# BAMBI: VERTICAL FEDERATED BILEVEL OPTIMIZATION WITH PRIVACY-PRESERVING AND COMPUTATION EFFICIENCY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vertical federated learning (VFL) has shown promising in meeting the vast demands of multi-party privacy-preserving machine learning. However, existing VFL methods are not applicable to popular machine learning tasks falling under bilevel programming, such as hyper-representation learning and hyperparameter tuning. A desirable solution is adopting bilevel optimization (BO) into VFL, but on-shelf BO methods are shackled by the difficulty in computing the hypergradient with privacy-preserving and computation-efficient under the setting of VFL. To address this challenge, this paper proposes a stochastic Bilevel optimizAtion Method with a desirable JacoBian estImator (BAMBI), which constructs a novel zeroth-order (ZO) estimator to locally approximate the Jacobian matrix. This approximation enables BAMBI to compute the hypergradient in a privacy-preserving and computation-efficient manner. We prove that BAMBI convergences in the rate of $\mathcal{O}(1/\sqrt{K})$ ($K$ is the total number of the upper-level iterations) under the nonconvex-strongly-convex setting that covers most practical scenarios. This convergence rate is comparable with algorithms without a ZO estimator, which justifies the advantage in privacy preservation without sacrifice in convergence rate. Moreover, we design a BAMBI-DP for further mitigating the concerns on label privacy by leveraging the differential privacy (DP) technique. Extensive experiments fully support our algorithms. The code will be released publicly. To our best knowledge, this is the first work on the bilevel optimization under the setting of VFL.

## 1 INTRODUCTION

Vertical federated learning (VFL) attracts increasing attention due to the emerging concerns over data privacy in multi-party collaborative learning Hardy et al. (2017); Vepakomma et al. (2018); Liu et al. (2019b); Hu et al. (2019); Zhang et al. (2021b;c). Currently, extensive VFL methods have gained success in various applications, such as medical study, financial risk, and targeted marketing Cheng et al. (2019); Hu et al. (2019); Liu et al. (2019a;b); Li et al. (2021); Zhang et al. (2021c). However, these methods are designed only for machine learning (ML) problems with single-level structure and are not applicable to those falling under bilevel programming, such as hyper-representation learning and hyperparameter tuning, which are becoming popular in practical VFL applications. Thus, it is desirable to design methods solving ML problems with bilevel structures under the setting of VFL.

A desirable solution is adopting bilevel optimization (BO) Willoughby (1979) into VFL because extensive stochastic BO methods have been proposed to well address various machine learning tasks falling under bilevel programming Grazzi et al. (2020); Ji et al. (2021); Rajeswaran et al. (2019); Ji & Liang (2021); Tarzanagh et al.; Gao (2022). However, it is challenging to achieve this because it is difficult to compute the hypergradient of BO problems under the setting of VFL (defined as vertical federated bilevel optimization problems, VFBO problems) with privacy-preserving and computation-efficient. On-shelf BO methods Chen et al. (2021); Yang et al. (2021); Ji & Liang (2021); Grazzi et al. (2020) use the second-order derivatives to approximate Chen et al. (2021); Yang et al. (2021); Ji & Liang (2021) or directly compute Grazzi et al. (2020) the inverse Hessian matrix used for computing the hypergradient. However, besides the high computation complexity, applying existing methods to VFBO problems will cause feature privacy leakage. Specifically, 1) directly computing the second-order derivatives requires each party to access data of all features (not only

its own features but also features of other parties), which will lead to feature privacy leakage, 2) approximating or directly computing the inverse Hessian matrix has a high computation complexity. Thus, it is challenging to design VFBO methods with privacy-preserving and computation efficiency.

In this paper, we address these challenges by proposing the novel stochastic Bilevel optimizAtion Method with a desirable JacoBian estImator (BAMBI). Specifically, BAMBI ingeniously adopts the zeroth-order (ZO) estimation technique to approximate the Jacobian matrix, which enables all parties to collaboratively compute the hypergradient with privacy-preserving and computation efficiency. We theoretically prove that BAMBI still has the convergence rate of $1/\sqrt{K}$ for nonconvex-strongly-convex problems, where $K$ is the total number of upper-level iterations. This convergence rate matches those of BO algorithms not using ZO estimation, which justifies the advantage in privacy preservation without sacrificing convergence rate.

In BAMBI, only the intermediate gradients (rather than raw labels) are transmitted (please refer to Fig. 1 and its explanation for details) from the server to parties not holding labels (denoted as passive parties), which enables passive parties to optimize models locally without directly accessing the labels. Therefore, it seems BAMBI can preserve label privacy. However, existing works have shown that transmitting intermediate gradient is vulnerable to label privacy leakage Li et al. (2021); Sun et al. (2022); Yang et al. (2022). Particularly, raw labels often contain highly sensitive information (e.g., important demographic information Ghazi et al. (2021) or disease diagnosis results Vepakomma et al. (2018)). Thus, it is also important to preserve the label privacy of BAMBI.

To address this important problem, we design the BAMBI-DP by leveraging differential privacy (DP) technique to further preserve the label privacy. Specifically, BAMBI-DP adds well designed noises to the intermediate gradients, which is proved to guarantee $(\epsilon, 0)$-differentially private with respect to (w.r.t.) the label. We summarize the contributions of this paper as follows.

- To our best knowledge, we are the first to propose methods, i.e. BAMBI and BAMBI-DP, for solving VFL problems falling under bilevel programming.
- We design a desirable Jacobian estimator in BAMBI, which enables all parties to collaboratively compute the hypergradient with privacy-preserving and computation efficiency. We also derive the convergence rate of BAMBI for nonconvex-strongly-convex problems, which justifies our advantage in privacy preservation without sacrificing convergence rate.
- We further design the BAMBI-DP to preserve the label privacy, which is proved to be $(\epsilon, 0)$-differentially private w.r.t. the label.

**Notations** $\mathbf{a}^i \in \mathbb{R}^d$ denotes features of sample $i$, and $\mathbf{b}^i$ denotes its label. Given a positive integer $l$, $[l]$ denotes the set $\{1, \cdots, l\}$. We use subscript $m \in [l]$ to denote notations associated with party $m$, e.g., $\mathbf{a}_m^i \in \mathbb{R}^{d_m}$, $\mathbf{x}_m \in \mathbb{R}^{p_m}$ and $\mathbf{y}_m \in \mathbb{R}^{q_m}$ denote the features, upper- and lower-level variables on party $m$, respectively. We use superscript $t = 0, \cdots, N-1$, $k = 1, \cdots, K$ to denote the timestamp of variables, where $N$ and $K$ are the total number of low- and upper-level iterations.

## 2 Problem Formulation

Considering a VFL system with $l$ parties, where each party holds different features of the same sample, i.e., given a sample $\mathbf{a}^i \in \mathbb{R}^d$, it can be represented as $\mathbf{a}^i = [\mathbf{a}_1^i, \ldots, \mathbf{a}_l^i]$, where $\mathbf{a}_m^i \in \mathbb{R}^{d_m}$ and $\sum_{m=1}^{l} d_m = d$. We further assume that only one party (denoted as active party) holds labels and the rest (denoted as passive parties) do not. Moreover, this active party plays the role of the server.

In this paper, we consider the VFL problems falling under bilevel programming, where party $m$ only learns the $m$-th components of both the lower- and upper-level variables. Mathematically, such problems can be formulated as the VFBO problems with the following form.

$$\begin{aligned} &\min_{[\mathbf{x}_1, \cdots, \mathbf{x}_l] \in \mathbb{R}^p} F(\mathbf{x}) := \mathbb{E}_{\xi_{\mathbf{x}}}[f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}); \xi_{\mathbf{x}})] \quad \text{(upper-level)} \\ &\text{s.t.} \quad \mathbf{y}^*(\mathbf{x}) = \underset{[\mathbf{y}_1, \cdots, \mathbf{y}_l] \in \mathbb{R}^q}{\arg\min} \ \mathbb{E}_{\xi_{\mathbf{y}}}[g(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{y}})] \quad \text{(lower-level)}, \end{aligned} \tag{1}$$

where the upper- and lower-level objectives $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ and $g : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ are continuously differentiable w.r.t. the upper-level variable $\mathbf{x} \in \mathbb{R}^p$ and the lower-level variable $\mathbf{y} \in \mathbb{R}^q$, respectively. Random samples $\xi_{\mathbf{x}}$ and $\xi_{\mathbf{y}}$ are uniformly drawn from $\mathcal{D}_{up}$ and $\mathcal{D}_{low}$ that are training datasets used

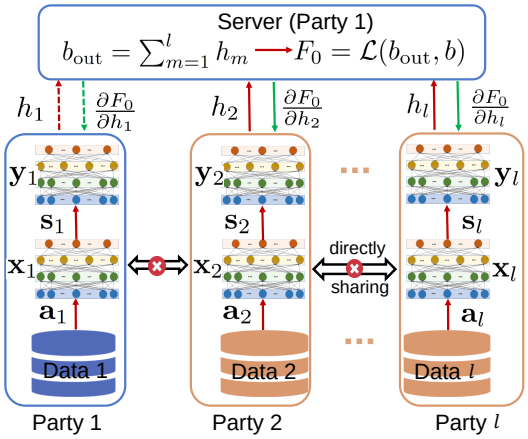



Figure 1: An illustration of VFBO, where the red line denotes forward process, the green line denotes backward process, $\mathbf{s}_m = T_{m,1}(\mathbf{x}_m, \mathbf{a}_m)$ and $h_m = T_m[(\mathbf{x}_m, \mathbf{y}_m), \mathbf{a}_m] = T_{m,2}(\mathbf{y}_m, \mathbf{s}_m)$ for $m \in [l]$, where $T_{m,1}, T_{m,2}, T_m$ denote the models parameterized by $\mathbf{x}_m$, $\mathbf{y}_m$, and $(\mathbf{x}_m, \mathbf{y}_m)$, respectively.

for the upper- and lower-level optimization, respectively. Note that, there are $\mathbf{x} \in \mathbb{R}^p = [\mathbf{x}_1, \cdots, \mathbf{x}_l]$ and $\mathbf{y} \in \mathbb{R}^q = [\mathbf{y}_1, \cdots, \mathbf{y}_l]$, which means that both the upper- and lower-level optimizations fall under the vertical federated learning. VFBO problems with form 1 capsule many bilevel optimization problems under VFL setting, such as hyper-representation learning and hyperparameter optimization in VFL (please refer to Problems 1 and 2 in Section 5 for details).

## 3 Proposed Algorithms

In this section, we propose BAMBI and BAMBI-DP for solving VFL problems with bilevel structures.

### 3.1 Novel Stochastic BO Method with A Desirable Jacobian Estimator

**Challenges of Computing the Hypergradient in VFBO:** For VFBO problems, the key step is how to compute the hypergradient (i.e., the gradient of the upper-level objective w.r.t. the upper-level variable $\mathbf{x}$), which takes the following form:

$$\nabla^* F(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \mathcal{J}^*(\mathbf{x})^\top \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})), \tag{2}$$

where the Jacobian matrix $\mathcal{J}^*(\mathbf{x}) = \frac{\partial \mathbf{y}^*(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{q \times p}$. As for $\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ and $\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ in Eq. 2, we can compute them easily because the $m$-components of them can be computed locally on each party $m$ for $\forall m \in [l]$. As for the Jacobian matrix $\mathcal{J}^*(\mathbf{x})$, existing BO methods Chen et al. (2021); Yang et al. (2021); Ji & Liang (2021) always first use the optimality condition of $\mathbf{y}^*(\mathbf{x})$ and the strongly-convexity of $g(\mathbf{y}, \mathbf{x})$ to obtain

$$\mathcal{J}^*(\mathbf{x}) = -\left[\nabla^2_{\mathbf{y}} g\left(\mathbf{x}, \mathbf{y}^*(\mathbf{x})\right)\right]^{-1} \nabla_{\mathbf{x}} \nabla_{\mathbf{y}} g\left(\mathbf{x}, \mathbf{y}^*(x)\right), \tag{3}$$

where $\nabla^2_{\mathbf{y}} g\left(\mathbf{x}, \mathbf{y}^*(\mathbf{x})\right)$ is assumed to be invertible. Then, these methods adopt the following strategies: 1) directly computing the inverse Hessian matrix Guo & Yang (2021), 2) approximating the inverse Hessian matrix by solving the linear system $\nabla^2_{\mathbf{y}} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))v = \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ Grazzi et al. (2020); Ji et al. (2020), 3) or approximating the inverse Hessian matrix through the Neumann series $\sum_{i=1}^{\infty} U^i = (I - U)^{-1}$ Chen et al. (2021); Ji et al. (2020). However, besides the high computation complexity, adopting existing strategies in VFBO will also cause feature privacy leakage. Specifically, 1) direct computation of the second-order derivatives requires each party in VFL to access data of all features not only its own features (we give the detailed explanation in the supplemental material), which leads to feature privacy leakage if no privacy-preserving techniques are adopted, 2) although Jacobian- or/and Hessian-vector implementations are adopted in these methods Grazzi et al. (2020); Chen et al. (2021); Ji et al. (2020), the computation complexity of approximating or directly computing the inverse Hessian matrix can still be very high for high-dimensional problems. Thus, it is challenging to compute the hypergradient of VFBO problems with privacy-preserving and computation efficiency.

As for the data privacy, cryptographic techniques, such as secure multi-party computation Micali et al. (1987) and homomorphic encryption Brakerski et al. (2014), can be adopted to preserve the raw data from sharing Gascón et al. (2016); Bonawitz et al. (2017); Hardy et al. (2017). However, they will cause significant computation and communication costs Liu et al. (2019a); Zhang et al. (2021c). In this paper, we propose to use the ZO estimation technique to elaborately construct a desirable Jacobian estimator, which enables all parties to collaboratively compute the hypergradient with privacy-preserving and computation efficiency.

**Algorithm 1** BAMBI on party $m$ (performed in synchronous parallel manner).

**Initialize:** Stepsizes $\{\alpha_k, \beta_k\}$, $\mathbf{x}_m^0 \in \mathbb{R}^{p_m}$, $\mathbf{y}_m^{0,0}$, and input $K, N, Q$.
**for** $k = 1, 2, ..., K$ **do**
  Set $\mathbf{y}_m^{k,0} = \mathbf{y}_m^{k-1,N}$ and $\hat{\mathbf{y}}_m^{k,0,j} = \mathbf{y}_m^{k-1,N}$ for $j \in [Q]$
  Generate $u_{k,j} \in \mathcal{N}(0,1)$ for $j \in [Q]$.
  **for** $t = 0, 1, ..., N-1$ **do**
    Uniformly draw a batch samples $\xi_{\mathbf{y}}$ form $\mathcal{D}_{\text{low}}$.
    Compute $h_m^{k,t}$ and $\{\hat{h}_m^{k,t,j}\}_{j=1}^Q$.
    **For passive parties:**
      Send $h_m^{k,t}$ and $\{\hat{h}_m^{k,t,j}\}_{j=1}^Q$ to the server.
      Receive $\theta_m^{k,t}$ and $\{\hat{\theta}_m^{k,t,j}\}_{j=1}^Q$ from the server.
    **For the server:**
      Compute $\theta_m^{k,t}$ and $\{\hat{\theta}_m^{k,t}\}_{j=1}^Q$, and sent it to all parties.
    Compute $\mathbf{v}_m^{k,t}$ and $\{\hat{\mathbf{v}}_m^{k,t,j}\}_{j=1}^Q$.
    Update $\mathbf{y}_m^{k,t+1} = \mathbf{y}_m^{k,t} - \beta_k \mathbf{v}_m^{k,t}$ and $\hat{\mathbf{y}}_m^{k,t+1,j} = \mathbf{y}_m^{k,t,j} - \beta_k \hat{\mathbf{v}}_m^{k,t,j}$ for $j \in [Q]$.
  **end for**
  Compute $\hat{\mathcal{J}}_m^N(\mathbf{x}^k; u_{k,j}) = \frac{\hat{\mathbf{y}}_m^{k,N,j} - \mathbf{y}_m^{k,N}(\mathbf{x}^k)}{\mu} u_{k,j}$ for $j \in [Q]$
  Uniformly draw a batch samples $\xi_{\mathbf{x}}$ form $\mathcal{D}_{\text{up}}$.
  Compute $H_m^k = T_m(\mathbf{x}_m^k, \mathbf{y}_m^{k,N}; \xi_{\mathbf{x}})$.
  **For passive parties:**
  Sent $H_m^k$ to the server.
    Receive $\vartheta_m^k = \frac{\partial F_0}{\partial H_m^k}$ from the server.
  **For the server:**
    Compute $\vartheta_m^k$ and sent it to party $m$ for $\forall m \in [l]$.
  Compute $\nabla_{\mathbf{x}_m} f(\mathbf{x}^k, \mathbf{y}^{k,N}; \xi_{\mathbf{x}})$ and $\nabla_{\mathbf{y}_m} f(\mathbf{x}^k, \mathbf{y}^{k,N}; \xi_{\mathbf{x}})$.
  Update $\mathbf{x}_m^{k+1} = \mathbf{x}_m^k - \alpha_k \widehat{\nabla}_m F(\mathbf{x}^k)$ with $\widehat{\nabla}_m F(\mathbf{x}^k)$ defined in Eq. 5.
**end for**

**Zeroth-Order Estimation Technique:** ZO estimation is a powerful technique to estimate the gradient without using training samples' feature data. Specifically, ZO estimation approximates the gradient of a black-box function $h : \mathbb{R}^d \to \mathbb{R}$ through the oracle based only on the function values Nesterov & Spokoiny (2017), i.e., $\widehat{\nabla} h(x; u) = \frac{h(x+\mu u) - h(x)}{\mu} u$, where $u \in \mathbb{R}^n$ is draw from a specific distribution, $\mu > 0$ is the smoothing parameter, and $\widehat{\nabla} h(x; u)$ is an unbiased estimator of the gradient of the smoothed function $\mathbb{E}_u [h(x + \mu u)]$, i.e., $\nabla h_u(x)$ Ghadimi & Lan (2013); Zhang et al. (2021a). ZO estimation has promising properties of preserving privacy and low computation complexity as analyzed in Zhang et al. (2021a), which motivates us to design the desirable Jacobian estimator.

