VAFL: a Method of Vertical Asynchronous Federated Learning

Tianyi Chen [1] **Xiao Jin** [1] **Yuejiao Sun** [2] **Wotao Yin** [3]

Abstract

Horizontal Federated learning (FL) handles multi-client data that share the same set of features, and vertical FL trains a better predictor that combine all the features from different clients. This paper targets solving vertical FL in an asynchronous fashion, and develops a simple FL method. The new method allows each client to run stochastic gradient algorithms without coordination with other clients, so it is suitable for intermittent connectivity of clients. This method further uses a new technique of *perturbed local embedding* to ensure data privacy and improve communication efficiency. Theoretically, we present the convergence rate and privacy level of our method for strongly convex, nonconvex and even nonsmooth objectives separately. Empirically, we apply our method to FL on various image and healthcare datasets. The results compare favorably to centralized and synchronous FL methods.

1. Introduction

Federated learning (FL) is an emerging machine learning framework where a central server and multiple clients (e.g., financial organizations) collaboratively train a machine learning model [19; 24; 4]. Compared with existing distributed learning paradigms, FL raises new challenges including the difficulty of synchronizing clients, the heterogeneity of data, and the privacy of both data and models.

Most of existing FL methods consider the scenario where each client has data of a different set of subjects but their data share many common features. Therefore, they can collaboratively learn a joint mapping from the feature space to the label space. This setting is also referred to data-partitioned or horizontal FL [20; 24].

Unlike the data-partitioned setting, in many learning scenarios, multiple clients handle data about the same set of subjects, but *each client has a unique set of features*. This case arises in e-commerce, financial, and healthcare applications [13]. For example, an e-commerce company may want to predict a customer's credit using her/his historical transactions from multiple financial institutions; and, a healthcare company wants to evaluate the health condition of a particular patient using his/her clinical data from various hospitals [36]. In these examples, data owners (e.g., financial institutions and hospitals) have different records of those users in their joint user base, so by combining their features, they can establish a more accurate model. We refer to this setting as feature-partitioned or vertical FL [42].

Compared to the relatively well-studied horizontal FL setting [25], the vertical FL setting has its unique features and challenges [15; 18]. In horizontal FL, the global model update at a server is an additive aggregation of the local models, which are updated by each client using its own data. In contrast, the global model in vertical FL is the concatenation of local models, which are coupled by the loss function, so updating a client's local model requires the information of the other clients' models. Stronger model dependence in the vertical setting leads to challenges on privacy protection and communication efficiency.

1.1. Prior art

We review prior work from the following three categories.

Federated learning. Since the seminal work [19; 24], there has been a large body of studies on FL in diverse settings. The most common FL setting is the horizontal setting, where a large set of data are partitioned among clients that share the same feature space [20]. To account for the personalization, multi-task FL has been studied in [34] that preserves the specialty of each client while also leveraging the similarity among clients, and horizontal FL with local representation learning has been empirically studied in [22]. Agnostic FL has also been proposed in [26], where the federated model is optimized for any target distribution formed by a mixture of the client distributions. Communication efficiency has been an important issue in FL. Popular methods generally aim to: c1) reduce the number of bits per communication

[1]Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA. [2]Department of Mathematics, University of California, Los Angeles, Los Angeles, CA, USA. [3]DAMO Academy, Alibaba US, Seattle, WA, USA. Authors are listed in alphabetical order. Correspondence to: Wotao Yin <wotao.yin@alibaba-inc.com>.

Proceedings of the ICML Workshop on Federated Learning for User Privacy and Data Confidentiality, July, 2020.

round, including [33; 3; 35; 2], to list a few; and, c2) save the number of communication rounds [6; 38; 23; 7].

Privacy-preserving learning. More recently, feature-partitioned vertical FL has gained popularity in the financial and healthcare applications [13; 42; 28; 17]. Different from the aggregated gradients in the horizontal case, the local gradients in the vertical FL may involve raw data of those features owned by other clients, which raises additional concerns on privacy. Data privacy has been an important topic since decades ago [43]. But early approaches typically require expensive communication and signaling overhead when they are applied to the FL settings. Recently, the notion of differential privacy becomes popular because i) it is a quantifiable measure of privacy [11; 1; 8]; and, ii) many existing learning algorithms can achieve differential privacy via simple modifications. In the context of learning from multiple clients, it has been studied in [4; 12]. But all these approaches are not designed for the vertical FL models and the flexible client update protocols.

Asynchronous and parallel optimization. Regarding methodology, asynchronous and parallel optimization methods are often used to solve problems with asynchrony and delays, e.g., [31]. For the feature-partitioned vertical FL setting in this paper, it is particularly related to the Block Coordinate Descent (BCD) method [40; 30]. The asynchronous BCD and its stochastic variant have been developed under the condition of bounded delay in [29; 21; 5]. The Recent advances in this direction established convergence under unbounded delay with blockwise or stochastic update [37; 10]. However, all the aforementioned works consider the shared memory structure so that each computing node can access the entire dataset, which significantly alleviates the negative effect of asynchrony and delays. Moreover, the state-of-the-art asynchronous methods cannot guarantee i) the convergence when the loss is nonsmooth, and, ii) the privacy of the local update which is at the epicenter of FL.

