsymbol{r},1},\\ &\tilde{\sigma}_f^2 = \ell^2_{\boldsymbol{r},0}\sigma_f^2 + (\ell^2_{f,0} + \sigma_f^2)\sigma^2_{\boldsymbol{r},1},\ \ \ \tilde{D}_f^2 = (\ell^2_{f,0} + \sigma_f^2)(\ell^2_{\boldsymbol{r},0} + \sigma^2_{\boldsymbol{r},1}).\end{aligned} \tag{72}$$

E.2. Proof of Corrollary 3.3

Proof. By our assumption $T = 1$. Let $\bar{\alpha} = 1$ and $\eta = 1/L_{\boldsymbol{yx}}$. From (55) and (56), we obtain

$$\alpha_k = \min\left\{\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \frac{1}{\sqrt{K}}\right\},\qquad \beta_k = \bar{\beta}\alpha_k, \tag{73a}$$

where

$$\begin{aligned}\bar{\beta} &= \frac{1}{\mu_g}\left(11\ell_{f,1}\ell_{\boldsymbol{r},0}^2 + \tilde{D}_f^2\bar{\alpha}_1 + \frac{\ell_{f,1}\ell_{\boldsymbol{r},0}^2\bar{\alpha}_1}{2}\right),\\ \bar{\alpha}_1 &= \frac{1}{2\ell_{f,0}\ell_{\boldsymbol{r},1} + 6\ell_{f,1}\ell_{\boldsymbol{r},0}^2 + 2\ell_{f,1}\ell_{\boldsymbol{r},1}^2},\quad \bar{\alpha}_2 = \frac{1}{8\ell_{\boldsymbol{r},0}\bar{\beta}},\quad \bar{\alpha}_3 = \frac{1}{216(\ell_{\boldsymbol{r},0}\ell_{f,1})^2 + 5\ell_{\boldsymbol{r},0}\ell_{f,1}}.\end{aligned} \tag{73b}$$

Then, using (59), we obtain

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^k)\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{K}}\right). \tag{74}$$

This completes the proof. □

F. Proof for Federated Single-Level Optimization

Next, we re-derive Lemmas C.2 and C.4 for single-level nonconvex FL under Assumptions C and D.

Lemma F.1 (Counterpart of Lemma C.2)**.** *Suppose Assumptions C and D hold. Further, assume $\tau_i \geq 1$ and $\alpha_i = \alpha/\tau_i, \forall i \in \mathcal{S}$ for some positive constant α. Then,* FEDOUT *guarantees:*

$$\begin{aligned}\mathbb{E}\left[f(\boldsymbol{x}^+)\right] - \mathbb{E}\left[f(\boldsymbol{x})\right] \leq& -\frac{\alpha}{2}(1-\alpha L_f)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu})\right\|^2\right]\\ &- \frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + \frac{\alpha L_f^2}{2m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right] + \frac{\alpha^2 L_f}{2}\sigma_f^2.\end{aligned} \tag{75}$$

Proof. By applying Algorithm 2 to the single-level optimization problem (13), we have

$$\boldsymbol{x}_{i,0} = \boldsymbol{x}\ \ \forall i \in \mathcal{S},\ \ \boldsymbol{x}^+ = \boldsymbol{x} - \frac{1}{m}\sum_{i=1}^m\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}).$$

where

$$\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}) := \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x}_{i,\nu};\xi_{i,\nu}) - \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x};\xi_{i,\nu}) + \frac{1}{m}\sum_{i=1}^m\nabla_{\boldsymbol{x}} f_i(\boldsymbol{x};\xi_i).$$

This together with Assumption C implies that

$$\begin{aligned}\mathbb{E}\left[f(\boldsymbol{x}^+)\right] - \mathbb{E}\left[f(\boldsymbol{x})\right] \leq& \mathbb{E}\left[\langle\boldsymbol{x}^+ - \boldsymbol{x}, \nabla f(\boldsymbol{x})\rangle\right] + \frac{L_f}{2}\mathbb{E}\left[\|\boldsymbol{x}^+ - \boldsymbol{x}\|^2\right]\\ =& -\mathbb{E}\left[\left\langle\frac{1}{m}\sum_{i=1}^m\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}), \nabla f(\boldsymbol{x})\right\rangle\right]\\ &+ \frac{L_f}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu})\right\|^2\right].\end{aligned} \tag{76}$$