**The Desirable Jacobian Estimator:** Since ZO estimation can approximate the gradient without using the feature data, a nature idea is using it to estimate $\mathcal{J}^*(\mathbf{x})$ and then compute the hypergradient based on the analytical structure (Eq. 2). Specifically, we construct the novel Jacobian estimator $\hat{\mathcal{J}}(\mathbf{x}) \in \mathbb{R}^{q \times p}$ for $\mathcal{J}(\mathbf{x})$ as

$$\hat{\mathcal{J}}(\mathbf{x}) = \frac{\mathbf{y}(\mathbf{x}^k + \mu u) - \mathbf{y}(\mathbf{x})}{\mu} u^\top, \tag{4}$$

where $u \in \mathbb{R}^p$ is a Gaussian vector with independent and identically distributed (i.i.d.) entries. Then, $\mathcal{J}(\mathbf{x}) = [\mathcal{J}_1(\mathbf{x}), \cdots, \mathcal{J}_l(\mathbf{x})]$ can be estimated collaboratively by all parties with privacy-preserving and computation efficiency, where $\mathcal{J}_m(\mathbf{x}) = \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}_m}$ for $\forall m \in [l]$.

**The Proposed BAMBI:** Motivated by the above analyses, we propose BAMBI to solve bilevel optimization problems under VFL setting with privacy-preserving and computation efficiency. A simple illustration of BAMBI is presented in Fig. 1, where party $m$ learns the corresponding components of the upper- and lower-level parameters (i.e., $\mathbf{x}_m$ and $\mathbf{y}_m$) locally. In BAMBI, each updating includes a forward process (red line) and a backward process (green line). During the forward process, feature data $\mathbf{a}_m$ are embedded into embedding $h_m$ and then $h_m$ is sent to the server. Finally, the server completes one forward process by computing the loss. During the backward process, the server sends

**Algorithm 2** GradPerturb Algorithm on the Server

1: **Input:** True label $b \in \{0, 1\}$, gradients $\theta_0^{k,t}$ and $\theta_1^{k,t}$
2: The server samples $u \sim \mathrm{Lap}(c)$ with probability distribution function $p(x|c) = \frac{1}{2c}e^{-|x|/c}$
3: **Output:** $\widetilde{\theta}^{k,t} = [\widetilde{\theta}_1^{k,t}, \cdots, \widetilde{\theta}_l^{k,t}] = \theta_b^{k,t} + u \cdot (\theta_b^{k,t} - \theta_{1-b}^{k,t})$

the intermediate (partial) gradients to the passive parties. In this case, all parties (include the passive parties) can compute its local gradients by the chain rule. Moreover, in VFBO, either $\mathbf{x}_m$ or $\mathbf{y}_m$ is fixed and only the another one is updated during each update.

The proposed BAMBI is summarized in Algorithm 1. Note that, this is a synchronous parallel algorithm, which means that only after all parties have finished the communication with the server the next step will then be performed. At steps 3-15, party $m$ runs an $N$-step stochastic gradient descent to approximate $\mathbf{y}_m^{k,*}(\mathbf{x}^k)$ and generate $\hat{\mathbf{y}}_m^{k,N}(\mathbf{x}^k + \mu u_{k,j})$, where $h_m^{k,t} = T_m(\mathbf{x}_m^k, \mathbf{y}_m^{k,t}; \xi_\mathbf{y})$ and $\hat{h}_m^{k,t,j} = T_m(\mathbf{x}_m^k + \mu u_{k,j}, \mathbf{y}_m^{k,t}; \xi_\mathbf{y})$, $\theta_m^{k,t} = \frac{\partial F_0}{\partial \hat{h}_m^{k,t}}$ and $\hat{\theta}_m^{k,t,j} = \frac{\partial F_0}{\partial h_m^{k,t,j}}$, $\mathbf{v}_m^{k,t} = \nabla_{\mathbf{y}_m} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_\mathbf{y})$ and $\hat{\mathbf{v}}_m^{k,t,j} = \nabla_{\mathbf{y}_m} g(\mathbf{x}^k + \mu u_{k,j}, \mathbf{y}^{k,t}; \xi_\mathbf{y})$ are computed through the chain rule, e.g., $\mathbf{v}_m^{k,t} = \theta_m^{k,t} \frac{\partial h_m^{k,t}}{\partial \mathbf{y}_m}$. At step 16, the Jacobian estimator $\hat{\mathcal{J}}^N(\mathbf{x}; u_{k,j})$ is computed as Eq. 4. Note that, to reduce the estimation variance, we use the sample average over the $Q$ estimators to construct the following hypergradient estimator of party $m$.

$$\widehat{\nabla}_m F(\mathbf{x}^k) = \nabla_{\mathbf{x}_m} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \frac{1}{Q} \sum_{j=1}^{Q} \hat{\mathcal{J}}_m^N(\mathbf{x}^k; u_{k,j}) \nabla_{\mathbf{y}_m} f(\mathbf{x}^k, \mathbf{y}^{k,N}), \tag{5}$$

where $\hat{\mathcal{J}}_m^N(\mathbf{x}^k; u_{k,j})$ is the ZO estimation of $\frac{\partial \mathbf{y}^N(\mathbf{x}; \xi_\mathbf{y})}{\partial \mathbf{x}_m}$. At step 24, two stochastic gradients are computed by using the chain rule and $\vartheta_m^k$. Then, BAMBI enables all parties to collaboratively optimize the VFBO problem by the approximate hypergradient (Eq. 5), which does not require each party to access other parties' feature data and has a low computation complexity.

### 3.2 BAMBI-DP Algorithm

**Label Differential Privacy:** Label DP is a notation proposed for label privacy guarantee Chaudhuri & Hsu (2011); Wang & Xu (2019); Ghazi et al. (2021); Malek et al. (2021), where the labels are considered sensitive and their privacy needs to be protected (e.g., important demographic information Ghazi et al. (2021) or disease diagnosis results Vepakomma et al. (2018)).

In the backward process, the intermediate gradients are transmitted from the server to the passive parties, which has the risk of leaking label privacy Ghazi et al. (2021); Malek et al. (2021); Li et al. (2021). In many real-world applications, the labels are very sensitive Ghazi et al. (2021); Malek et al. (2021); Li et al. (2021) and their privacy is necessary to protect. Thus, we propose the BAMBI-DP algorithm, which mitigates the concerns on the label privacy of BAMBI by leveraging the differential privacy technique. The core step of BAMBI-DP is adding well designed noise to the transmitted intermediate gradients. We adopt GradPerturb procedure Yang et al. (2022) to achieve this. GradPerturb generates $\widetilde{\theta}_m^{k,t}$ and $\widetilde{\vartheta}_m^{k,t}$, which are $\theta_m^{k,t}$ and $\vartheta_m^{k,t}$ perturbed by the well designed noises. GradPerturb procedure for binary classification task is summarized in 2 ( Please refer to the supplemental material for the multi-class version), where $\theta_b^{k,t} = [\theta_{1,b}^{k,t}, \cdots, \theta_{l,b}^{k,t}]$, $\theta_{m,b}^{k,t} = \frac{\partial F_0}{\partial h_m^{k,t}}$ is computed with label $b$ for $m \in [l]$ and $b \in \{0, 1\}$, $\mathrm{Lap}(c)$ is the Laplace distribution with parameter $c$. Then we can obtain BAMBI-DP by replacing $\theta_m^{k,t}$ and $\vartheta_m^{k,t}$ in Algorithm 1 with $\widetilde{\theta}_m^{k,t}$ and $\widetilde{\vartheta}_m^{k,t}$ generated by Algorithm 2. In subsection 4.3, we prove that BAMBI-DP is $(\epsilon, 0)$-differentially private w.r.t. the label for the $l$-party ($l \geq 2$) VFL system, when $c \geq 1/\epsilon$.

## 4 Theoretical Analysis

In this section, we provide the convergence analysis of BAMBI, label privacy analysis of BAMBI-DP, and computation complexity analysis of both BAMBI and BAMBI-DP. Here we only provide the results, please refer to the supplemental material for the detailed analyses and proofs.

### 4.1 CONVERGENCE ANALYSIS OF BAMBI

We first make the following necessary assumptions for convergence analysis.

**Assumption 1 (Lipschitz continuity)** *$f, \nabla f, \nabla g, \nabla^2 g$ are respectively $\ell_{f,0}, \ell_{f,1}, \ell_{g,1}, \ell_{g,2}$-Lipschitz continuous, and $\nabla f, \nabla g, \nabla^2 g$ are also block-coordinate Lipschitz continuous, i.e., for $\forall m \in [l]$, $\nabla_m f, \nabla_m g, \nabla_m^2 g$ are respectively $\ell_{f,1}, \ell_{g,1,m}, \ell_{g,2,m}$-Lipschitz continuous.*

**Assumption 2 (Convexity of $f$ and $g$)** *For any fixed $\mathbf{y}$, $f(\mathbf{x}, \mathbf{y})$ is nonconvex in $\mathbf{x}$, and for any fixed $\mathbf{x}$, $g(\mathbf{x}, \mathbf{y})$ is $\mu_g$-strongly convex in $\mathbf{y}$.*

**Assumption 3 (Bounded Variance)** *For $\forall m \in [l]$, the stochastic block-coordinate derivatives $\nabla_m f(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{x}})$, $\nabla_m g(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{y}})$, $\nabla_m^2 g(\mathbf{x}, \mathbf{y}, \xi_{\mathbf{y}})$ are unbiased estimators of $\nabla_m f(\mathbf{x}, \mathbf{y})$, $\nabla_m g(\mathbf{x}, \mathbf{y})$, $\nabla_m^2 g(\mathbf{x}, \mathbf{y})$, respectively; and their variances are bounded by $\sigma_{f,m}^2, \sigma_{g,1,m}^2, \sigma_{g,2,m}^2$, respectively, for $|\xi_{\mathbf{x}}| = |\xi_{\mathbf{y}}| = 1$.*

Assumptions related to block-coordinate ones are common in VFL works Hardy et al. (2017); Liu et al. (2019b); Hu et al. (2019); Zhang et al. (2021b;c) and others are common in bilevel optimization literature Ghadimi & Wang (2018); Hong et al. (2020); Chen et al. (2021); Sow et al. (2021). Assumptions 1 and 2 together ensure that the first- and second-order derivations of $f(\mathbf{x}, \mathbf{y}), g(\mathbf{x}, \mathbf{y})$ as well as the solution mapping $\mathbf{y}^*(\mathbf{x})$ are well-behaved. We further define $\sigma_f^2 = \sum_{m=1}^l \sigma_{f,m}^2, \sigma_{g,1}^2 = \sum_{m=1}^l \sigma_{g,1,m}^2, \sigma_{g,2}^2 = \sum_{m=1}^l \sigma_{g,2,m}^2$.

**Proof Sketch:** In the following, we present the proof sketch towards obtaining Theorem 1. Following Chen et al. (2021), we construct the Lyapunov function as $\mathbb{V}^k := F(\mathbf{x}^k) + \|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2$. Then the difference between two Lyapunov functions is

$$\mathbb{V}^{k+1} - \mathbb{V}^k = F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) + (\|\mathbf{y}^{k+1,N} - \mathbf{y}^*(\mathbf{x}^{k+1})\| - \|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|), \tag{6}$$

where the first term on the right hand side quantifies the descent of the upper-level objective, the second term on the right hand side denotes the descent of the lower-level errors. We then obtain two lemmas to bound these two terms.

**Lemma 1** *Under Assumptions 1 to 3, for the low-level optimization process in Algorithm 1, we have*

$$\mathbb{E}[\|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2] \leq (1 - \beta_k \mu_g)^N \mathbb{E}[\|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2] + \frac{\beta_k \sigma_{g,1}^2}{B\mu_g}, \tag{7}$$

*where $B$ is the batchsize, $N$ is the total number of the lower-level iterations. Given $\mathbf{y}^{k+1,0} = \mathbf{y}^{k,N}$ as set in Algorithm 1, there is*

$$\begin{aligned}\mathbb{E}[\|\mathbf{y}^{k+1,0} - \mathbf{y}^*(\mathbf{x}^{k+1})\|^2] \leq (1 + \gamma_k + \ell_{yx} C_f^2 \alpha_k^2/4)\mathbb{E}[\|\mathbf{y}^{k+1,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2] \\ + (\ell_{\mathbf{y}}^2 \alpha_k^2 + \alpha_k/4 + \alpha_k^2 \ell_{yx}/4)\mathbb{E}[\widehat{\nabla} F(\mathbf{x}^k)] + (\ell_{\mathbf{y}}^2 \alpha_k^2 + \ell_{yx}\alpha_k^2/4)\widetilde{\sigma}^2,\end{aligned} \tag{8}$$

*where $\ell_{\mathbf{y}} = \frac{\ell_{g,1}}{\mu_g}$, $\ell_{yx} = \frac{\ell_{g,2} + \ell_{g,2}\frac{\ell_{g,1}}{\mu_g}}{\mu_g} + \frac{\ell_{g,1}(\ell_{g,2} + \ell_{g,2}\frac{\ell_{g,1}}{\mu_g})}{\mu_g^2}$, $C_f^2 = (\ell_{f,0} + \frac{\ell_{g,1}}{\mu_g}\ell_{f,1})^2 + \delta_\mu^2 + \widetilde{\sigma}^2$, $\delta_\mu^2 = \frac{\mu^2 \ell_\mu^2 q(p+6)^2}{2Q}$, $\ell_\mu = (1 + \frac{\ell_{g,1}}{\mu_g})(\frac{\ell_{g,2}}{\mu_g} + \frac{\ell_{f,1}\ell_{g,1}}{\mu_g^2})$, $\widetilde{\sigma}^2 = 2\frac{\sigma_f^2}{B} + 2(\ell_{f,0}^2 + \frac{\sigma_f^2}{B})(12\delta_\mu^2 + \frac{4(2p+8)}{Q}\frac{\ell_{g,1}^2}{\mu_g^2}) + 4(\frac{\ell_{g,1}^2}{\mu_g^2} + \delta_\mu^2)\frac{\sigma_f^2}{B}$, and $Q$ is number of performing ZO estimation.*

**Lemma 2** *Under Assumptions 1 to 3, for successive upper-level iterations, we have*

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}^{k+1})] - \mathbb{E}[F(\mathbf{x}^k)] \leq -\frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k)\|^2 - (\frac{\alpha_k}{2} - \frac{\ell_{f,1}\alpha_k^2}{2})\|\widehat{\nabla} F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1}\alpha_k^2}{2}\tilde{\sigma}^2 \\ + \alpha_k(\ell_{f,0}^2\delta_\mu^2 + 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2)\|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2) + 2\ell_{f,0}^2\|\mathcal{J}^N(\mathbf{x}^k) - \mathcal{J}^*(\mathbf{x}^k)\|_F^2,\end{aligned} \tag{9}$$

where $\mathcal{J}^N(\mathbf{x}^k) = \frac{\partial \mathbf{y}^N(\mathbf{x}^k; \xi_{\mathbf{y}})}{\partial \mathbf{x}^k}$. Then, we can bound term $\|\mathcal{J}^N(\mathbf{x}^k) - \mathcal{J}^*(\mathbf{x}^k)\|_F^2$ by the definition of the Jacobian matrix and Lemma 1. Finally, applying Lemmas 1 and 2 to Eq. 6, and telescoping it from $k = 1$ to $k = K$, we have Theorem 1.

**Theorem 1** *Under Assumptions 1-3, we define*

$$\hat{\alpha}_1 = \frac{\mu_g^2}{2\ell_{f,1}\mu_g^2 + 4\ell_{g,1}^2 + \ell_{yx}\mu_g^2}, \quad \hat{\alpha}_2 = \ell_{yx}C_f^2\alpha_k^2 + 8\alpha_k(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + \frac{\ell_{g,1}^2}{\mu_g^2}),$$
$$\hat{\alpha}_3 = \frac{\mu_g}{8\ell_{f,0}^2\ell_{g,2}^2(\frac{1}{2\mu_g} + \frac{\ell_{g,1}+\mu_g}{2\mu_g^2} + \frac{\ell_{g,1}}{2\mu_g})(1 + \frac{\ell_{g,1}}{\mu_g})}, \tag{10}$$

*and carefully choose the following stepsizes*

$$\alpha_k = \min\{\hat{\alpha}_1, \hat{\alpha}_2, \hat{\alpha}_3, \frac{1}{\sqrt{K}}\}, \quad \beta_k = \min\{\frac{\ell_{yx}C_f^2\hat{\alpha}_1 + 8(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + \frac{\ell_{g,1}^2}{\mu_g^2})}{2N\mu_g}\alpha_k, \frac{1}{2\mu_g}\},$$

*when applying Algorithm 1 to solve nonconvex-strongly-convex VFBO problems, we have*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2$$
$$\leq \frac{\mathbb{V}^0}{K\min\{\hat{\alpha}_1, \hat{\alpha}_2\}} + \frac{\mathbb{V}^0}{\sqrt{K}} + c_\kappa^0\delta_\mu^2 + \frac{\alpha c_\kappa^1}{\sqrt{K}NB}\frac{\sigma_{g,1}^2}{Q} + \frac{\alpha c_\kappa^2}{\sqrt{K}}\widetilde{\sigma}^2 + \frac{c_\kappa^3 + c_\kappa^4(\widetilde{\sigma}^2 + \delta_\mu^2)}{NB}\frac{\sigma_{g,1}^2}{Q}, \tag{11}$$

*where $c_\kappa^0, c_\kappa^1, c_\kappa^2, c_\kappa^3$ only dependent on the constants imposed in Assumptions 1-3, $\mathbb{V}^0 = F(\mathbf{x}^0) + \|\mathbf{y}^0 - \mathbf{y}^*(\mathbf{x}^0)\|^2$, $K$ is the total number of the upper-level iterations.*

**Remark 1** *Given assumptions and parameters in Theorem 1, choosing $\mu = \frac{\sqrt{Q}}{\sqrt{qp^3K}}$ and $NB = \mathcal{O}(\frac{1}{\sqrt{K}})$, then Algorithm 1 can obtain the convergence rate of $\frac{1}{\sqrt{K}}$, whose order matches those of BO algorithms not using ZO estimation Chen et al. (2021); Ji et al. (2020).*

### 4.2 Computation Complexity Analysis of BAMBI and BAMBI-DP

The rough computation complexity (CC) of a low-level iteration is $\mathcal{O}(B(Q+1)(p+q))$ and, especially, the CC of ZO estimation is $\mathcal{O}(BQ(p+q))$. Thus, the total CC of performing an upper-level iteration in BAMBI is $\mathcal{O}(NBQ(p+q))$. The CC of computing the second-order derivative in existing BO methods Chen et al. (2021); Yang et al. (2021) is $\mathcal{O}(Bpq)$. Thus, comparing to existing methods using the second-order derivatives to approximate the Jacobian matrix, BAMBI reduces the CC from $\mathcal{O}(Bpq)$ to $\mathcal{O}(BQ(p+q))$. For practical choice of $Q$ (typical choice is $Q \leq 10$) and practical VFBO problems with large $p, q$, such reduction is significant. The analysis of BAMBI-DP is similar.