1.2. This work

The present paper puts forth an optimization method for vertical FL, which is featured by three main components.

1. A *general optimization formulation* for vertical FL that consists of a global model and one local embedding model for each client. The local embedding model can be linear or nonlinear, or even nonsmooth. It maps raw data to compact features and, thus, reduces the number of parameters that need to be communicated to and from the global model.

2. Flexible *federated learning algorithms* that allow intermittent or even strategic client participation, uncoordinated training data selections, and data protection by differential-privacy based methods (for specific loss functions, one can instead apply multiple-party secure computing protocols).

3. *Rigorous convergence analysis* that establishes the performance lower bound and the privacy level.

We have also numerically validated our vertical FL algorithms and their analyses on federated logistic regression and deep learning. Tests on image and medical datasets demonstrate the competitive performance of our algorithms relative to centralized and synchronous FL algorithms.

2. Vertical federated learning

This section introduces the formulation of vertical FL.

2.1. Problem statement

Consider a set of M clients: $\mathcal{M} := \{1, \ldots, M\}$. A dataset of N samples, $\{\mathbf{x}_n, y_n\}_{n=1}^N$, are maintained by M local clients. Each client m is also associated with a unique set of features. For example, client m maintains feature $x_{n,m} \in \mathbb{R}^{p_m}$ for $n = 1, \ldots, N$, where $x_{n,m}$ is the m-th block of n-th sample vector $\mathbf{x}_n := [x_{n,1}^\top, \cdots, x_{n,M}^\top]^\top$ at client m. Suppose the n-th label y_n is stored at the server.

To preserve the privacy of data, the client data $x_{n,m} \in \mathbb{R}^{p_m}$ are not shared with other clients as well as the server. Instead, each client m learns a local (linear or nonlinear) embedding h_m parameterized by θ_m that maps the high-dimensional vector $x_{n,m} \in \mathbb{R}^{p_m}$ to a low-dimensional one $h_{n,m} := h_m(\theta_m; x_{n,m}) \in \mathbb{R}^{\underline{p_m}}$ with $\underline{p_m} \ll p_m$. Ideally, the clients and the server want to solve

$$F(\theta_0, \boldsymbol{\theta}) := \frac{1}{N} \sum_{n=1}^{N} \ell\left(\theta_0, h_{n,1}, \ldots, h_{n,M}; y_n\right) + \sum_{m=1}^{M} r(\theta_m)$$
$$\text{with} \quad h_{n,m} := h_m(\theta_m; x_{n,m}), \quad m = 1, \cdots, M \tag{1}$$

where θ_0 is the global model parameter kept at and learned by the server, and $\boldsymbol{\theta} := [\theta_1^\top, \cdots, \theta_M^\top]^\top$ concatenates the local models kept at and learned by local clients, ℓ is the loss capturing the accuracy of the global model parameters $\theta_0, \theta_1, \ldots, \theta_M$, and r is the per-client regularizer that confines the complexity of or encodes the prior knowledge about the local model parameters.

For problem (1), the local information of client m is fully captured in the embedding vector $h_{n,m}$, $\forall n = 1, \cdots, N$. Hence, the quantities that will be exchanged between server and clients are $\{h_{n,m}\}$ and the gradients of $\ell(\theta_0, h_{n,1}, \ldots, h_{n,M}; y_n)$ with respect to (w.r.t.) $\{h_{n,m}\}$. See a diagram for VAFL implementation in Figure 1.

2.2. Asynchronous client updates

For FL, we consider solving (1) without coordination among clients. Asynchronous optimization methods have been used to solve such problems. However, state-of-the-art methods cannot guarantee i) the convergence when the mapping $h_{n,m}$

Algorithm 1 Vertical asynchronous federated learning

1: **initialize:** θ_0, $\{\theta_m\}$, datum index n, client index m
2: **while** not convergent **do**
3: **when** a **Client** m is activated, **do**:
4: †select private datum (or data mini-batch) $x_{n,m}$
5: †**upload** secure information $h_{n,m}=h_m(\theta_m;x_{n,m})$
6: **query** $\nabla_{h_{n,m}}\ell(\theta_0,h_{n,1},\ldots,h_{n,M};y_n)$ from Server
7: update local model θ_m
8: **when Server** receives $h_{n,m}$ from Client m, **do**:
9: compute $\nabla_{\theta_0}\ell(\theta_0,h_{n,1},\ldots,h_{n,M};y_n)$
10: update server's local model θ_0
11: **when Server** receives a query from Client m, **do**:
12: compute $\nabla_{h_{n,m}}\ell(\theta_0,h_{n,1},\ldots,h_{n,M};y_n)$
13: **send** it to Client m
14: **end while**

†We can let Step 5 also send $h_{n,m}$ for those n *not* selected in Step 4. We can re-order Steps 4–7 as 6, 7, *then* 4, and 5. They reduce information delay, yet analysis is unchanged.