For the first term on the RHS of (76), we obtain

$$\begin{aligned}
-\mathbb{E}\left[\left\langle \frac{1}{m}\sum_{i=1}^{m}\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}),\nabla f(\boldsymbol{x})\right\rangle\right] &= -\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\frac{\alpha}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\langle \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}),\nabla f(\boldsymbol{x})\rangle \mid \mathcal{F}_{i,\nu-1}\right]\right] \\
&= -\frac{\alpha}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu})\right\|^2\right] - \frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] \\
&\quad + \frac{\alpha}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu}) - \nabla f(\boldsymbol{x})\right\|^2\right], \\
&= -\frac{\alpha}{2}\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu})\right\|^2\right] - \frac{\alpha}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] \\
&\quad + \frac{\alpha L_f^2}{2m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right],
\end{aligned} \tag{77}$$

where the first equality follows from the law of total expectation and the last inequality is obtained from Assumption C.

For the second term on the RHS of (76), Assumption D gives

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\alpha_i\sum_{\nu=0}^{\tau_i-1}\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu},\boldsymbol{y}^+)\right\|^2\right] &= \alpha^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\left(\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}) - \nabla f_i(\boldsymbol{x}_{i,\nu}) + \nabla f_i(\boldsymbol{x}_{i,\nu})\right)\right\|^2\right] \\
&\le \alpha^2\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu})\right\|^2\right] + \alpha^2\sigma_f^2.
\end{aligned} \tag{78}$$

Plugging (78) and (77) into (76) gives the desired result. □

Lemma F.2 (Counterpart of Lemma C.4)**.** *Suppose Assumptions C and D hold. Further, assume $\tau_i \ge 1$ and $\alpha_i = \alpha/\tau_i, \forall i \in \mathcal{S}$, where $\alpha \le 1/(3L_f)$. Then, for all $\nu \in \{0,\dots,\tau_i-1\}$,* FEDOUT *gives*

$$\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}\|^2\right] \le 12\tau_i^2\alpha_i^2\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 27\tau_i\alpha_i^2\sigma_f^2. \tag{79}$$

Proof. The result trivially holds for $\tau_i = 1$. Similar to what is done in the proof of Lemma C.4, let $\tau_i > 1$ and define

$$\begin{aligned}
\boldsymbol{v}_{i,\nu} &:= \nabla f_i(\boldsymbol{x}_{i,\nu}) - \nabla f_i(\boldsymbol{x}) + \nabla f(\boldsymbol{x}), \\
\boldsymbol{w}_{i,\nu} &:= \boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}) - \nabla f_i(\boldsymbol{x}_{i,\nu}) + \nabla f_i(\boldsymbol{x}) - \boldsymbol{h}_i(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x}) - \nabla f(\boldsymbol{x}).
\end{aligned} \tag{80}$$

where $\boldsymbol{h}_i(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x};\xi_{i,\nu})$ and $\boldsymbol{h}(\boldsymbol{x}) = 1/m\sum_{i=1}^m \nabla_{\boldsymbol{x}} f_i(\boldsymbol{x};\xi_i)$.