### 4.3 Label Privacy Analysis of BAMBI-DP

In the following, we analyze the label privacy of BAMBI-DP and show that BAMBI-DP is differentially private w.r.t. the label. Please refer to the supplemental material for the detailed proofs. First, we show that, noised gradients generated by Algorithm 2 are differentially private.

**Lemma 3** *Let $\epsilon > 0$, when Laplace distribution parameter satisfies $c \geq 1/\epsilon$, Algorithm 2 is $(\epsilon, 0)$-differentially private w.r.t. the label.*

Note that, BAMBI-DP adopts Algorithm 2 to generate the noised gradients, thus, privacy budget of BAMBI-DP depends on that of Algorithm 2. Formally, we give the following result.

**Theorem 2** *Let $\epsilon > 0$, if Algorithm 2 is $(\epsilon, 0)$-differentially private under Laplace distribution, then BAMBI-DP is also $(\epsilon, 0)$-differentially private w.r.t. the label.*

Theorem 2 shows that when gradients with well designed noise are transmitted in BAMBI, the label is guaranteed to be differentially private.

## 5 Experiments

In this section, we implement extensive experiments to support the claims of our proposed algorithms in solving vertical federated bilevel optimization problems. For more experiments details please refer to the supplemental material.

**Experiment Settings:** All experiments are performed on a machine with 4 sockets, and each socket has 12 cores. The MPI is used for communication. Similar to previous VFL works Zhang et al. (2021c;b), we vertically partition the data into $l = 4$ non-overlapped parts with nearly equal number of features. We introduce following two representative VFBO problems for valuation.

**Problem 1: Hyper-Representation Learning in VFL.** Hyper-representation learning (HRL) Franceschi et al. (2018); Grazzi et al. (2020) trains a classifier following a two-phased optimization process. The upper level solves for the optimal embedding model (i.e., representation) parameters $\mathbf{x}$, and lower-level solves the optimal linear classifier parameters $\mathbf{y}$. For vertical federated HRL (VFHRL) problems, each party $m$ learns the embedding function of its own feature $\mathbf{a}_m$, thus we have $T(\mathbf{x}, \mathbf{a}^i) = [T_1(\mathbf{x}_1, \mathbf{a}_1^i), \cdots, T_l(\mathbf{x}_l, \mathbf{a}_l^i)]$. Similarly, classifier is distributed over all parties, and $\mathbf{y}_m$ on party $m$ is an operation on its own embedding, thus there is $T(\mathbf{x}, \mathbf{a}^i)\mathbf{y} = \sum_{m=1}^{l} T_m(\mathbf{x}_m, \mathbf{a}_m^i)\mathbf{y}_m$. Mathematically, the VFHRL problems can be modeled as the following VFBO problem:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^q} F(\mathbf{x}) := \tfrac{1}{|\mathcal{D}_{\text{up}}|}\textstyle\sum_{(\mathbf{a}^i,\mathbf{b}^i)\in\mathcal{D}_{\text{up}}} \mathcal{L}\left(\sum_{m=1}^{l} T_m(\mathbf{x}_m, \mathbf{a}_m^i)\mathbf{y}_m^*(\mathbf{x}), \mathbf{b}^i\right) \\ &\text{s.t.} \quad \mathbf{y}^*(\mathbf{x}) = \underset{\mathbf{y}\in\mathbb{R}^{c_n\times p}}{\arg\min}\, g(\mathbf{y}, \mathbf{x}) := \tfrac{1}{|\mathcal{D}_{\text{low}}|} \sum_{(\mathbf{a}^i,\mathbf{b}^i)\in\mathcal{D}_{\text{low}}} (\mathcal{L}(\textstyle\sum_{m=1}^{l} T(\mathbf{x}_m, \mathbf{a}_m^i)\mathbf{y}_m, \mathbf{b}^i) + \gamma\|\mathbf{y}\|^2), \end{aligned}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function, $c_n$ is the number of categories. Each embedding function $T_m(\cdot, \mathbf{x}_m)$ can be a linear transformation, a multi layer perceptron, or even a deep convolution neural network.

**Problem 2: Hyperparameter Optimization in VFL.** Hyperparameter optimization (HO) aims to find the set of the best hyperparameters (e.g., regularization coefficients) that yields the optimal value of some criterion of model quality (e.g., a validation loss on unseen data). HO can be posed as a bilevel optimization problem in which the lower problem corresponds to finding the model parameters by minimizing a training loss (usually regularized) for the given regularization coefficients and then the upper problem minimizes over the regularization coefficients. Note that, the regularization coefficients are different for different parties $m$. Mathematically, vertical federated HO (VFHO) problems can be formulated as follows.

$$\begin{aligned} &\min_{\mathbf{x}=[\mathbf{x}_1,\cdots,\mathbf{x}_l]} F(\mathbf{x}) := \tfrac{1}{|\mathcal{D}_{\text{up}}|}\textstyle\sum_{\xi_{\mathbf{x}}\in\mathcal{D}_{\text{up}}} \mathcal{L}\left(\mathbf{y}^*(\mathbf{x}); \xi_{\mathbf{x}}\right) \\ &\text{s.t.} \quad \mathbf{y}^*(\mathbf{x}) = \underset{\mathbf{y}=[\mathbf{y}_1,\cdots,\mathbf{y}_l]}{\arg\min}\, \mathcal{L}_{\text{low}}(\mathbf{y}, \mathbf{x}) := \tfrac{1}{|\mathcal{D}_{\text{low}}|}\textstyle\sum_{\xi_{\mathbf{y}}\in\mathcal{D}_{\text{low}}} (\mathcal{L}(\mathbf{y}, \mathbf{x}; \xi_{\mathbf{y}}) + \sum_{m=1}^{l} \mathbf{x}_m\mathcal{R}(\mathbf{y}_m)), \end{aligned}$$

where $\mathcal{L}$ is a loss function (e.g., logistic loss), $\mathcal{R}(\mathbf{y}_m)$ is a regularizer on $\mathbf{y}_m$, and $\mathbf{x}_m$ is the regularization coefficient.

**Datasets:** In the experiment, following existing works Ji et al. (2020); Sow et al. (2021) we use three datasets, including MNIST and FashionMnist datasets for VFHR problem, and News20 dataset for VFHO problem. For detail descriptions of these datasets please refer to the supplemental material. For all datasets, we use the test data or randomly choose $20\%$ of the total data as the validation data.

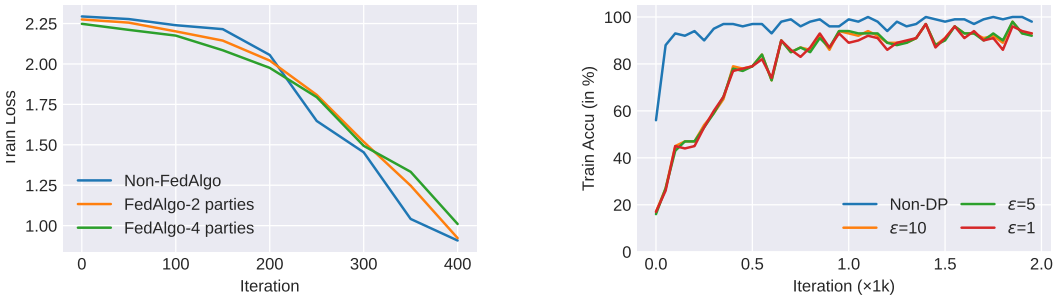
In the following, we only provide the results of VFHRL problem on MNIST, for more experimental results please refer to the supplemental material.

### 5.1 Evaluation of Comparable Performance

As for VFL, a general lossless constraint is that its performance should be comparable to the performance of model learned under the non-federated learning Zhang et al. (2021c;a). To demonstrate that BAMBI also satisfies this lossless constraint, we implement experiments and show that BAMBI has a comparable performance to its non-federated counterpart (Non-FedAlgo). Specifically, in Non-FedAlgo, all data are gathered in a union for training. The corresponding experimental results are shown in Fig. 2(a). According to the results, the convergence performance of BAMBI (FedAlgo) is comparable to that of Non-FedAlgo, which supports our claim.

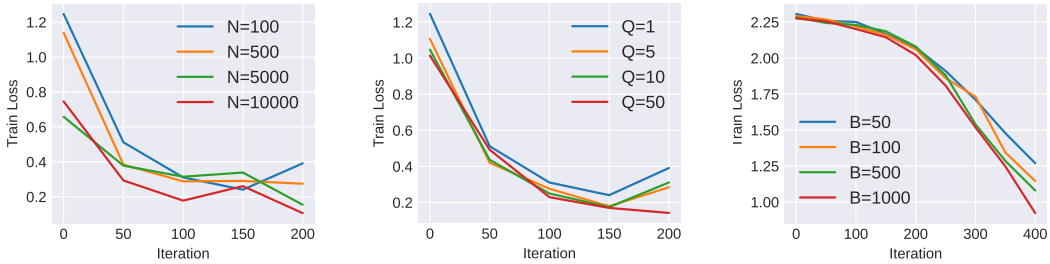
### 5.2 Evaluation of Different Parameters

In BAMBI-DP, parameter $\epsilon$ characters the level of noise and the level of label DP. A smaller $\epsilon$ means a better DP but a potential larger noise level. To study the influence of $\epsilon$, we implement BAMBI-DP with $\epsilon = 1, 5, 10$, and report their train accuracy v.s. iteration number curves in Fig. 2(b) and test accuracy in Table 1. Compare the results of BAMBI-DP with those of Non-DP (i.e., BAMBI), we




(a) Performance comparison (b) Performance comparison

Figure 2: (a) Performance comparison between FedAlgo and Non-FedAlgo on MNIST dataset, (b) Convergence performance with different levels of DP guarantee.




(a) Evaluation of different $N$ (b) Evaluation of different $Q$ (c) Evaluation of different $B$

Figure 3: Evaluation of using different parameters on MNIST dataset.

Table 1: Comparisons between BAMBI and BAMBI-DP with different label DP levels.

| | Non-DP | $\epsilon = 10$ | $\epsilon = 5$ | $\epsilon = 1$ |
|---|---|---|---|---|
| Test Accu | 97.59 | 94.35 | 94.23 | 93.60 |

have that a better DP guarantee (i.e., a smaller $\epsilon$) leads to a poorer model performance. Generally, we can conclude that there exists a trade-off between model performance and DP level, which is consistent with the results in Yang et al. (2022) and the no free lunch theorem for security and utility in federated learning Zhang et al. (2022).

In the proposed algorithm, the total number of inner epochs $N$ influences the quality of the approximate $\mathbf{y}^*(\mathbf{x})$, number of ZO estimation $Q$ influences the error introduced by ZO estimation, batch size $B$ influences the variance introduced by the stochastic samples. Thus, we implement experiments on MNIST dataset to study the influence on convergence rate of different $N$, $Q$ and $B$. In the experiments, we preset $N = 1000, Q = 1, B = 1000$ and vary one of them. The corresponding results in Fig. 3 show that although large $N, Q, B$ is helpful for fast convergence, choosing small $N, Q, B$ (e.g. $N = 500, Q = 1, B = 50$) is sufficient to obtain a good convergence result. This is beneficial for practical use.

## 6 Conclusion and Discussion

In this paper, we proposed two novel algorithms, i.e. BAMBI and BAMBI-DP, for solving vertical federated bilevel optimization problems. BAMBI adopts ZO estimation to estimate the Jacobian matrix, which enables all parties to collaboratively compute the hypergradient with privacy-preserving and computation efficiency. We theoretically proved the convergence rate of BAMBI for nonconvex-strongly-convex problems, whose order matches those of BO algorithms not using ZO estimation to preserve privacy. To preserve the label privacy, we further proposed BAMBI-DP by leveraging DP technique. We proved that BAMBI-DP is $(\epsilon, 0)$-differentially private w.r.t. the label. To our best knowledge, this is the first work focusing on the VFBO problems.

**Ethics Statement:**

We discuss and present the following potential limitations of this work. 1) We only consider the stochastic VFBO methods but not also the deterministic one, though it can be obtained easily based on our work. 2) As for BAMBI, we only study the synchronous parallel case, and do not consider the asynchronous parallel case.

Organizations (e.g., banks and hospitals) have the demands of training VFBO models with privacy-preserving (both feature and label privacy) will benefit from our proposed methods, while those earn illegal profits by inferring the private feature and label data of other parties will put at disadvantage from this work.

**Reproducibility Statement :**

As for reproducing the experiments, one can refer to Section E in the Appendix to obtain more details and refer to Zhang et al. (2021a); Yang et al. (2021) for how to implement the zeroth-order technique and bilevel optimization in VFL. For the detailed theoretical analysis, one can refer to Section D in the Appendix.

## REFERENCES

Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*, 2017.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 155–186. JMLR Workshop and Conference Proceedings, 2011.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. *arXiv preprint arXiv:2106.13781*, 2021.

Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. Secureboost: A lossless federated learning framework. *arXiv preprint arXiv:1901.08755*, 2019.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

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

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning (ICML)*, pp. 1568–1577, 2018.

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

Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans. Secure linear regression on vertically partitioned datasets. *IACR Cryptology ePrint Archive*, 2016:892, 2016.

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

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

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in Neural Information Processing Systems*, 34, 2021.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. Optimizing millions of hyperparameters by implicit differentiation. *International Conference on Machine Learning (ICML))*, 2020.

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

Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

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

Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. Fdml: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD*, pp. 2232–2240, 2019.

Kaiyi Ji and Yingbin Liang. Lower bounds and accelerated algorithms for bilevel optimization. *arXiv preprint arXiv:2102.03926*, 2021.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Nonasymptotic analysis and faster algorithms. *arXiv preprint arXiv:2010.07962*, 2020.

Kayi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. *International Conference on Machine Learning (ICML))*, 2021.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pp. 0. Lille, 2015.

Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. *arXiv preprint arXiv:2102.08504*, 2021.

Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient vertical federated learning framework. *arXiv preprint arXiv:1912.11187*, 2019a.

Yang Liu, Yingting Liu, Zhijie Liu, Junbo Zhang, Chuishi Meng, and Yu Zheng. Federated forest. *arXiv preprint arXiv:1905.10053*, 2019b.

Jelena Luketina, Mathias Berglund, Klaus Greff, and Tapani Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*, pp. 2952–2960. PMLR, 2016.

Mani Malek, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramèr. Antipodes of label differential privacy: Pate and alibi. *arXiv preprint arXiv:2106.03408*, 2021.

Silvio Micali, Oded Goldreich, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*, pp. 218–229. ACM, 1987.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, pp. 527–566, 2017.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 113–124, 2019.

Daouda Sow, Kaiyi Ji, and Yingbin Liang. Es-based jacobian enables faster bilevel optimization. *arXiv preprint arXiv:2110.07004*, 2021.

Jiankai Sun, Xin Yang, Yuanshun Yao, and Chong Wang. Label leakage and protection from forward embedding in vertical federated learning. *arXiv preprint arXiv:2203.01451*, 2022.

Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. Fednest: Federated bilevel optimization.

Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. In *International Conference on Machine Learning*, pp. 6628–6637. PMLR, 2019.

Ralph A Willoughby. Solutions of ill-posed problems (an tikhonov and vy arsenin). *SIAM Review*, 21 (2):266, 1979.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *arXiv preprint arXiv:2106.04692*, 2021.

Xin Yang, Jiankai Sun, Yuanshun Yao, Junyuan Xie, and Chong Wang. Differentially private label protection in split learning. *arXiv preprint arXiv:2203.02073*, 2022.

Qingsong Zhang, Bin Gu, Zhiyuan Dang, Cheng Deng, and Heng Huang. Desirable companion for vertical federated learning: New zeroth-order gradient based algorithm. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2598–2607, 2021a.

Qingsong Zhang, Bin Gu, Cheng Deng, Songxiang Gu, Liefeng Bo, Jian Pei, and Heng Huang. Asysqn: Faster vertical federated learning algorithms with better computation resource utilization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3917–3927, 2021b.

Qingsong Zhang, Bin Gu, Cheng Deng, and Heng Huang. Secure bilevel asynchronous vertical federated learning with backward updating. *arXiv preprint arXiv:2103.00958*, 2021c.

X. Zhang, H. Gu, L. Fan, K. Chen, and Q. Yang. No free lunch theorem for security and utility in federated learning. *arXiv preprint arXiv:2203.05816*, 2022.

# A Appendix

In the Appendix we provide following content.

- **Section B:** Explain why computation of the second-order derivatives requires each party in VFBO to access data of all features not only its local features.
- **Section C:** Provide multi-class version of BAMBI-DP.
- **Section D:** Provide complete proofs of theoretical results in the manuscript.
- **Section E:** Provide details of the experiments, and additional experimental results on Fashion-Mnist and News20 datasets.

## B COMPUTATION OF THE SECOND-ORDER DERIVATIVES IN VFBO

In this section, we explain why computation of the second-order derivatives requires each party in VFL to access data of all features not only its local features. In the following, we use Fig. 1 in the manuscript for illustration of the VFHRL problems. For other VFBO problems, the analyses are similar.

Feed a sample $\mathbf{a} \in \mathbb{R}^d$ with label $b$ to the model, we have $F_0 = \mathcal{L}(\sum_{m=1}^{\ell} h_m, b)$, where $h_m = T_{m,2}(T_{m,1}(\mathbf{x}_m, \mathbf{a}_m), \mathbf{y}_m)$, $T_{m,1}$ is the local model parameterized by $\mathbf{x}_m$, and $T_{m,2}$ is the local model parameterized by $\mathbf{y}_m$. Let $\mathbf{x}_{m,1}$ be the 1-th layer parameter of local model $T_{m,1}$ (i.e., model $T_{m,1,1}$) and $\mathbf{x}_{m,r}$ be the parameter of the rest model i.e., $T_{m,1,r}$, and $\mathbf{y}_{m,1}$ be the 1-th layer parameter of local model $T_{m,2}$ (i.e., mode $T_{m,2,1}$) and $\mathbf{y}_{m,r}$ be the parameter of the rest model, i.e., $T_{m,2,r}$. For analysis simplification, we further assume that the first layers of local models $T_{m,1}$ and $T_{m,2}$ are both linear mappings, which is reasonable in the practice and consist with our experimental setting. Then we have that $s_m = T_{m,1}(\mathbf{x}_m, \mathbf{a}_m) = T_{m,1,r}(s_{m,1}, \mathbf{x}_{m,r})$ and $h_m = T_{m,2}(\mathbf{y}_m, s_m) = T_{m,2,r}(h_{m,1}, \mathbf{y}_{m,r})$, where $s_{m,1} = \mathbf{x}_{m,1}\mathbf{a}_m$ and $h_{m,1} = \mathbf{y}_{m,1}s_m$. Let $\nabla_{\mathbf{y}}$ be derivative of the upper-level object w.r.t. $\mathbf{y}$, then, fix $\mathbf{x}$, for $m \in [l]$, we have that

$$\begin{aligned}\nabla_{\mathbf{y},m} &= \frac{\partial \mathcal{L}(h,b)}{\partial h}\frac{\partial h}{\partial h_m}\frac{\partial h_m}{\partial h_{m,1}}\frac{\partial h_{m,1}}{\partial \mathbf{y}_{m,1}} = \frac{\partial \mathcal{L}(h,b)}{\partial h}\frac{\partial h}{\partial h_m}\frac{\partial h_m}{\partial h_{m,1}} \cdot s_m \\ &= \underbrace{\frac{\partial \mathcal{L}(h,b)}{\partial h}}_{S_1}\underbrace{\left(\frac{\partial h}{\partial h_m}\frac{\partial h_m}{\partial h_{m,1}} \cdot T_{m,1,r}(\mathbf{x}_{m,1}\mathbf{a}_m, \mathbf{x}_{m,r})\right)}_{S_2}. \end{aligned} \tag{12}$$

According to the above equation, in VFBO, each party $m$ can compute $\nabla_{\mathbf{y},m}$ locally. However, computing the second-order derivative $\frac{\partial \nabla_{\mathbf{y}}}{\partial \mathbf{y}}$, i.e. $\nabla^2_{\mathbf{y}}$, is difficult. When computing $\nabla_{\mathbf{y}_m,\mathbf{y}'_m}$ for $m' \neq m$, we only need consider $\mathbf{y}_{m'}$ hided in $S_1$ (because $h = \sum_{m=1}^{l} h_m$) and $S_2$ is independent to $\mathbf{y}_{m'}$. Thus, we have

$$\nabla_{\mathbf{y}_m,\mathbf{y}'_m} = \frac{\partial S_1}{\partial \mathbf{y}_{m'}} \cdot S_2 = \frac{\partial F_0}{\partial h}\underbrace{\left(\frac{\partial h}{\partial h_{m'}}\frac{\partial h_{m'}}{\partial h_{m',1}} \cdot T_{m',1,r}(\mathbf{x}_{m',1}\mathbf{a}_{m'}, \mathbf{x}_{m',r})\right)}_{S_3} \cdot S_2. \tag{13}$$

According to the above analysis, it is obvious that computation of $S_3$ in Eq .13 requires party $m$ to access features owned by party $m'$. Thus, computation of $\nabla_{\mathbf{y}_m,\mathbf{y}}$ requires party $m$ to access data of all features not only its local feature. **This completes the explanation**.

Obviously, in VFBO, direct computation of $\nabla_{\mathbf{y}_m,\mathbf{y}'_m}$ ($m' \neq m$) is impossible unless party $m'$ share its feature data with party $m$. However, this will lead to the feature privacy leakage. Following the above analysis, it is easy to have that $\nabla_{\mathbf{y}_m,\mathbf{y}_m}$ can be computed by party $m$ locally. Thus, as for

$$\nabla^2_{\mathbf{y}} = \begin{pmatrix} \nabla_{\mathbf{y}_1,\mathbf{y}_1} & \cdots & \nabla_{\mathbf{y}_1,\mathbf{y}_l} \\ \vdots & \ddots & \vdots \\ \nabla_{\mathbf{y}_l,\mathbf{y}_1} & \cdots & \nabla_{\mathbf{y}_l,\mathbf{y}_l} \end{pmatrix}, \tag{14}$$

only the the block matrices located on the diagonal, i.e. $\nabla^2_{\mathbf{y}_m}$ for $m \in [l]$, can be computed without leaking the feature privacy. The drawbacks of using privacy-preserving techniques has discussed in the manuscript. The analysis of $\nabla^2_{\mathbf{x}}$ is similar.

## C BAMBI-DP FOR MULTI-CLASS CLASSIFICATION PROBLEMS

In this section, we provide the core procedure of BAMBI-DP for multi-class classification problems, i.e., GradPerturb for multi-class classification problems. We summarize it in Algorithm 3, where $C$ is the number of classes, and $\theta_b^{k,t}$ is intermediate gradient calculated with label $b$. As for BAMBI-DP for multi-class classification problems, one just need replace the intermediate gradients with the perturbed ones generated by Algorithm 3. For Algorithm 3 and BAMBI-DP for multi-class classification problems, we have the same DP guarantees, i.e., both of them are $(\epsilon, 0)$-differentially private, which can be proved by following Appendix D.2.

**Algorithm 3** GradPerturb for Multi-Class Classification Problems

1: **Input:** True label $b \in \{0, \cdots, C-1\}$, gradients $\theta_0^{k,t}, \theta_1^{k,t}, \cdots, \theta_{C-1}^{k,t}$
2: The server samples $u \sim \text{Lap}(c)$ with probability distribution function $p(x|c) = \frac{1}{2c} e^{-|x|/c}$
3: **Output:** $\widetilde{\theta}^{k,t} = [\widetilde{\theta}_1^{k,t}, \cdots, \widetilde{\theta}_l^{k,t}] = \theta_b^{k,t} + u \cdot \sum_{b=0}^{C-1} \theta_b^{k,t}$

# D Complete Proofs of Section 4 in the Manuscript

in this section, we provide the complete proofs of Theorem 1 and label differential privacy.

## D.1 Convergence Analysis of BAMBI

We have following properties of $g_u(\mathbf{x}, \mathbf{y})$, which are proved in Ghadimi & Lan (2013).

**Lemma 4 (Ghadimi & Lan (2013))** *Consider a function $g : \mathbb{R}^n \to R$, with each point $\mathbf{y} \in E$ is differentiable along any direction with $\ell$-Lipsctitz gradient. Then its Gaussian approximation is $g_u(\mathbf{x}, \mathbf{y}) = \mathbb{E}_u g(\mathbf{x} + \mu u, \mathbf{y})$, where $\mu > 0$, $u$ is a standard Gaussian random vector. Then, $g_u$ is differential and have following properties: 1) The gradient pf $g_u$ has the following form*

$$\nabla_{\mathbf{x}} g_u = \mathbb{E}_u \frac{g(\mathbf{x} + \mu u, \mathbf{y}) - g(\mathbf{x}, \mathbf{y})}{\mu} u \tag{15}$$

*2) For any $\mathbf{y} \in \mathbb{R}^n$, there is*

$$|g_u(\mathbf{x}, \mathbf{y}) - g(\mathbf{x}, \mathbf{y})| \leq \frac{\ell_u \mu^2 q}{2} \tag{16}$$

$$\|\nabla_{\mathbf{x}} g_u(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{y})\|_2^2 \leq \frac{\mu^2 \ell_u^2 (q+3)^3}{4}, \tag{17}$$

$$\mathbb{E}_u \| \frac{g(\mathbf{x} + \mu u, \mathbf{y}) - g(\mathbf{x}, \mathbf{y})}{\mu} u \| \leq 4(n+4) \|\nabla_{\mathbf{x}} g_u(\mathbf{x}, \mathbf{y})\|^2 + \frac{3}{2} \mu^2 \ell_\mu^2 (n+5)^3 \tag{18}$$

*where $\ell_u = \ell$ in this paper.*

In this paper, we define

$$\hat{\mathcal{J}}^N(\mathbf{x}^k, u_{k,j}) = \begin{pmatrix} \frac{\mathbf{y}_1^{k,N}(\mathbf{x}^k + \mu u_{k,j,1}; \xi_{\mathbf{y}}) - \mathbf{y}_1^{k,N}(\mathbf{x}^k)}{\mu} u_{k,j,1}^\top \\ \vdots \\ \frac{\mathbf{y}_l^{k,N}(\mathbf{x}^k + \mu u_{k,j,l}; \xi_{\mathbf{y}}) - \mathbf{y}_l^{k,N}(\mathbf{x}^k)}{\mu} u_{k,k,l}^\top \end{pmatrix}$$

where $u_{k,j,m} \in \mathbb{R}_m^q, j = 1, \ldots, Q, m \in [l]$ are standard Gaussian vectors and $\mathbf{y}^N(\mathbf{x}^k)$ is the output of SGD obtained with the minibatches $\{\xi_{\mathbf{y}}^0, ..., \xi_{\mathbf{y}}^{N-1}\}$, we have that conditioning on $\mathbf{x}^k$ and $\mathbf{y}^N(\mathbf{x}^k)$ and taking expectation over $u_j$ yields

$$\begin{aligned} \mathbb{E}_u \hat{\mathcal{J}}^N(\mathbf{x}^k, u_{k,j}) &= \mathbb{E}_u \begin{pmatrix} \frac{\mathbf{y}_1^{k,N}(\mathbf{x}^k + \mu u_{k,j,1}; \xi_{\mathbf{y}}) - \mathbf{y}_1^{k,N}(\mathbf{x}^k; \xi_{\mathbf{y}})}{\mu} u_{k,j,1}^\top \\ \vdots \\ \frac{\mathbf{y}_l^{k,N}(\mathbf{x}^k + \mu u_{k,j,l}; \xi_{\mathbf{y}}) - \mathbf{y}_l^{k,N}(\mathbf{x}^k; \xi_{\mathbf{y}})}{\mu} u_{k,j,l}^\top \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{J}_{u,1}^N(\mathbf{x}^k; \xi_{\mathbf{y}}) \\ \vdots \\ \mathcal{J}_{u,q}^N(\mathbf{x}^k; \xi_{\mathbf{y}}) \end{pmatrix} \\ &= \mathcal{J}_\mu^N(\mathbf{x}^k, \xi_{\mathbf{y}}) \end{aligned}$$

where $\mathbf{y}_m^{k,N}(\mathbf{x}^k; \xi_{\mathbf{y}})$ is the $m$-th component of vector $\mathbf{y}^{k,N}(\mathbf{x}; \xi_{\mathbf{y}})$ that is the entry-wise Gaussian smooth approximation of vector $\mathbf{y}^{k,N}(\mathbf{x}^k; \xi_{\mathbf{y}})$.

We further define some notations necessary for analysis. First, we define the true hypergradient at point $\mathbf{y}^*(\mathbf{x}^k)$

$$\nabla^* F(\mathbf{x}^k) = \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k)) + \mathcal{J}^*(\mathbf{x}^k)^\top \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))$$

where $\mathcal{J}^*(\mathbf{x}^k) = \frac{\partial \mathbf{y}^*(\mathbf{x}^k)}{\partial \mathbf{x}^k}$, and $\mathbf{y}^*(\mathbf{x}^k)$ denotes the optimal solution of the inner loop given $\mathbf{x}^k$.

$$\nabla F(\mathbf{x}^k) = \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \mathcal{J}^N(\mathbf{x}^k) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})$$

where $\mathcal{J}^N(\mathbf{x}^k) = \frac{\partial \mathbf{y}^{k,N}(\mathbf{x}^k)}{\partial \mathbf{x}^k}$.

$$\begin{aligned}\widehat{\nabla} F(\mathbf{x}^k) &= \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \hat{\mathcal{J}}^N(\mathbf{x}^k, u_k) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^N(\mathbf{x}^k)) \\ &= \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \frac{1}{Q} \sum_{j=1}^{Q} \hat{\mathcal{J}}^N(\mathbf{x}^k, u_{k,j}) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^N(\mathbf{x}^k))\end{aligned}$$

and where $\hat{\mathcal{J}}^N(\mathbf{x}^k, u_{k,j})$ is the approximate $\mathcal{J}^N(\mathbf{x}^k)$ with Gaussian smoothing.

$$\bar{\nabla} F(\mathbf{x}^k) = \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}; \xi_{\mathbf{x}}) + \hat{\mathcal{J}}^N(\mathbf{x}^k, u_k, \xi_{\mathbf{y}}) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^N; \xi_{\mathbf{x}})$$

where $\hat{\mathcal{J}}^N(\mathbf{x}^k, u_k; \xi_{\mathbf{y}}) = \frac{1}{Q} \sum_{j=1}^{Q} \frac{\mathbf{y}^N(\mathbf{x}^k + \mu u_{k,j}; \xi_{\mathbf{y}}) - \mathbf{y}^N(\mathbf{x}^k; \xi_{\mathbf{y}})}{\mu} u_{k,j}^\top$. Note that $\mathbb{E}_\xi[\hat{\mathcal{J}}^N(\mathbf{x}^k, u_k; \xi_{\mathbf{y}})] = \hat{\mathcal{J}}^N(\mathbf{x}^k, u_k)$

**Lemma 5 (Repeat of Lemma 1-1 in the Manuscript)** *At iteration $k$, for the inner loop, we have*

$$\mathbb{E}[\|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2] \leq (1 - \beta_k \mu_g)^t \mathbb{E}[\|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2] + \frac{\beta_k \sigma_{g,1}^2}{B \mu_g} \tag{19}$$

**Lemma 6 (Bounded Norm of Jacobian)** *Suppose Assumptions x hold, then for $\forall t \in [T-1], k \in [N]$, the Jacobian $\mathcal{J}^t(\mathbf{x}^k) = \frac{\partial \mathbf{y}^{k,N}(\mathbf{x}^k)}{\partial \mathbf{x}^k}$ has bounded norm:*

$$\mathbb{E}[\|\mathcal{J}^t(\mathbf{x}^k)\|_F \leq \frac{\ell_{g,1}}{\mu_g}, \tag{20}$$

**Lemma 7 (Error Between $\mathcal{J}^{k,t} - \mathcal{J}^*$ )** *Give the definition of $\mathcal{J}^N(\mathbf{x}^k)$ and $\mathcal{J}^*(\mathbf{x}^k)$, for the stochastic case we have*

$$\|\mathcal{J}^t(\mathbf{x}^k; \xi_{\mathbf{y}}) - \mathcal{J}^*(\mathbf{x}^k)\|_F^2 \leq C_1 \|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + C_2 + C_3 \tag{21}$$

*where $C_1 = 2(C_{xy} + C_{yy}) \ell_{g,2}^2 t (1 - \beta_k \mu_g)^t$, $C_2 = (1 - \beta_k \mu_g)^t \frac{\ell_{g,1}^2}{\mu_g^2}$, $C_3 = 2 \frac{1}{\beta \mu_g} (C_{xy} + C_{yy})(\sigma_{g,2}^2 + \ell_{g,2}^2 \sigma_{g,1}^2 \frac{\beta_k}{\mu_g})$, and $C_{xy} = \left(\beta_k + \gamma(1 - \beta_k \mu_g) + \beta_k \frac{\ell_{g,1}}{\mu_g}\right)$, and $C_{yy} = \beta_k C_{xy}$.*

**Lemma 8 (Smoothness of $\mathcal{J}^*(\mathbf{x})$ )** *Recalling the definition of $\mathcal{J}^*(\mathbf{x}) = \frac{\partial \mathbf{y}^*(\mathbf{x})}{\partial \mathbf{x}}$, for any $\mathbf{x}', \mathbf{x}$ there is*

$$\|\mathcal{J}^*(\mathbf{x}^1) - \mathcal{J}^*(\mathbf{x})\| \leq \ell_{yx} \|\mathbf{x}^1 - \mathbf{x}\| \tag{22}$$

*where $\ell_{yx} := \frac{\ell_{g,2} + \ell_{g,2} \ell_{\mathbf{y}}}{\mu_g} + \frac{\ell_{g,1}(\ell_{g,2} + \ell_{g,2} \ell_{\mathbf{y}})}{\mu_g^2}$.*