From Algorithm 2, for each $i \in \mathcal{S}$, and $\forall \nu \in \{0,\dots,\tau_i-1\}$, we obtain

$$\begin{aligned}
\boldsymbol{x}_{i,\nu+1} - \boldsymbol{x} &= \boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i\left(\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}) - \boldsymbol{h}_i(\boldsymbol{x}) + \boldsymbol{h}(\boldsymbol{x})\right) \\
&= \boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i\left(\boldsymbol{v}_{i,\nu} + \boldsymbol{w}_{i,\nu}\right),
\end{aligned}$$

which implies that

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu+1}-\boldsymbol{x}\|^2\right] &= \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}-\alpha_i\boldsymbol{v}_{i,\nu}\|^2\right] + \alpha_i^2\mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right] \\
&\quad - 2\mathbb{E}\left[\mathbb{E}\left[\langle \boldsymbol{x}_{i,\nu}-\boldsymbol{x}-\alpha_i\boldsymbol{v}_{i,\nu}, \alpha_i\boldsymbol{w}_{i,\nu}\rangle \mid \mathcal{F}_{i,\nu-1}\right]\right] \\
&= \mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}-\boldsymbol{x}-\alpha_i\boldsymbol{v}_{i,\nu}\|^2\right] + \alpha_i^2\mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right].
\end{aligned} \tag{81}$$

Here, the last equality uses Lemma G.3 since $\mathbb{E}[\boldsymbol{w}_{i,\nu}|\mathcal{F}_{i,\nu-1}] = 0$.

From Assumption D and Lemma G.1, for $\boldsymbol{w}_{i,\nu}$ defined in (80), we have

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{w}_{i,\nu}\|^2\right] &\leq 3\mathbb{E}\left[\|\boldsymbol{h}_i(\boldsymbol{x}_{i,\nu}) - \nabla f_i(\boldsymbol{x}_{i,\nu})\|^2 + \|\nabla f_i(\boldsymbol{x}) - \boldsymbol{h}_i(\boldsymbol{x})\|^2 + \|\boldsymbol{h}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2\right] \\ &\leq 9\sigma_f^2.\end{aligned} \tag{82}$$

Substituting (82) into (81), we get

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x} - \alpha_i \boldsymbol{v}_{i,\nu}\|^2\right] &\leq \left(1 + \frac{1}{2\tau_i - 1}\right)\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 2\tau_i\alpha_i^2\mathbb{E}\left[\|\boldsymbol{v}_{i,\nu}\|^2\right] + 9\alpha_i^2\sigma_f^2 \\ &\leq \left(1 + \frac{1}{2\tau_i - 1} + 4\tau_i\alpha_i^2 L_f^2\right)\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 4\tau_i\alpha_i^2\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 9\alpha_i^2\sigma_f^2 \\ &\leq \left(1 + \frac{1}{\tau_i - 1}\right)\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] + 4\tau_i\alpha_i^2\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 9\alpha_i^2\sigma_f^2.\end{aligned} \tag{83}$$

Here, the first inequality follows from Lemma G.2; the second inequality uses Assumption C and Lemma G.1; and the last inequality follows by noting $\alpha_i = \alpha/\tau_i, \forall i \in \mathcal{S}$ and $\alpha \leq 1/(3L_f)$.

Now, iterating equation (83) and using $\boldsymbol{x}_{i,0} = \boldsymbol{x}, \forall i \in \mathcal{S}$, we obtain

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu} - \boldsymbol{x}\|^2\right] &\leq \left(4\tau_i\alpha_i^2\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 9\alpha_i^2\sigma_f^2\right)\sum_{j=0}^{\nu-1}\left(1 + \frac{1}{\tau_i - 1}\right)^j \\ &\leq 12\tau_i^2\alpha_i^2\mathbb{E}\left[\|\nabla f(\boldsymbol{x})\|^2\right] + 27\tau_i\alpha_i^2\sigma_f^2,\end{aligned} \tag{84}$$

where the second inequality uses (45). This completes the proof. □

F.1. Proof of Theorem 3.2

Proof. Let $\bar{\alpha}_1 := 1/(3L_f(1 + 8L_f))$. Note that by our assumption $\alpha_k \leq \bar{\alpha}_1$. Hence, the stepsize α_k satisfies the condition of Lemma F.2, and we have $6L_f^2\alpha_k^3 \leq \alpha_k^2/4$. This together with Lemmas F.1 and F.2 gives