**Lemma 9 (Repeat of Lemma 1-2 in the Manuscript)** *Given $\mathbf{y}^{k+1,0} = \mathbf{y}^{k,N}$, there is*

$$\begin{aligned}\|\mathbf{y}^{k+1,0} - \mathbf{y}^*(\mathbf{x}^{k+1})\|^2 \leq &\left(1 + \gamma_k + \frac{\ell_{yx} C_f^2 \alpha_k^2}{4}\right) \mathbb{E}[\|\mathbf{y}^{k+1,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2] \\ &+ (\ell_y^2 \alpha_k^2 + \frac{\alpha_k^2 \ell_{g,1}^2}{4 \mu_g^2 \gamma_k} + \frac{\alpha_k^2 \ell_{yx}}{4\eta}) \mathbb{E}[\widehat{\nabla} F(\mathbf{x}^k)] + (\ell_y^2 \alpha_k^2 + \frac{\ell_{yx} \alpha_k^2}{4\eta}) \widetilde{\sigma}^2\end{aligned} \tag{23}$$

*where $\ell_y = \frac{\ell_{g,1}}{\mu_g}$, $C_f^2 = (\ell_{f,0} + \frac{\ell_{g,1}}{\mu_g} \ell_{f,1})^2 + \delta_\mu^2 + \widetilde{\sigma}^2$, $\delta_\mu^2 = \frac{\mu^2 \ell_\mu^2 q (p+6)^2}{2Q}$, $\ell_\mu = (1 + \frac{\ell_{g,1}}{\mu_g})(\frac{\ell_{g,2}}{\mu_g} + \frac{\ell_{f,1} \ell_{g,1}}{\mu_g^2})$, and $\widetilde{\sigma}^2 = 2 \frac{\sigma_f^2}{B} + 2(\ell_{f,0}^2 + \frac{\sigma_f^2}{B})(12 \delta_\mu^2 + \frac{4(2p+8)}{Q} \frac{\ell_{g,1}^2}{\mu_g^2}) + 4(\frac{\ell_{g,1}^2}{\mu_g^2} + \delta_\mu^2) \frac{\sigma_f^2}{B}$.*

**Lemma 10 (Repeat of Lemma 2 in the Manuscript)** *For the outer loop, we have*

$$\begin{aligned}
&\mathbb{E}[F(\mathbf{x}^{k+1})|\mathcal{F}_k] - \mathbb{E}[F(\mathbf{x}^k)|\mathcal{F}_{k-1}] \\
&\overset{(b)}{\leq} -\frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k)\|^2 - (\frac{\alpha_k}{2} - \frac{\ell_{f,1}\alpha_k^2}{2})\|\widehat{\nabla} F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1}\alpha_k^2}{2}\tilde{\sigma}^2 \\
&+ \alpha_k \left( \frac{\mu^2}{2}\ell_{f,0}^2\ell_\mu^2 q(p+3)^2 + 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2)\frac{\beta_k\sigma_{g,1}^2}{\mu_g} + 2\ell_{f,0}^2(C_2+C_3) \right) \\
&+ \alpha_k \left( 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2) + 4\ell_{f,0}^2\ell_{g,2}^2\beta_k(C_{xx}+C_{yy})N \right)(1-\beta_k\mu_g)^N\|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2
\end{aligned} \tag{24}$$

*where* $\tilde{\sigma}^2 = 2\sigma_f^2 + 2(\ell_{f,0}^2 + \sigma_f^2)\sigma_{\mathcal{J}}^2 + 2(2\frac{\ell_{g,1}^2}{\mu_g^2} + \mu^2\ell_\mu^2 q(p+3)^2)\sigma_f^2$

**Proof of Lemma 5** Given $\mathbf{x}^k$ fixed, let $\mathbf{v}^{k,t} = [\mathbf{v}_1^{k,t}, \cdots, \mathbf{v}_l^{k,t}]$ and, similarly, $\hat{\mathbf{v}}^{k,t} = [\hat{\mathbf{v}}_q, \cdots, \hat{\mathbf{v}}_l]$, for $\mathbf{y}^{k,t}, t = 0, \cdots, N$ (for notation abbreviation, we denote $\mathbf{y}^{k,t}$ as $\mathbf{y}^t$, and $\mathbf{y}^*(\mathbf{x}^k)$ as $\mathbf{y}^*$), in the inner loop, we have

$$\begin{aligned}
&\mathbb{E}\|\mathbf{y}^{t+1} - \mathbf{y}^*\|^2 = \|\mathbf{y}^t - \beta_k\mathbf{v} - \mathbf{y}^*\|^2 \\
&= \|\mathbf{y}^t - \mathbf{y}^*\|^2 + \beta_k^2\|\mathbf{v}\|^2 - 2\beta_k\mathbb{E}\left\langle \mathbf{y}^t - \mathbf{y}^*, \nabla g(\mathbf{y}^t)\right\rangle \\
&\overset{(a)}{\leq} \|\mathbf{y}^t - \mathbf{y}^*\|^2 + \beta_k^2\|\nabla g(\mathbf{y}^{k,t})\|^2 + \beta_k^2\|\mathbf{v} - \nabla g(\mathbf{y}^t)\|^2 - 2\beta_k\mathbb{E}\left\langle \mathbf{y}^t - \mathbf{y}^*, \nabla g(\mathbf{y}^t)\right\rangle \\
&\overset{(b)}{\leq} \|\mathbf{y}^t - \mathbf{y}^*\|^2 + \beta_k^2\|\nabla g(\mathbf{y}^{k,t})\|^2 + \beta_k^2\|\mathbf{v} - \nabla g(\mathbf{y}^t)\|^2 - 2\beta_k(g(\mathbf{y}^t) - \mathbf{y}^* + \frac{\mu_g}{2}\|\mathbf{y}^t - \mathbf{y}^*\|) \\
&\overset{(c)}{\leq} (1-\beta_k\mu_g)\|\mathbf{y}^t - \mathbf{y}^*\|^2 - 2\beta_k(1-\beta_k\ell_{g,1})(g(\mathbf{y}^t) - g^*) + \beta_k^2\sum_{m=1}^{l}\frac{\sigma_{g,1,m}^2}{B} \\
&\overset{(d)}{\leq} (1-\beta_k\mu_g)\|\mathbf{y}^t - \mathbf{y}^*\|^2 + \beta_k^2\frac{\sigma_{g,1}^2}{B}
\end{aligned} \tag{25}$$

where (a) follows from $\mathbb{E}\|X\|^2 \leq \|\mathbb{E}X\|^2 + \|X - \mathbb{E}X\|^2$, (b) uses the $\mu_g$-strongly convexity of $g$, (c) uses $\|\nabla g(\mathbf{y}^t) - \nabla g(\mathbf{y}^*)\|^2 \leq 2\ell_{g,1}(g(\mathbf{y}^t) - g^*)$ and Assumption 3, (d) follows from that we let $\beta_k \leq \frac{\ell_{g,1}}{2}$ and $\sigma_{g,1}^2 = \sum_{m=1}^{l}\sigma_{g,1,m}^2$. Reusing above equality and using the complete notations, there is

$$\|\mathbf{y}^{k,t} - \mathbf{y}^*(\mathbf{x}^k)\|^2 \leq (1-\beta_k\mu_g)^t\|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + \frac{\beta_k\sigma_{g,1}^2}{B\mu_g} \tag{26}$$

this completes the proof

**Proof of Lemma 6** According to the updating rule of Algorithm 1, there is

$$\mathbf{y}^{k,t+1} = \mathbf{y}^{k,t} - \beta_k\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}), \quad t = 0, \cdots, N-1 \tag{27}$$

Taking derivatives w.r.t. $\mathbf{x}^k$ yields:

$$\begin{aligned}
\mathcal{J}^{k,t+1} &= \mathcal{J}^{k,t} - \beta_k\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \mathcal{J}^{k,t}\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) \\
&= \mathcal{J}^{k,t}(I - \beta_k\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}})) - \beta_k\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}),
\end{aligned} \tag{28}$$

where $\mathcal{J}^{k,t+1}$ is the abbreviation of $\mathcal{J}^{t+1}(\mathbf{x}^k)$. Telescoping over $t$ from 1 to $N-1$ yields

$$\begin{aligned}
\mathcal{J}^{k,N} &= \mathcal{J}^{k,0}\prod_{t=0}^{N-1}\left(I - \beta_k\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}})\right) \\
&\quad - \beta_k\sum_{t=0}^{N-1}\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}})\prod_{i=t+1}^{N-1}\left(I - \beta_k\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}})\right) \\
&= -\beta_k\sum_{t=0}^{N-1}\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g\left(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}\right)\prod_{i=t+1}^{N-1}\left(I - \beta_k\nabla_{\mathbf{y}}^2 g\left(\mathbf{x}^k, \mathbf{y}^{k,i}; \xi_{\mathbf{y}}^{k,i}\right)\right)
\end{aligned} \tag{29}$$

Hence, we have

$$\begin{aligned}\left\|\mathcal{J}^{k,N}\right\|_F &\leq \beta_k \sum_{t=0}^{N-1} \left\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g\left(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}\right)\right\|_F \left\| \prod_{m=i+1}^{N-1} \left(I - \beta_k \nabla_{\mathbf{y}}^2 g\left(\mathbf{x}^k, \mathbf{y}^{k,i}; \xi_{\mathbf{y}}^{k,i}\right)\right)\right\| \\ &\overset{(i)}{\leq} \beta_k \sum_{t=0}^{N-1} \ell_{g,1} \prod_{m=t+1}^{N-1} \left\|I - \beta_k \nabla_{\mathbf{y}}^2 g\left(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}^{k,t}\right)\right\| \\ &\overset{(ii)}{\leq} \alpha \ell_{g,1} \sum_{t=0}^{N-1} (1 - \beta_k \mu_g)^{N-1-t} \\ &= \beta_k \ell_{g,1} \sum_{t=0}^{N-1} (1 - \beta_k \mu_g)^t \leq \frac{\ell_{g,1}}{\mu_g}\end{aligned}$$

where (i) uses Assumption 1, (ii) uses the strongly-convexity of $g$, and the last inequality follows from the characteristic of geometric progression. This directly means

$$\left\|\mathcal{J}^{N}(\mathbf{x}^k)\right\|_F \leq \frac{\ell_{g,1}}{\mu_g}$$

This completes the proof.

**Proof of Lemma 7** In the following proof, the $\mathcal{J}^{t+1}(\mathbf{x}^k; \xi_{\mathbf{y}})$ is simplified as $\mathcal{J}^{k,t+1}$ and $\mathcal{J}^{k,*}$ denotes $\mathcal{J}^*(\mathbf{x}^k)$.

$$\mathcal{J}^{k,t+1} = \mathcal{J}^{k,t} - \beta_k \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \beta_k \mathcal{J}^{k,t} \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) \tag{30}$$

and thus

$$\begin{aligned}&\mathcal{J}^{k,t+1} - \mathcal{J}^{k,*} \\ &= \mathcal{J}^{k,t} - \mathcal{J}^{k,*} - \beta_k(\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))) \\ &- \beta_k(\mathcal{J}^{k,t} - \mathcal{J}^{k,*})\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \beta_k \mathcal{J}^{k,*}(\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k)) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}))\end{aligned} \tag{31}$$

Using triangle inequality , we have

$$\begin{aligned}\|\mathcal{J}^{k,t+1} - \mathcal{J}^{k,*}\|_F = &\|(\mathcal{J}^{k,t} - \mathcal{J}^{k,*})(I - \beta_k \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}))\|_F \\ &+ \beta_k \|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F \\ &+ \beta_k \|\mathcal{J}^{k,*}(\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k)))\|_F\end{aligned} \tag{32}$$

and then there is

$$\begin{aligned}&\|\mathcal{J}^{k,t+1} - \mathcal{J}^{k,*}\|_F^2 \\ &\leq (1 - \beta_k \mu_g)^2 \|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2 + \beta_k^2 \|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2 \\ &+ \beta_k^2 \frac{\ell_{g,1}^2}{\mu_g^2} \|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2 \\ &+ 2\beta_k(1 - \beta_k \mu_g) \underbrace{\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*})\|_F \|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F}_{Q_1} \\ &+ 2\beta_k(1 - \beta_k \mu_g) \underbrace{\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F \|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F}_{Q_2} \\ &+ 2\beta_k^2 \frac{\ell_{g,1}}{\mu_g} \underbrace{\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F \|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F}_{Q_3}\end{aligned} \tag{33}$$

Especially,

$$Q_1 \leq \frac{1}{2\gamma} \|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2 + \frac{\gamma}{2} \|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2$$

$$Q_2 \leq \frac{1}{2\gamma}\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2 + \frac{\gamma}{2}\|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2$$

$$Q_3 \leq \frac{1}{2\gamma}\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2 \tag{34}$$

$$+ \frac{\gamma}{2}\|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2 \tag{35}$$

Combine Eqs. 33 and 34, we have

$$\begin{aligned}
&\|\mathcal{J}^{k,t+1} - \mathcal{J}^{k,*}\|_F^2 \qquad (36)\\
&\leq \left((1-\beta_k\mu_g)^2 + \frac{\beta_k}{\gamma}(1-\beta_k\mu_g) + \frac{\beta_k\ell_{g,1}}{\gamma\mu_g}(1-\beta_k\mu_g)\right)\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2\\
&+ \left(\beta_k^2 + \beta_k\gamma(1-\beta_k\mu_g) + \beta_k^2\frac{\ell_{g,1}}{\mu_g}\right)\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2\\
&+ \left(\beta_k^2\frac{\ell_{g,1}}{\mu_g} + \beta_k\gamma\frac{\ell_{g,1}}{\mu_g}(1-\beta_k\mu_g) + \beta_k^2\frac{\ell_{g,1}}{\mu_g}\right)\|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))\|_F^2
\end{aligned}$$

Let $\mathcal{F}_{k,t}$ denotes all randomness of iteration $k, t-1$. Conditioning on $\mathcal{F}_{k,t}$ and taking exception, we have

$$\begin{aligned}
&\|\mathcal{J}^{k,t+1} - \mathcal{J}^{k,*}|\mathcal{F}_{k,t}\|_F^2\\
&\leq \beta_k C_{xy}\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))|\mathcal{F}_{k,t}\|_F^2 \qquad (37)\\
&\quad C_\gamma\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}|\mathcal{F}_{k,t}\|_F^2 + \beta_k C_{yy}\|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}; \xi_{\mathbf{y}}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))|\mathcal{F}_{k,t}\|_F^2\\
&\leq C_\gamma\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}|\mathcal{F}_{k,t}\|_F^2 + \beta_k C_{xy}(2\|\nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^{k,t}) - \nabla_{\mathbf{x}}\nabla_{\mathbf{y}} g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))|\mathcal{F}_{k,t}\|_F^2 + 2\frac{\sigma_{g,2}^2}{B})\\
&\quad + \beta_k C_{yy}(2\|\nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^{k,t}) - \nabla_{\mathbf{y}}^2 g(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}^k))|\mathcal{F}_{k,t}\|_F^2 + 2\frac{\sigma_{g,2}^2}{B})\\
&\leq C_\gamma\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}|\mathcal{F}_{k,t}\|_F^2 + \beta_k C_{xy}(2\ell_{g,2}^2\|\mathbf{y}^{k,t} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2\frac{\sigma_{g,2}^2}{B})\\
&\quad + \beta_k C_{yy}(2\ell_{g,2}^2\|\mathbf{y}^{k,t} - \mathbf{y}^{k,*}\|^2 + 2\frac{\sigma_{g,2}^2}{B}) \qquad (38)\\
&\leq C_\gamma\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}|\mathcal{F}_{k,t}\|_F^2 + 2\beta_k(C_{xy} + C_{yy})\ell_{g,2}^2\|\mathbf{y}^{k,t} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2(C_{xy} + C_{yy})\frac{\sigma_{g,2}^2}{B}
\end{aligned}$$

where

$$\begin{aligned}
C_\gamma &= (1-\beta_k\mu_g)(1 - \beta_k\mu_g + \frac{\beta_k}{\gamma} + \frac{\beta_k\ell_{g,1}}{\gamma\mu_g}) \qquad (39)\\
C_{xy} &= \left(\beta_k + \gamma(1-\beta_k\mu_g) + \beta_k\frac{\ell_{g,1}}{\mu_g}\right), \quad C_{yy} = \beta_k C_{xy}
\end{aligned}$$

Taking total exceptions of the above equation and using lemma x, there is

$$\begin{aligned}
&\mathbb{E}\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2\\
&\leq C_\gamma\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2 + 2\beta_k(C_{xy} + C_{yy})\ell_{g,2}^2\|\mathbf{y}^{k,t} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2\beta_k(C_{xy} + C_{yy})\frac{\sigma_{g,2}^2}{B} \qquad (40)
\end{aligned}$$

Telescoping eq over $t$ yields

$$\begin{aligned}
&\mathbb{E}\|\mathcal{J}^{k,t} - \mathcal{J}^{k,*}\|_F^2 \qquad (41)\\
&\leq\ C_\gamma^N\|\mathcal{J}^{k,0} - \mathcal{J}^{k,*}\|_F^2 + 2\beta_k(C_{xy} + C_{yy})\ell_{g,2}^2\sum_{t=0}^{N-1} C_\gamma^{(N-1-t)}\|\mathbf{y}^{k,t} - \mathbf{y}^*(\mathbf{x}^k)\|^2\\
&\quad + 2\beta_k(C_{xy} + C_{yy})\frac{\sigma_{g,2}^2}{B}\sum_{t=0}^{N-1} C_\gamma^t \qquad (42)
\end{aligned}$$

$$
\begin{aligned}
&\leq C_\gamma^N \|\mathcal{J}^{k,0} - \mathcal{J}^{k,*}\|_F^2 + 2\beta_k(C_{xy}+C_{yy})\frac{\sigma_{g,2}^2}{B}\sum_{t=0}^{N-1} C_\gamma^t \\
&+ 2\beta_k(C_{xy}+C_{yy})\ell_{g,2}^2 \sum_{t=0}^{N-1} C_\gamma^{(N-1-t)} \left((1-\beta_k\mu_g)^t \|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + \frac{\beta_k\sigma_{g,1}^2}{B\mu_g}\right) \\
&\leq C_\gamma^N \|\mathcal{J}^{k,0} - \mathcal{J}^{k,*}\|_F^2 + 2\beta_k(C_{xy}+C_{yy})\ell_{g,2}^2 \sum_{t=0}^{N-1} C_\gamma^{(N-1-t)}(1-\beta_k\mu_g)^t \|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2 \\
&+ 2\beta_k(C_{xy}+C_{yy})\frac{\sigma_{g,2}^2}{B}\sum_{t=0}^{N-1} C_\gamma^t + 2\beta_k(C_{xy}+C_{yy})\ell_{g,2}^2 \sum_{t=0}^{N-1} C_\gamma^{(N-1-t)}\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}
\end{aligned}
$$

Choose $\gamma \geq \frac{\ell_{g,1}+\mu_g}{\mu_g^2}$ such $C_\gamma \leq 1-\beta_k\mu_g$, then we obtain

$$
\|\mathcal{J}^{k,t}(\mathbf{x}^k;\xi_\mathbf{y}) - \mathcal{J}^*(\mathbf{x}^k)\|_F^2 \leq C_1\|\mathbf{y}^{k,0} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + C_2 + C_3 \tag{43}
$$

where $C_1 = 2\beta_k(C_{xy}+C_{yy})\ell_{g,2}^2 t(1-\beta_k\mu_g)^t$, $C_2 = (1-\beta_k\mu_g)^t\frac{\ell_{g,1}^2}{\mu_g^2}$, $C_3 = 2\frac{1}{\mu_g}(C_{xy}+C_{yy})(\frac{\sigma_{g,2}^2}{B} + \ell_{g,2}^2\frac{\sigma_{g,1}^2\beta_k}{B\mu_g})$. This completes the proof.

**Proof of Lemma 8** Recalling the definition of $\mathcal{J}^*(\mathbf{x}) = \frac{\partial \mathbf{y}^*(\mathbf{x})}{\partial \mathbf{x}}$, for any $\mathbf{x}', \mathbf{x}$, we have

$$
\begin{aligned}
&\|\mathcal{J}^*(\mathbf{x}') - \mathcal{J}^*(\mathbf{x})\| \qquad (44)\\
&= \|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))]^{-1} - \nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}, y^*(\mathbf{x}))]^{-1}\| \\
&\leq \|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\|\|[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))]^{-1}\| \qquad (45)\\
&\quad + \|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\|\|[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))]^{-1} - [\nabla_{\mathbf{yy}}^2 g(\mathbf{x}, y^*(\mathbf{x}))]^{-1}\| \\
&\overset{(a)}{\leq} \frac{1}{\mu_g}\|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\quad + \ell_{g,1}\|[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}'))]^{-1}\left(\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{yy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right)[\nabla_{\mathbf{yy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))]^{-1}\| \\
&\overset{(b)}{\leq} \frac{1}{\mu_g}\|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| + \frac{\ell_{g,1}}{\mu_g^2}\|\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{yy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\|
\end{aligned}
$$

where both (a) and (b) follow from Assumption 1. In addition, we have that

$$
\frac{1}{\mu_g}\|\nabla_{\mathbf{xy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| + \frac{\ell_{g,1}}{\mu_g^2}\|\nabla_{\mathbf{yy}}^2 g(\mathbf{x}', \mathbf{y}^*(\mathbf{x}')) - \nabla_{\mathbf{yy}}^2 g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \tag{46}
$$

$$
\begin{aligned}
&\leq \frac{\ell_{g,2}}{\mu_g}\|\mathbf{x}' - \mathbf{x}\| + \frac{\ell_{g,2}}{\mu_g}\|\mathbf{y}^*(\mathbf{x}') - \mathbf{y}^*(\mathbf{x})\| + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g^2}\|\mathbf{x}' - \mathbf{x}\| + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g^2}\|\mathbf{y}^*(\mathbf{x}') - \mathbf{y}^*(\mathbf{x})\| \\
&\overset{(c)}{\leq} \left(\frac{\ell_{g,2} + \ell_{g,2}\ell_y}{\mu_g} + \frac{\ell_{g,1}(\ell_{g,2} + \ell_{g,2}\ell_y)}{\mu_g^2}\right)\|\mathbf{x}' - \mathbf{x}\|
\end{aligned}
$$

where (c) follows from $\|\mathbf{y}^*(\mathbf{x}') - \mathbf{y}^*(\mathbf{x})\| \leq \ell_\mathbf{y}\|\mathbf{x}' - \mathbf{x}\|$, where $\ell_y = \frac{\ell_{g,1}}{\mu_g}$ (please refer to Lemma 2.2 in Ghadimi & Wang (2018) for the detailed proofs). As a result, we have

$$
\|\mathcal{J}^*(\mathbf{x}^1) - \mathcal{J}^*(\mathbf{x})\| \leq \ell_{yx}\|\mathbf{x}^1 - \mathbf{x}\| \tag{47}
$$

where $\ell_{yx} := \frac{\ell_{g,2}+\ell_{g,2}\ell_y}{\mu_g} + \frac{\ell_{g,1}(\ell_{g,2}+\ell_{g,2}\ell_y)}{\mu_g^2}$. Similarly, we hvae

$$
C_f^2 := 2\left(\ell_{f,0} + \frac{\ell_{g,1}}{\mu_g}\ell_{f,1}\right)^2 + 2\delta_\mu^2 + \tilde{\sigma}^2 = \mathcal{O}(\kappa^2). \tag{48}
$$

This completes the proof.

**Proof of Lemma 9** We start this by decomposing the error of the lower level variable as

$$\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^{k+1})\|^2=\underbrace{\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|^2}_{Q_4}+\underbrace{\|\mathbf{y}^*(\mathbf{x}^{k+1})-\mathbf{y}^*(\mathbf{x}^k)\|^2}_{Q_5}+\underbrace{2\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathbf{y}^*(\mathbf{x}^{k+1})-\mathbf{y}^*(\mathbf{x}^k)\right\rangle}_{Q_6} \tag{49}$$

Note that, $\mathbf{y}^{k+1,0}=\mathbf{y}^{k,N}$, thus

$$\begin{aligned}\mathbb{E}Q_4&=\mathbb{E}\|\mathbf{y}^{k,N}-\mathbf{y}^*(\mathbf{x}^k)\|^2\leq(1-\beta_k\mu_g)^t\|\mathbf{y}^{k,0}-\mathbf{y}^*\|^2+\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}\\&=(1-\beta_k\mu_g)^t\|\mathbf{y}^k-\mathbf{y}^{k,*}\|^2+\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}\end{aligned} \tag{50}$$

The upper bound of $Q_5$ can be obtained as

$$\begin{aligned}\mathbb{E}[Q_5]&=\mathbb{E}[\|\mathbf{y}^*(\mathbf{x}^{k+1})-\mathbf{y}^*(\mathbf{x}^k)\|^2]\leq\ell_{\mathbf{y}}^2\mathbb{E}\|\mathbf{x}^{k+1}-\mathbf{x}^k||^2\\&=\ell_{\mathbf{y}}^2\alpha_k^2\mathbb{E}[\mathbb{E}[\bar{\nabla}F(\mathbf{x}^k)-\widehat{\nabla}F(\mathbf{x}^k)+\widehat{\nabla}F(\mathbf{x}^k)\|\mathcal{F}_{k,t}\|]]\leq\ell_{\mathbf{y}}^2\alpha_k^2(\mathbb{E}\|\widehat{\nabla}F(\mathbf{x}^k)\|^2+\tilde{\sigma}^2)\end{aligned} \tag{51}$$

where the last equality follows from Eq. 62. The term $Q_6$ can be decomposed as

$$\begin{aligned}\mathbb{E}[Q_6]=&\underbrace{-\mathbb{E}[\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathcal{J}^*(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k)\right\rangle]}_{Q_6^1}\\&\underbrace{-\mathbb{E}[\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathbf{y}^*(\mathbf{x}^{k+1})-\mathbf{y}^*(\mathbf{x}^k)-\mathcal{J}^*(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k)\right\rangle]}_{Q_6^1}\end{aligned} \tag{52}$$

As for $Q_6^1$ there is

$$\begin{aligned}\mathbb{E}[Q_6^1]&=-\mathbb{E}[\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathbb{E}[\mathcal{J}^*(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k)|\mathcal{F}_{k,t}]\right\rangle]\\&=-\alpha_k\mathbb{E}[\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathcal{J}^*(\mathbf{x}^k)\widehat{\nabla}F(\mathbf{x}^k)\right\rangle]\\&\leq\alpha_k\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|\|\mathcal{J}^*(\mathbf{x}^k)\widehat{\nabla}F(\mathbf{x}^k)\|]\\&\leq\alpha_k\frac{\ell_{g,1}}{\mu_g}\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|\|\widehat{\nabla}F(\mathbf{x}^k)\|]\\&\leq\gamma_k\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|]+\frac{\alpha_k^2\ell_{g,1}^2}{4\mu_g^2\gamma_k}\mathbb{E}\|\widehat{\nabla}F(\mathbf{x}^k)\|\end{aligned} \tag{53}$$

As for $Q_6^2$, there is

$$\begin{aligned}\mathbb{E}[Q_6^2]&=-\mathbb{E}[\left\langle\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k),\mathbf{y}^*(\mathbf{x}^{k+1})-\mathbf{y}^*(\mathbf{x}^k)-\mathcal{J}^*(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k)\right\rangle]\\&=\frac{\ell_{yx}}{2}\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2]\\&\leq\frac{\eta\ell_{yx}}{4}\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|^2\mathbb{E}[\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2|\mathcal{F}_{k,t}]]+\frac{\ell_{yx}\alpha_k^2}{4\eta}\mathbb{E}[\mathbb{E}[\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2|\mathcal{F}_{k,t}]]\\&\leq\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|^2]+\frac{\ell_{yx}\alpha_k^2}{4\eta}(\mathbb{E}[\widehat{\nabla}F(\mathbf{x}^k)]+\tilde{\sigma}^2)\end{aligned} \tag{54}$$

Combining Eqs. 52, 53, and 54, we have

$$\mathbb{E}Q_6=\left(\gamma_k+\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|^2]+(\frac{\alpha_k^2\ell_{g,1}^2}{4\mu_g^2\gamma_k}+\frac{\ell_{yx}\alpha_k^2}{4\eta})\mathbb{E}[\widehat{\nabla}F(\mathbf{x}^k)]+\frac{\ell_{yx}\alpha_k^2}{4\eta}\tilde{\sigma}^2) \tag{55}$$

Combining EqS. 49, 50, 51, and 55, there is

$$\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^{k+1})\|^2\leq\left(1+\gamma_k+\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\mathbb{E}[\|\mathbf{y}^{k+1,0}-\mathbf{y}^*(\mathbf{x}^k)\|^2]$$

$$+ (\ell_y^2\alpha_k^2 + \frac{\alpha_k^2\ell_{g,1}^2}{4\mu_g^2\gamma_k} + \frac{\ell_{yx}\alpha_k^2}{4\eta})\mathbb{E}[\widehat{\nabla}F(\mathbf{x}^k)] + (\ell_y^2\alpha_k^2 + \frac{\ell_{yx}\alpha_k^2}{4\eta})\tilde{\sigma}^2) \quad (56)$$

This completes the proof.

**Proof of Lemma 10** Note that, In the following, we still use $\mathcal{F}^{k,t}$ to denote the filtration that captures all randomness at iteration stamp $k, t$

$$\mathbb{E}[F(\mathbf{x}^{k+1})|\mathcal{F}_{k,t}] \leq F(\mathbf{x}^k) + \mathbb{E}[\left\langle \nabla^* F(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \right\rangle |\mathcal{F}_{k,t}] + \frac{\ell_{f,1}}{2}\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2|\mathcal{F}_{k,t}]$$

$$= F(\mathbf{x}^k) - \alpha_k \left\langle \nabla^* F(\mathbf{x}^k), \widehat{\nabla}F(\mathbf{x}^k) \right\rangle + \frac{\ell_{f,1}\alpha_k^2}{2}\mathbb{E}[\|\bar{\nabla}F(\mathbf{x}^k)\|^2|\mathcal{F}_{k,t}] \quad (57)$$

$$\overset{(a)}{=} F(\mathbf{x}^k) - \frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k)\|^2 - \frac{\alpha_k}{2}\|\widehat{\nabla}F(\mathbf{x}^k)\|^2 + \frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2 \quad (58)$$

$$+ \frac{\ell_{f,1}\alpha_k^2}{2}\|\widehat{\nabla}F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1}\alpha_k^2}{2}\mathbb{E}[\|\bar{\nabla}F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2|\mathcal{F}_{k,t}] \quad (59)$$

$$\overset{(b)}{\leq} F(\mathbf{x}^k) - \frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k)\|^2 - (\frac{\alpha_k}{2} - \frac{\ell_{f,1}\alpha_k^2}{2})\|\widehat{\nabla}F(\mathbf{x}^k)\|^2 \quad (60)$$

$$+ \frac{\alpha_k}{2}\|\nabla^* F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1}\alpha_k^2}{2}\mathbb{E}[\|\bar{\nabla}F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2|\mathcal{F}_{k,t}] \quad (61)$$

**First we bound term** $\mathbb{E}[\|\bar{\nabla}F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2|\mathcal{F}_{k,t}]$.

$$\mathbb{E}[\|\bar{\nabla}F(\mathbf{x}^k) - \widehat{\nabla}F(\mathbf{x}^k)\|^2|\mathcal{F}_{k,t}] \quad (62)$$

$$= \mathbb{E}[\|\nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \hat{\mathcal{J}}^{k,N}\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}) - \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}; \xi_{\mathbf{x}}^k)$$
$$- \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}; \xi_y^k)\|^2|\mathcal{F}_{k,t}] \quad (63)$$

$$\leq 2\mathbb{E}\|\nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}; \xi_{\mathbf{x}}^k)\|^2$$
$$+ 2\mathbb{E}[\|\hat{\mathcal{J}}^{k,N}\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}) - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}; \xi_y^k)\|^2|\mathcal{F}_{k,t}] \quad (64)$$

$$\leq 2\sigma_f^2 + 2\mathbb{E}[\|\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}; \xi_y^k)\|^2\|\hat{\mathcal{J}}^{k,N} - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|^2|\mathcal{F}_{k,t}]$$
$$+ 2\mathbb{E}[\|\hat{\mathcal{J}}^{k,N}\|^2\|\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}) - \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}; \xi_y^k)\|^2|\mathcal{F}_{k,t}] \quad (65)$$

$$\leq 2\sigma_f^2 + 2\mathbb{E}[\|\nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}^{k,N}; \xi_y^k)\|^2\|\hat{\mathcal{J}}^{k,N} - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|^2|\mathcal{F}_{k,t}] + 2\|\hat{\mathcal{J}}^{k,N}\|^2\sigma_f^2$$
$$\leq 2\sigma_f^2 + 2(\ell_{f,0}^2 + \sigma_f^2)\|\hat{\mathcal{J}}^{k,N} - \mathcal{J}^{k,N} + \mathcal{J}^{k,N} - \mathcal{J}^{k,N}(\xi_y) + \mathcal{J}^{k,N}(\xi_y) - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|^2$$
$$+ 2(2\|\mathcal{J}^{k,N}\|^2 + 2\|\hat{\mathcal{J}}^{k,N} - \mathcal{J}^{k,N}\|^2)\sigma_f^2 \quad (66)$$

$$\leq 2\sigma_f^2 + 2(\ell_{f,0}^2 + \sigma_f^2)(12\delta_\mu + \frac{4(2p+8)}{Q}\frac{\ell_{g,1}^2}{\mu_g^2}) + 2(2\frac{\ell_{g,1}^2}{\mu_g^2} + 2\delta_\mu)\sigma_f^2$$
$$= \tilde{\sigma}^2 \quad (67)$$

where we use $\|\hat{\mathcal{J}}^{k,N} - \mathcal{J}^{k,N}\|_F^2 \leq \frac{\mu^2\ell_\mu^2 q(p+3)^2}{2Q} \leq \frac{\mu^2\ell_\mu^2 q(p+6)^2}{2Q} = \delta_\mu$. Further, we have

$$\|\hat{\mathcal{J}}_\mu^N - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|_F^2 \leq \quad (68)$$
$$\|\hat{\mathcal{J}}_\mu^N - \mathcal{J}^{k,N} + \mathcal{J}^{k,N} - \mathcal{J}^{k,N}(\xi_{\mathbf{y}}) + \mathcal{J}^{k,N}(\xi_{\mathbf{y}}) - \hat{\mathcal{J}}_\mu^{k,N}(\xi_{\mathbf{y}}) + \hat{\mathcal{J}}_\mu^{k,N}(\xi_{\mathbf{y}}) - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|_F^2$$
$$\leq 4\|\hat{\mathcal{J}}_\mu^N - \mathcal{J}^{k,N}\|_F^2 + 4\|\mathcal{J}^{k,N} - \mathcal{J}^{k,N}(\xi_{\mathbf{y}})\|_F^2 + 4\|\mathcal{J}^{k,N}(\xi_{\mathbf{y}}) - \hat{\mathcal{J}}_\mu^{k,N}(\xi_{\mathbf{y}})\|_F^2$$
$$+ 4\|\hat{\mathcal{J}}_\mu^{k,N}(\xi_{\mathbf{y}}) - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|_F^2 \quad (69)$$
$$\leq 12\delta_\mu + 4\frac{\ell_{g,1}^2}{\mu_g^2} + \frac{4(2p+8)}{Q}\frac{\ell_{g,1}^2}{\mu_g^2}$$

where we use Lemma 4 to obtain $\|\hat{\mathcal{J}}_\mu^{k,N}(\xi_{\mathbf{y}}) - \hat{\mathcal{J}}^{k,N}(\xi_{\mathbf{y}})\|_F^2 \leq \delta_\mu + \frac{(2p+8)}{Q}\ell_{g,1}^2\mu_g^2$.