$$\begin{aligned}\mathbb{E}\left[f(\boldsymbol{x}^{k+1})\right] - \mathbb{E}\left[f(\boldsymbol{x}^k)\right] &\leq -\frac{\alpha_k}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] + \frac{L_f^2\alpha_k}{2m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}^k - \boldsymbol{x}^k\|^2\right] \\ &\quad - \frac{\alpha_k}{2}(1 - \alpha_k L_f)\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\nabla f_i(\boldsymbol{x}_{i,\nu}^k)\right\|^2\right] + \frac{\alpha_k^2 L_f}{2}\sigma_f^2 \\ &\leq -\frac{\alpha_k}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] + \frac{L_f^2\alpha_k}{2m}\sum_{i=1}^m\frac{1}{\tau_i}\sum_{\nu=0}^{\tau_i-1}\mathbb{E}\left[\|\boldsymbol{x}_{i,\nu}^k - \boldsymbol{x}^k\|^2\right] + \frac{\alpha_k^2 L_f}{2}\sigma_f^2 \\ &\leq -\frac{\alpha_k}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] + 6L_f^2\alpha_k^3\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] + \left(\frac{27}{2}\alpha_k^3 L_f^2 + \frac{\alpha_k^2 L_f}{2}\right)\sigma_f^2 \\ &\leq -\frac{\alpha_k}{4}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] + (4 + L_f)\alpha_k^2\sigma_f^2,\end{aligned} \tag{85}$$

where the second and last inequalities follow from (14).

Summing (85) over k and using our choice of stepsize in (14), we obtain

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(\boldsymbol{x}^k)\|^2] &\leq \frac{4\Delta_f}{K}\cdot\min\left\{\frac{1}{\bar{\alpha}_1}, \frac{\sqrt{K}}{\bar{\alpha}}\right\} + 4(4 + L_f)\sigma_f^2\cdot\frac{\bar{\alpha}}{\sqrt{K}} \\ &\leq \frac{4\Delta_f}{\bar{\alpha}_1 K} + \left(\frac{4\Delta_f}{\bar{\alpha}} + 4(4 + L_f)\bar{\alpha}\sigma_f^2\right)\frac{1}{\sqrt{K}},\end{aligned} \tag{86}$$

where $\Delta_f = f(\boldsymbol{x}^0) - \mathbb{E}[f(\boldsymbol{x}^K)]$. □

G. Other Technical Lemmas

We collect additional technical lemmas in this section.

Lemma G.1. *For any set of vectors $\{\boldsymbol{x}_i\}_{i=1}^m$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, we have*

$$\left\|\sum_{i=1}^m \boldsymbol{x}_i\right\|^2 \leq m \sum_{i=1}^m \|\boldsymbol{x}_i\|^2. \tag{87}$$

Lemma G.2. *For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the following holds for any $c > 0$:*

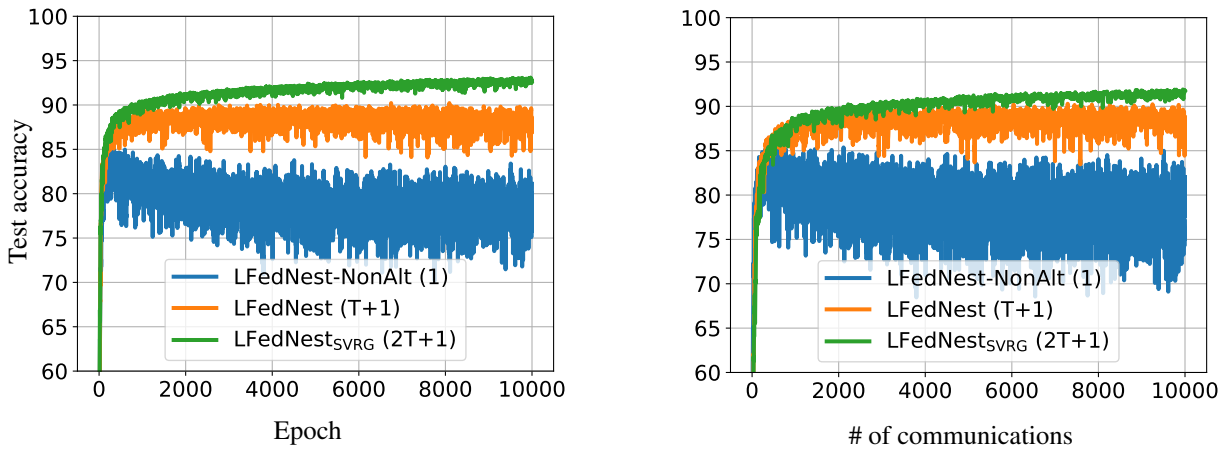
$$\|\boldsymbol{x}+\boldsymbol{y}\|^2 \leq (1+c)\|\boldsymbol{x}\|^2 + \left(1+\frac{1}{c}\right)\|\boldsymbol{y}\|^2. \tag{88}$$