**Then we bound term** $\|\nabla^* F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2$

$$\begin{aligned}\|\nabla^* F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2 &= \|\nabla^* F(\mathbf{x}^k) - \nabla F(\mathbf{x}^k) + \nabla F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2 \\ &= 2\|\nabla^* F(\mathbf{x}^k) - \nabla F(\mathbf{x}^k)\|^2 + 2\|\nabla F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2\end{aligned} \tag{70}$$

where for $\|\nabla F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2$ there is

$$\begin{aligned}&\|\nabla F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2| \\ &= \|\nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) + \mathcal{J}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \hat{\mathcal{J}}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \\ &\leq \|\mathcal{J}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \hat{\mathcal{J}}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \\ &\leq \|\nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \|\mathcal{J}^{k,N} - \hat{\mathcal{J}}^{k,N}\|^2 \\ &\leq \delta_\mu \ell_{f,0}^2\end{aligned} \tag{71}$$

and for $\|\nabla^* F(\mathbf{x}^k) - \nabla F(\mathbf{x}^k)\|^2$ there is

$$\begin{aligned}&\|\nabla^* F(\mathbf{x}^k) - \nabla F(\mathbf{x}^k)\|^2 \\ &= \|\nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x})) + \mathcal{J}^*(\mathbf{x}^k) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x})) - \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \mathcal{J}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \\ &\leq 2\|\nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x})) - \nabla_{\mathbf{x}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 + 2\|\mathcal{J}^*(\mathbf{x}^k) \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x})) - \hat{\mathcal{J}}^{k,N} \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \\ &\leq 2\ell_{f,1}^2 \|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2\|\nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N})\|^2 \|\mathcal{J}^*(\mathbf{x}^k) - \mathcal{J}^{k,N}(\mathbf{x}^k)\|^2 \\ &\quad + 2\|\mathcal{J}^*(\mathbf{x}^k)\|^2 \|\nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^{k,N}) - \nabla_{\mathbf{y}} f(\mathbf{x}^k, \mathbf{y}^*(\mathbf{x}))\|^2\end{aligned} \tag{72}$$

$$\leq 2\ell_{f,1}^2 \|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2 \|\mathbf{y}^{k,N} - \mathbf{y}^*(\mathbf{x}^k)\|^2 + 2\ell_{f,0}^2 \|\mathcal{J}^{k,N}(\mathbf{x}^k) - \mathcal{J}^*(\mathbf{x}^k)\|^2 \tag{73}$$

$$\begin{aligned}&\leq 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2) \left( (1 - \beta_k \mu_g)^t \|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2 + \frac{\beta_k \sigma_{g,1}^2}{B \mu_g} \right) + 2\ell_{f,0}^2 (C_1 \|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2 + C_2 + C_3) \\ &= (1 - \beta_k \mu_g)^k \left( 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2) + 4\ell_{f,0}^2 \ell_{g,2}^2 \beta_k (C_{xx} + C_{yy}) t \right) \|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2 \\ &\quad + 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2) \frac{\beta_k \sigma_{g,1}^2}{B \mu_g} + 2\ell_{f,0}^2 (C_2 + C_3)\end{aligned} \tag{74}$$

Combining Eqs. 57, 70, and 72, we have that

$$\mathbb{E}[F(\mathbf{x}^{k+1}) | \mathcal{F}_{k,t}] \overset{(b)}{\leq} F(\mathbf{x}^k) - \frac{\alpha_k}{2} \|\nabla^* F(\mathbf{x}^k)\|^2 - (\frac{\alpha_k}{2} - \frac{\ell_{f,1} \alpha_k^2}{2}) \|\widehat{\nabla} F(\mathbf{x}^k)\|^2 \tag{75}$$

$$+ \frac{\alpha_k}{2} \|\nabla^* F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1} \alpha_k^2}{2} \mathbb{E}[\|\bar{\nabla} F(\mathbf{x}^k) - \widehat{\nabla} F(\mathbf{x}^k)\|^2 | \mathcal{F}_{k,t}] \tag{76}$$

$$\begin{aligned}&\overset{(b)}{\leq} F(\mathbf{x}^k) - \frac{\alpha_k}{2} \|\nabla^* F(\mathbf{x}^k)\|^2 - (\frac{\alpha_k}{2} - \frac{\ell_{f,1} \alpha_k^2}{2}) \|\widehat{\nabla} F(\mathbf{x}^k)\|^2 + \frac{\ell_{f,1} \alpha_k^2}{2} \tilde{\sigma}^2 \\ &+ \alpha_k \left( \frac{\mu^2}{2} \ell_{f,0}^2 \ell_\mu^2 q (p+3)^2 + 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2) \frac{\beta_k \sigma_{g,1}^2}{B \mu_g} + 2\ell_{f,0}^2 (C_2 + C_3) \right) \\ &+ \alpha_k \left( 2(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2} \ell_{f,1}^2) + 4\ell_{f,0}^2 \ell_{g,2}^2 \beta_k (C_{xx} + C_{yy}) N \right) (1 - \beta_k \mu_g)^N \|\mathbf{y}^{k,0} - \mathbf{y}^*\|^2\end{aligned}$$

where $\tilde{\sigma}^2 =\leq 2\sigma_f^2 + 2(\ell_{f,0}^2 + \sigma_f^2)(12\delta_\mu + \frac{4(2p+8)}{Q} \frac{\ell_{g,1}^2}{\mu_g^2}) + 2(2\frac{\ell_{g,1}^2}{\mu_g^2} + 2\delta_\mu)\sigma_f^2$.

**Proof of Theorem 1** Following the work Chen et al. (2021), we construct the Lyapunov function as $\mathbb{V}^k := F(\mathbf{x}^k) + \|\mathbf{y}^k - \mathbf{y}^*(\mathbf{x}^k)\|^2$. Then the difference between two Lyapunov functions is formulated as

$$\mathbb{V}^{k+1} - \mathbb{V}^k = F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) + (\|\mathbf{y}^{k+1,0} - \mathbf{y}^*(\mathbf{x}^{k+1})\| - \|\mathbf{y}^k - \mathbf{y}^*(\mathbf{x}^k)\|) \tag{77}$$

We bound the first term according to Lemma 10, and the second one by Lemmas 5 and 9. Then, we have

$$\mathbb{V}^{k+1}-\mathbb{V}^k \leq -\frac{\alpha_k}{2}\mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2 - \underbrace{(\frac{\alpha_k}{2}-\frac{\ell_{f,1}\alpha_k^2}{2}-\ell_y^2\alpha_k^2-\frac{\alpha_k^2\ell_{g,1}^2}{4\mu_g^2\gamma_k}-\frac{\alpha_k^2\ell_{yx}}{4\eta})}_{Q_7}\|\widehat{\nabla}F(\mathbf{x}^k)\|^2 \tag{78}$$

$$+(\underbrace{\left(1+\gamma_k+\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}+\alpha_k\left(2(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2)+4\ell_{f,0}^2\ell_{g,2}^2\beta_k(C_{xx}+C_{yy})N\right)\right)}_{Q_8}\cdot$$

$$(1-\beta_k\mu_g)^N-1)\|\mathbf{y}^{k,0}-\mathbf{y}^*\|^2 \tag{79}$$

$$+\alpha_k\left(\frac{\mu^2}{2}\ell_{f,0}^2\ell_\mu^2 q(p+3)^2+2(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}+2\ell_{f,0}^2(C_2+C_3)\right)$$

$$+(\ell_y^2\alpha_k^2+\frac{\ell_{yx}\alpha_k^2}{4\eta}+\frac{\ell_{f,1}\alpha_k^2}{2})\tilde{\sigma}^2+\left(1+\gamma_k+\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}$$

To guarantee the descent of $\mathbb{V}^k$, the following constrains must be satisfied:

$$\begin{aligned}\alpha_k &\leq \frac{\mu_g^2}{2\ell_{f,1}\mu_g^2+4\ell_{g,1}^2+\ell_{yx}\mu_g^2}\\ \ell_{yx}C_f^2\alpha_k^2+8\alpha_k(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2+\frac{\ell_{g,1}^2}{\mu_g^2}) &\leq N\beta_k\mu_g\\ \alpha_k &\leq \frac{\mu_g}{4\ell_{f,0}^2\ell_{g,2}^2(C_{xx}+C_{yy})}\end{aligned} \tag{80}$$

with selecting $\gamma_k=\frac{\ell_{g,1}^2\alpha_k}{\mu_g^2}, \eta=1$. Defining

$$\begin{aligned}&\hat{\alpha}_1=\frac{\mu_g^2}{2\ell_{f,1}\mu_g^2+4\ell_{g,1}^2+\ell_{yx}\mu_g^2}, \quad \hat{\alpha}_2=\ell_{yx}C_f^2\alpha_k^2+8\alpha_k(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2+\frac{\ell_{g,1}^2}{\mu_g^2}),\\ &\hat{\alpha}_3=\frac{\mu_g}{8\ell_{f,0}^2\ell_{g,2}^2(\frac{1}{2\mu_g}+\frac{\ell_{g,1}+\mu_g}{2\mu_g^2}+\frac{\ell_{g,1}}{2\mu_g})(1+\frac{\ell_{g,1}}{\mu_g})}, \quad \delta_\mu=\frac{\mu^2\ell_\mu^2 q(p+6)^2}{2Q}\end{aligned} \tag{81}$$

and then, we can choose the following stepsize

$$\alpha_k=\min\{\hat{\alpha}_1,\hat{\alpha}_2,\hat{\alpha}_3,\frac{1}{\sqrt{K}}\}, \quad \beta_k=\min\{\frac{\ell_{yx}C_f^2\hat{\alpha}_1+8(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2+\frac{\ell_{g,1}^2}{\mu_g^2})}{2N\mu_g}\alpha_k,\frac{1}{2\mu_g}\}, \tag{82}$$

where $\alpha$ is dependent on the constants imposed in Assumptions, and is independent on the iteration number. With the above choice of stepsize, Eq. 78 can be simplified as

$$\mathbb{V}^{k+1}-\mathbb{V}^k \tag{83}$$

$$\leq -\frac{\alpha_k}{2}\mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2+\alpha_k\left(\frac{\mu^2}{2}\ell_{f,0}^2\ell_\mu^2 q(p+3)^2+2(\ell_{f,1}^2+2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}+2\ell_{f,0}^2(C_2+C_3)\right)$$

$$+\frac{\alpha_k^2}{4}(4\ell_y^2+\ell_{yx}+2\ell_{f,1})\tilde{\sigma}^2+\left(1+\frac{\ell_{g,1}^2}{\mu_g^2}+\frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}$$

$$\leq -\frac{\alpha_k}{2}\mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2 + \alpha_k\left(\ell_{f,0}^2\delta_\mu + \beta_k\frac{\sigma_{g,1}^2}{B\mu_g}\left(2\ell_{f,1}^2 + 4\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + 2\ell_{f,0}^2(C_2 + C_3\frac{2\ell_{g,2}^2}{\mu_g}\right)\right)$$
$$+ \frac{\alpha_k^2}{4}(4\ell_y^2 + \ell_{yx} + 2\ell_{f,1})\tilde{\sigma}^2 + \left(1 + \frac{\ell_{g,1}^2}{\mu_g^2} + \frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}$$

Telescoping above equation leads to

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^K \mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2 \qquad (84)\\
&\leq \frac{\mathbb{V}^0 + \sum_{k=1}^K \alpha_k\left(\ell_{f,0}^2\delta_\mu + \beta_k\frac{\sigma_{g,1}^2}{B\mu_g}\left(2\ell_{f,1}^2 + 4\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + 2\ell_{f,0}^2(C_2+C_3)\frac{2\ell_{g,2}^2}{\mu_g}\right)\right)}{\frac{1}{2}\sum_{k=1}^K\alpha_k}\\
&+ \frac{\sum_{k=1}^K\frac{\alpha_k^2}{4}(4\ell_y^2 + \ell_{yx} + 2\ell_{f,1})\tilde{\sigma}^2 + \sum_{k=1}^K\left(1 + \frac{\ell_{g,1}^2}{\mu_g^2} + \frac{\eta\ell_{yx}C_f^2\alpha_k^2}{4}\right)\frac{\beta_k\sigma_{g,1}^2}{B\mu_g}}{\frac{1}{2}\sum_{k=1}^K\alpha_k}\\
\leq&\alpha_k\left(2\ell_{f,1}^2 + 4\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + 2\ell_{f,0}^2(C_2+C_3)\frac{2\ell_{g,2}^2}{\mu_g}\right)\frac{\ell_{yx}C_f^2\alpha_0 + 8(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + \frac{\ell_{g,1}^2}{\mu_g^2})}{N}\frac{\sigma_{g,1}^2}{B\mu_g^2}\\
&+ \frac{\alpha_k}{2}(4\ell_y^2 + \ell_{yx} + 2\ell_{f,1})\tilde{\sigma}^2 + \frac{\mathbb{V}^0}{K\min\{\hat{\alpha}_1,\hat{\alpha}_2\}} + \frac{\mathbb{V}^0}{\alpha\sqrt{K}} + \ell_{f,0}^2\delta_\mu\\
&+ \frac{\ell_{yx}C_f^2\alpha_0 + 8(\ell_{f,1}^2 + 2\frac{\ell_{g,1}^2}{\mu_g^2}\ell_{f,1}^2 + \frac{\ell_{g,1}^2}{\mu_g^2})}{N}\left(1 + \frac{\ell_{g,1}^2}{\mu_g^2} + \frac{\eta\ell_{yx}C_f^2\alpha_0^2}{4}\right)\frac{\sigma_{g,1}^2}{B\mu_g^2}
\end{aligned}$$

Simplify above equation by ignoring the constants independent on iteration number, we have that as following

$$\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\nabla^* F(\mathbf{x}^k)\|^2 = \frac{\mathbb{V}^0}{K\min\{\hat{\alpha}_1,\hat{\alpha}_2\}} + \frac{\mathbb{V}^0}{\alpha\sqrt{K}} + c_\kappa^0\delta_\mu + \frac{\alpha c_\kappa^1}{\sqrt{K}N}\frac{\sigma_{g,1}^2}{B} + \frac{\alpha c_\kappa^2}{\sqrt{K}}\tilde{\sigma}^2 + \frac{c_\kappa^3}{N}\frac{\sigma_{g,1}^2}{B} \qquad (85)$$

where $c_\kappa^0, c_\kappa^1, c_\kappa^2, c_\kappa^3$ only dependent on the constants in Assumptions. This completes the proof.

### D.2 Label Privacy Analysis of MABMI-DP

Following the proof in Dwork et al. (2014) and Yang et al. (2022), we can drive the following proofs. Note that, the proof in Yang et al. (2022) is only suitable for the two-party split learning and single level optimization case. In the following, we generalize it to the VFBO case with general $l$-party ($l \geq 2$).

**Proof of Lemma 3 in the Manuscript:** For binary classification task, let $b = 1$, fix $\theta_b^{k,t}, \theta_{1-b}^{k,t}$. We first provide an auxiliary statement in Dwork et al. (2014). Let the universe $\mathcal{B} \subset \mathbb{R}$ be $\{0,1\}$. Consider the identical mapping $f(x) := x$ and the Laplace mechanism $f^{DP} = f + r$ with $r \sim \mathsf{Lap}(c)$. Then, there is that when $c \geq \frac{1}{\epsilon}$, $f^{DP}$ is $(\epsilon, 0)$-DP which follows from the property of Laplace mechanism Dwork et al. (2014), where we use that $\ell_1$ sensitive of $f$ is 1.

Then, we consider the deterministic mapping $h : \mathbb{R} \to \mathbb{R}^d$ defined as

$$h(b) = b \cdot \theta_1^{k,t} + (1-b)\cdot\theta_0^{k,t}.$$

Since $f^{DP}$ is $(\epsilon, 0)$-DP, and DP is immune to post processing, $h(f^{DP})$ is also $(\epsilon, 0)$-DP. On the other hand, we have

$$h(f^{DP}(0)) = \theta_0^{k,t} + r\cdot(\theta_1^{k,t} - \theta_0^{k,t}),$$
$$h(f^{DP}(1)) = \theta_1^{k,t} - r\cdot(\theta_0^{k,t} - \theta_1^{k,t}).$$

where $r$ is symmetric and $-r$ is distributed identically to $r$. Thus, we have that $h(f^{DP})$ is the same as Algorithm 2, which completes the proof of the lemma.

Moreover, it is easy to prove that Algorithm 3 for multi-class classification is also $(\epsilon, 0)$-DP.

**Proof of Lemma 2 in the Manuscript:** In the following proof, we omit the superscript and subscript of $\mathbf{x}$, $\mathbf{y}$, $h$, $H$, $\theta$, $\vartheta$ and let the random variable $\xi$ has the superscript denoting the iteration stamp, here $\mathbf{x}$, $\mathbf{y}$, $h$, $H$, $\theta$, $\vartheta$ without subscript denotes the whole variable for all parties, e.g., $\mathbf{x}$ denotes $[\mathbf{x}_1, \cdots, \mathbf{x}_l]$. In our experiments, we use the same rand seed for all parties to guarantee that all parties have the same batches for the fixed iteration stamp. Note that, although party $m = 1$ (the server) has the label, its gradient used for updating is also perturbed by the noise generated by Algorithm 2 to guarantee the $(\epsilon, 0)$-DP. BAMBI-DP is a bilevel algorithm with two time-scale for the upper- and lower-variables, thus we use superscript $k, t = N + 1$ to denote the the iteration stamp $k$ of upper-level iteration (updates for $\mathbf{x}$), e.g., $\mathbf{x}^{k,N+1}$ denotes $\mathbf{x}^k$. In this case, we can use the same time scale to simplify the analyses of BO algorithms' label DP. Moreover, we let $\mathcal{D} = \mathcal{D}_{\text{lower}} \cup \mathcal{D}_{\text{upper}}$.