Lemma G.3. *For any set of independent, mean zero random variables $\{\boldsymbol{x}_i\}_{i=1}^m$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, we have*

$$\mathbb{E}\left[\left\|\sum_{i=1}^m \boldsymbol{x}_i\right\|^2\right] = \sum_{i=1}^m \mathbb{E}\left[\|\boldsymbol{x}_i\|^2\right]. \tag{89}$$

H. Additional Experimental Results

In this section, we first provide the detailed parameters in Section 4 and then discuss more experiments. In Section 4, our federated algorithm implementation is based on (Ji, 2018), both hyper-representation and loss function tuning use batch size 64 and Neumann series parameter $N = 5$. We conduct 5 SGD/SVRG epoch of local updates in FEDINN and $\tau = 1$ in FEDOUT. In FEDNEST, we use $T = 1$, have 100 clients in total, and 10 clients are selected in each FEDNEST epoch.



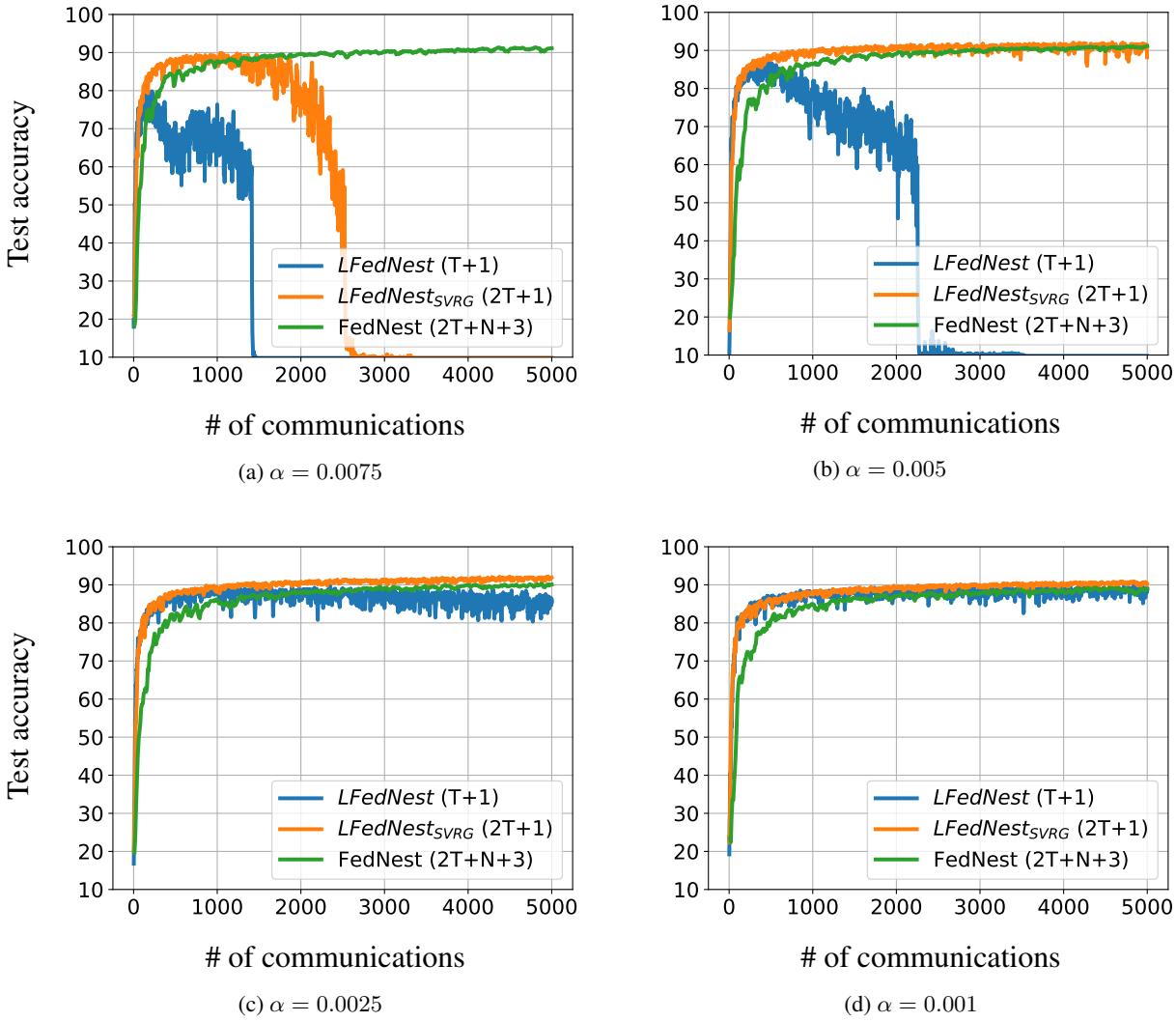
(a) The test accuracy w.r.t to the algorithm epochs.