For information transmitted in the forward and backward processes of BAMBI-DP, we let $G_t := \{h(\xi_{\mathbf{y}}^{k,t}), H(\xi_{\mathbf{x}}^{k,N+1,i})\}_{k\in[K],t\in[N],i\in[B]} \cup \{\widetilde{\theta}(\xi_{\mathbf{y}}^{k,t,i}), \widetilde{\vartheta}(\xi_{\mathbf{x}}^{k,N+1,i}), \}_{k\in[K],t\in[N],i\in[B]}$, where $\xi_{\mathbf{y}}^{k,t,i} \in \mathcal{D}_{\text{lower}}$, $\xi_{\mathbf{x}}^{k,N+1,i} \in \mathcal{D}_{\text{upper}}$ are both a batch of samples. We further consider the model updates at iteration stamp $k, t$, which we denote as $G_d := \{\nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{x}}^{k,N+1,i}), \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}, \xi_{\mathbf{y}}^{k,t,i})\}_{k\in[K],t\in[N],i\in[B]}$. Then $G_t, G_d$ are random functions of input dataset $\mathcal{D}$. We denote the output of $G_t$ and $G_d$ with input $\mathcal{D}$ by $(G_t, G_d)(\mathcal{D})$.

Let $\xi^{k_*,t_*,i_*}$ (here, we do not differ $\xi_{\mathbf{x}}$ and $\xi_{\mathbf{y}}$ and take them together) be the index of the different label in $D$ and $D'$, i.e., $b(\xi^{k_*,t_*,i_*}) = 1 - b'(\xi^{k_*,t_*,i_*})$. Then in batches that are sampled before $k_*, t_*$ (i.e., $\xi^{0,0}, \cdots, \xi^{0,N+1}, \cdots, \xi^{k_*,t_*-1}$), $D$ and $D'$ are identical, hence the probability restricted on the first $(k_* - 1)(N + 1) + t_* - 1$ batches is the same. Formally, let $G_-$ be the parts of $G_t, G_d$ that are in the first $(k_* - 1)(N + 1) + t_* - 1$ batches, then we have

$$\Pr\left[[(G_t, G_d)(\mathcal{D})]_{G_-} = [G]_{G_-}\right] = \Pr\left[[(G_t, G_d)(\mathcal{D}')]_{G_-} = [G]_{G_-}\right].$$

Furthermore, if $G_t$ and $G_d$ are the same for the first $(k_* - 1)(N + 1) + t_* - 1$ batches, then the model weights after the $k_*, t_*$-th batch will be the same for $D$ and $D'$, because all previous model updates are the same. Since the rest training data is identical in $D$ and $D'$, the remaining part in $(G_t, G_d)$ will also be identical. Formally, define $S_*$, $G_+$ as

$$\begin{aligned}
G_* = &\{h(\xi_{\mathbf{y}}^{k_*,t_*,i}), H(\xi_{\mathbf{x}}^{k_*,t_*,i}), \}_{i\in[B]} \cup \{\widetilde{\theta}(\xi_{\mathbf{y}}^{k_*,t_*,i}), \widetilde{\vartheta}(\xi_{\mathbf{x}}^{k_*,t_*,i}), \}_{i\in[B]} \\
&\cup \{\nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{x}}^{k_*,t_*}), \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}, \xi_{\mathbf{y}}^{k_*,t_*})\}, \\
G_+ = &\{h(\xi_{\mathbf{y}}^{k,t,i}), H(\xi_{\mathbf{x}}^{k,t,i}), \}_{(k,t)\in[(k_*,t_*),(K,N)],i\in[B]} \cup \{\widetilde{\theta}(\xi_{\mathbf{y}}^{k,t,i}), \widetilde{\vartheta}(\xi_{\mathbf{x}}^{k,t,i}), \}_{(k,t)\in[(k_*,t_*),(K,N)],i\in[B]} \\
&\cup \{\nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{x}}^{k,t}), \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}, \xi_{\mathbf{y}}^{k,t})\}(k,t) \in [(k_*,t_*),(K,N)]],
\end{aligned}$$

Then we have

$$\begin{aligned}
&\Pr\left[[(G_t, G_d)(\mathcal{D})]_{G_+} = [G]_{G_+} \middle| [(G_t, G_d)(\mathcal{D})]_{G_-\cup G_*} = [G]_{G_-\cup G_*}\right] \\
= \ &\Pr\left[[(G_t, G_d)(\mathcal{D}')]_{G_+} = [G]_{G_+} \middle| [(G_t, G_d)(\mathcal{D}')]_{G_-\cup G_*} = [G]_{G_-\cup G_*}\right].
\end{aligned}$$

So we conclude that

$$\frac{\Pr[(G_t, G_d)(\mathcal{D}) = G]}{\Pr[(G_t, G_d)(\mathcal{D}') = G]} = \frac{\Pr\left[[(G_t, G_d)(\mathcal{D})]_{G_*} = [G]_{G_*} \middle| [(G_t, G_d)(\mathcal{D})]_{G_-} = [G]_{G_-}\right]}{\Pr\left[[(G_t, G_d)(\mathcal{D}')]_{G_*} = [G]_{G_*} \middle| [(G_t, G_d)(\mathcal{D}')]_{G_-} = [G]_{G_-}\right]}$$

The RHS is concerning the ratio of the probability that the transcript and model updates in the $(k_*, t_*)$-th iteration are the same. In the following, we let $m' = \{2, \cdots, l\}$ denote the passive parties, and $m = 1$ as the active party (i.e., the server). Note that, in Algorithm 2, we add noise to the last layer of the whole mode, i.e., to $\theta$ and $\vartheta$. Moreover, generating noises for parties $m'$ and $m = 1$ is completed by once call of Algorithm 2. Given $G$, let $\{\widetilde{\theta}(\xi_{\mathbf{y}}^{k_*,t_*,i_*}), \widetilde{\vartheta}(\xi_{\mathbf{x}}^{k_*,t_*,i_*})\}$ be generated by Algorithm 2 with label $b$ and $\mathcal{D}$, and let $\{\widetilde{\theta}'(\xi_{\mathbf{y}}^{k_*,t_*,i_*}), \widetilde{\vartheta}'(\xi_{\mathbf{x}}^{k_*,t_*,i_*})\}$ be generated by Algorithm 2 with label $1 - b$ and $\mathcal{D}'$, then they are also $(\epsilon, 0)$-DP. Thus, combing the property of $(\epsilon, 0)$-DP and

that all the training samples except for the $k_*, t_*, i^*$-th one is the same in $\mathcal{D}$ and $\mathcal{D}'$, we have

$$\frac{\Pr\left[[(G_t, G_d)(\mathcal{D})]_{G_*} = [G]_{G_*} \middle| [(G_t, G_d)(\mathcal{D})]_{G_-} = [s]_{G_-}\right]}{\Pr\left[[(G_t, G_d)(\mathcal{D}')]_{G_*} = [G]_{G_*} \middle| [(G_t, G_d)(\mathcal{D}')]_{G_-} = [s]_{G_-}\right]} \leq e^{\epsilon}, \tag{86}$$

For any subset $S$ of possible assignments of $(G_t, G_d)$, we have

$$\begin{aligned}\Pr[(G_t, G_d)(\mathcal{D}) \in \mathcal{G}] &= \int_{G\in\mathcal{G}} \Pr[(G_t, G_d)(\mathcal{D}) = G] dG \\ &\leq \int_{G\in\mathcal{G}} e^{\epsilon} \Pr[(G_t, G_d)(\mathcal{D}') = G] dG \\ &= e^{\epsilon} \Pr[(G_t, G_d)(\mathcal{D}') \in \mathcal{G}],\end{aligned}$$

where the second step uses Eq. 86. Hence $(G_t, G_d)$ is $(\epsilon, 0)$-DP. In the ZO approximating process, we let $\hat{G}_t := \{\hat{h}(\xi_{\mathbf{y}}^{k,t,i,j})\}_{k\in[K],t\in[N],i\in[B],j\in[Q]} \cup \{\widetilde{\theta}(\xi_{\mathbf{y}}^{k,t,i,j}), \widetilde{\vartheta}(\xi_{\mathbf{x}}^{k,N+1,i}), \}_{k\in[K],t\in[N],i\in[B],j\in[Q]}$, and $\hat{G}_d := \{\widehat{\nabla}_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}; \xi_{\mathbf{x}}^{k,N+1,i}), \widehat{\nabla}_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}, \xi_{\mathbf{y}}^{k,t,i,j})\}_{k\in[K],t\in[N],i\in[B],j\in[Q]}$. Following above analysis, we have that $(\hat{G}_t, \hat{G}_d)$ is $(\epsilon, 0)$-differentially private. As a consequence, BAMBI-DP is $(\epsilon, 0)$ label DP which completes the proof. For the multi-class task, the analysis is similar.

### D.3 Computation Complexity Analysis of BAMBI and BAMBI-DP

In Algorithm 1, the computation complexity (CC) of steps 7 is $\mathcal{O}(B(Q+1)(p_m + q_m))$, that of steps 13 and 14 is $\mathcal{O}(q_m(Q+1))$, that of step 16 is $\mathcal{O}(Qp_m)$, that of steps 18 is $\mathcal{O}(B(p_m + q_m))$, and that of steps 24 and 25 is $\mathcal{O}(p_m + q_m)$. Especially, the CC of ZO estimation is $\mathcal{O}(BQ(p_m + q_m))$. Thus, the total complexity for all parties to perform an upper-level iteration (includes the $N$-step lower-level optimization process) in Algorithm 2 is $\mathcal{O}\left(NBQ(p+q)\right)$. The CC of computing the second-order derivative in existing BO methods Chen et al. (2021); Yang et al. (2021); Ji & Liang (2021) is $\mathcal{O}(Bpq)$. Thus, comparing to existing methods using the second-order derivatives to approximate the Jacobian matrix, BAMBI reduces the CC from $\mathcal{O}(pq)$ to $\mathcal{O}(BQ(p+q))$. For practical choice of $Q$ (typical choice is $Q \leq 10$) and practical VFBO problems with large $p, q$, such reduction is significant. The analysis of BAMBI-DP is similar.

# E EXPERIMENT DETAILS

In this section, we provide detailed content of the experiments and additional experimental results on FashionMnist and News20 datasets.

## E.1 EXPERIMENT ENVIRONMENT

We conducted our simulations on a deep learning workstation 48 cores. Codes are written using Python 3.6 and Pytorch 1.6. The distributed learning environment in Pytorch is used to simulate the federated learning.

## E.2 VFHRL PROBLEMS ON MNIST DATASET

**Experimental Details:** We use the standard MNIST dataset for learning. We random choose 50k samples as the training dataset, 10k samples for valuation, and 10k samples as the test dataset. In this experiment, for $\forall m \in [l]$, $\mathbf{x}_m$ is a 4-layers fully connected network (FCN) with a nonlinear activation function and $\mathbf{y}_m$ is a linear classifier. The stepsizes are set to $\alpha_k = 0.01$, $\beta_k = 0.01$.

**Experimental Results:** The experimental results are stated in the manuscript.

## E.3 EXPERIMENTS FOR VFHRL PROBLEMS ON FASHIONMNIST DATASET

**Experimental Details:** We use the standard FashionMNIST dataset for learning. The setting is set to the same as in standard MNIST dataset. We random choose 50k samples as the training dataset, 10k samples for valuation, and 10k samples as the test dataset. In this experiment, for $\forall m \in [l]$, $\mathbf{x}_m$ is a 4-layers fully connected network (FCN) with a nonlinear activation function and $\mathbf{y}_m$ is a linear classifier. The the stepsizes are set to $\alpha_k = 0.01$, $\beta_k = 0.01$.

**Evaluation of Comparable Performance:** The corresponding experimental results are shown in Fig. 4. According to the curves, the convergence performance of BAMBI (FedAlgo) is comparable to that of Non-FedAlgo, which supports our claim.

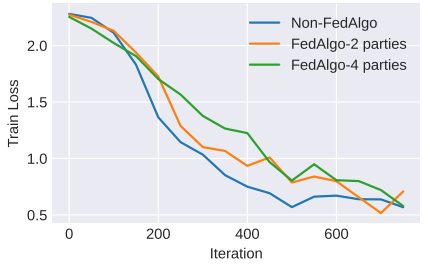


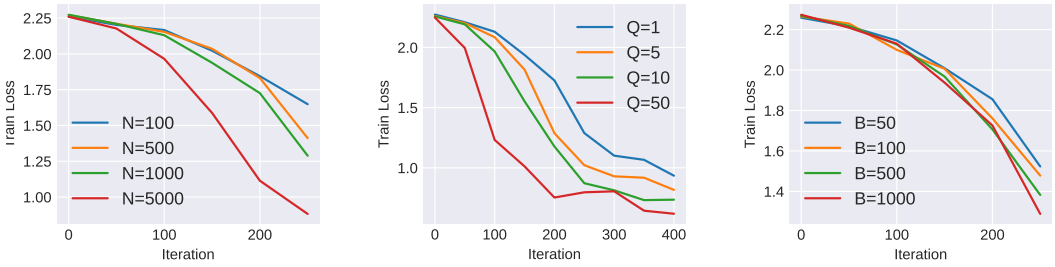

Figure 4: Performance comparison between FedAlgo and Non-FedAlgo on FashionMNIST dataset.

**Evaluation of Different Parameters:** Similar with statement in the manuscript, in the experiments, we set $N = 1000, Q = 1, B = 1000$ and vary one of them. The corresponding results in Fig. 5 show that although larger $N, Q, B$ is helpful for the faster convergence.

**Evaluation of Different Privacy Parameter:** We report the corresponding iteration number curves in Fig. 6 and test accuracy in Table 2. Compare the results of BAMBI-DP with those of Non-DP (i.e., BAMBI), we have that for this experiment, the value of $\epsilon$ has slight influence of the model performance, which may because that we choose a small but reasonable sensitive parameter.

Table 2: Comparisons between BAMBI and BAMBI-DP with different label DP levels on FashionMNIST dataset.

| | Non-DP | $\epsilon = 10$ | $\epsilon = 5$ | $\epsilon = 1$ |
|---|---|---|---|---|
| Test Accu | 85.52 | 85.21 | 85.31 | 84.62 |




(a) Evaluation of different $N$ (b) Evaluation of different $Q$ (c) Evaluation of different $B$

Figure 5: Evaluation of using different parameters on FashionMNIST dataset.

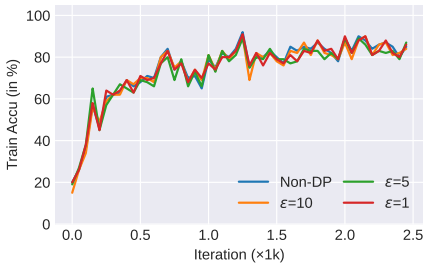



Figure 6: Convergence performance with different levels of DP guarantee on FashionMNIST dataset.

### E.4 Experiments for VFHO Problems on News20 Dataset

**Experimental Details:** In this experiment, we take samples in News20 dataset with labels related to "comp", "rec" and "misc" as the first class, and those with other labels as the second class. We random choose $40\%$ samples as the training dataset, $40\%$ for valuation, and the rest as the test dataset. In this experiment, for $\forall m \in [l]$, $\mathbf{x}_m$ is set to a linear regressor and $\mathbf{y}_m$ is set to a simple sum function to simulate the logistic regression. The the optimal stepsizes are $\alpha_k = 0.1$, $\beta_k = 10^{-5}$.

**Evaluation of Comparable Performance:** The corresponding experimental results are shown in Fig. 7. According to the results, the convergence performance of BAMBI (FedAlgo) is comparable to that of Non-FedAlgo, which supports our claim. It is also obvious that a larger $l$ will make the curves oscillation more serious but will not influence the convergence performance.

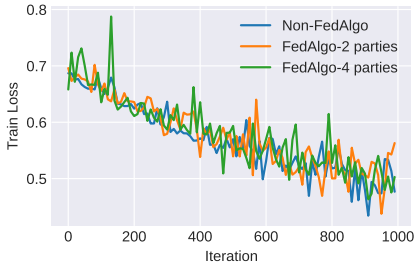
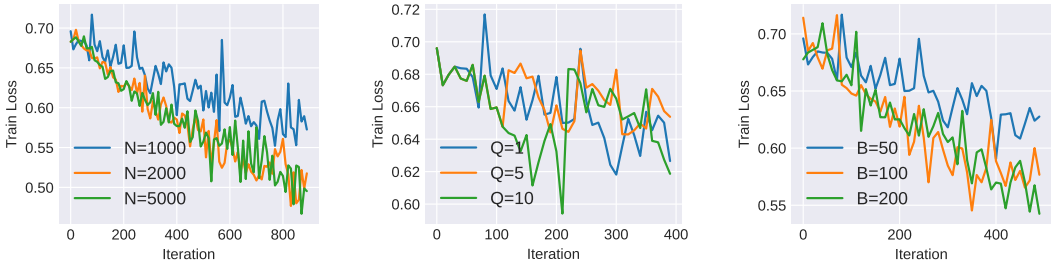



Figure 7: Performance comparison between FedAlgo and Non-FedAlgo on News20 dataset.

**Evaluation of Different Parameters:** In the experiments, we set $N = 1000, Q = 1, B = 1000$ and vary one of them. The corresponding results in Fig. 8 show that large $N, B$ is helpful for fast convergence, and varying the value of $Q$ has slight influence.

**Evaluation of Different Privacy Parameter:** To study the influence of $\epsilon$, we implement BAMBI-DP with $\epsilon = 1, 5, 10$, and report their train accuracy v.s. iteration number curves in Fig. 9. Compare the results of BAMBI-DP with those of Non-DP (i.e., BAMBI), we have that a better DP guarantee (i.e., a smaller $\epsilon$) leads to a slightly poorer model performance.




(a) Evaluation of different $N$ (b) Evaluation of different $Q$ (c) Evaluation of different $B$

Figure 8: Evaluation of using different parameters on News20 dataset.

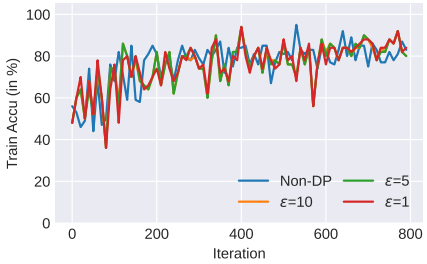



Figure 9: Convergence performance with different levels of DP guarantee on News20 dataset.