(b) The test accuracy w.r.t to the number of communications.

Figure 5: Hyper representation experiment comparing LFEDNEST, LFEDNEST$_{SVRG}$ and LFEDNEST-NONALT on **non-i.i.d** dataset. The number in parentheses corresponds to communication rounds shown in Table 2.

H.1. The effect of the alternating between inner and outer global variables

In our addition experiments, we investigate the effect of the alternating between inner and outer global variables $\boldsymbol{x}$ and $\boldsymbol{y}$. We use LFEDNEST-NONALT to denote the training where each client updates their local $\boldsymbol{y}_i$ and then update local $\boldsymbol{x}_i$ w.r.t. local $\boldsymbol{y}_i$ for all $i \in \mathcal{S}$. Hence, the nested optimization is performed locally (within the clients) and the joint variable $[\boldsymbol{x}_i, \boldsymbol{y}_i]$ is communicated with the server. One can notice that only one communication is conducted when server update global $\boldsymbol{x}$ and $\boldsymbol{y}$ by aggregating all $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$.



(a) $\alpha = 0.0075$ (b) $\alpha = 0.005$

(c) $\alpha = 0.0025$ (d) $\alpha = 0.001$

Figure 6: Learning rate analysis on **non-i.i.d.** data with respect to **# of communications**.

As illustrated in Figure 5, the test accuracy of LFEDNEST-NONALT remains around 80%, but both standard LFEDNEST and LFEDNEST$_{\text{SVRG}}$ achieves better performance. Here, the number of inner iterations is set to $T = 1$. The performance boost reveals the necessity of both averaging and SVRG in FEDINN, where the extra communication makes clients more consistent.

H.2. The effect of the learning rate and the global inverse Hessian

Figure 6 shows that on non-i.i.d. dataset, both SVRG and FEDOUT have the effect of stabilizing the training. Here, we set $T = 1$ and $N = 5$. As we observe in (a)-(d), where the learning rate decreases, the algorithms with more communications are easier to achieve convergence. We note that LFEDNEST successfully converges in (d) with a very small learning rate. In contrast, in (a), FEDNEST (using the global inverse Hessian) achieves better test accuracy in the same communication round with a larger learning rate.