Algorithm 2 Vertical t-synchronous federated learning

1: **Initialize:** θ_0, $\{\theta_m\}$, datum index n, client index m, integer $1\le t\le M$
2: **while** not convergent **do**
Algorithm 1, Lines 3–7
8: **when Server** receives $h_{n,m}$'s from t Clients, **do**:
Algorithm 1, Lines 9 and 10
11: **when Server** receives queries from t Clients, **do**:
Algorithm 1, Lines 12 and 13 for each of t clients
14: **end while**

is nonlinear (thus the loss is nonsmooth), and, ii) the privacy of the update which is at the epicenter of the FL paradigm.

We first describe our vertical asynchronous federated learning (VAFL) algorithm in a high level as follows. During the learning process, from the server side, it waits until receiving a message from an active client m, which is either
i) **a query** of the loss function's gradient w.r.t. to the embedding vector $h_{n,m}$; or,
ii) **a new embedding vector** $h_{n,m}$ calculated using the updated local model parameter θ_m.

To response to the query i), the server calculates the gradient for client m using its current $\{h_{n,m}\}$, and sends it to the client; and, upon receiving ii), the server computes the new gradient w.r.t. θ_0 using the embedding vectors it currently has from other clients and updates its model θ_0.

For each interaction with server, each *active* client m randomly selects a datum $x_{n,m}$, queries the corresponding gradient w.r.t. $h_{n,m}$ from server, and then it securely uploads the updated embedding vector $h_{n,m}$, and then updates the local model θ_m. The mechanism that ensures secure uploading will be described in Section 4. Without introducing cumbersome iteration, client, and sample indexes, we summarize the asynchronous client updates in Algorithm 1.

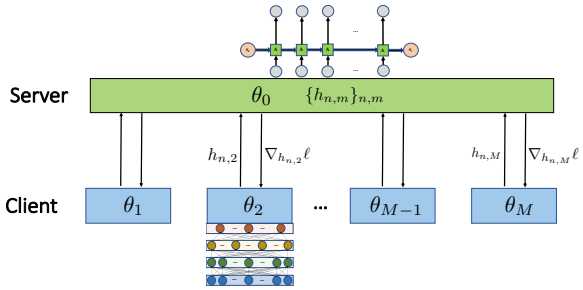

Figure 1. A diagram for VAFL. The local model at client m is denoted as θ_m which generates the local embedding $h_{n,m}$.

Specifically, since clients update the model without external coordination, we thereafter use k to denote the global counter (or iteration), which increases by one whenever i) the server receives the new embedding vector $h_{n,m}$ from a client, calculates the gradient, and updates the server model θ_0; and, ii) the corresponding client m obtains the gradient w.r.t. $h_{n,m}$, and updates the local model θ_m. Accordingly, we let m_k denote the client index that uploads at iteration k, and n_k denote the sample index used at iteration k.

For notation brevity, we use a single datum n_k for each uncoordinated update in the subsequent algorithms, but the algorithm and its analysis can be easily generalized to a minibatch of data $\mathcal{N}_k$. Let $\hat{g}_0^k$ denote the stochastic gradients of the loss at n_k-th sample w.r.t. server model θ_0 as

$$\hat{g}_0^k := \nabla_{\theta_0}\ell\big(\theta_0^k, h_{n_k,1}^{k-\tau_{n_k,1}^k},\ldots,h_{n_k,M}^{k-\tau_{n_k,M}^k};y_{n_k}\big) \tag{2a}$$

and the gradients w.r.t. the local model θ_m as

$$\begin{aligned}\hat{g}_m^k :=& \nabla_{\theta_m}\ell\big(\theta_0^k, h_{n_k,1}^{k-\tau_{n_k,1}^k},\ldots,h_{n_k,M}^{k-\tau_{n_k,M}^k};y_{n_k}\big) \\ =& \nabla_{\theta_m}h_{n_k,m}^k\nabla_{h_{n_k,m}}\ell\big(\theta_0^k, h_{n_k,1}^{k-\tau_{n_k,1}^k},\ldots,h_{n_k,M}^{k-\tau_{n_k,M}^k};y_{n_k}\big).\end{aligned} \tag{2b}$$

The delay for client m and sample n will increase via

$$\tau_{n,m}^{k+1} = \begin{cases} 1, & m=m^k,\ n=n^k, \\ \tau_{n,m}^k+1, & \text{otherwise.}\end{cases} \tag{3}$$

With the above short-hand notation, at iteration k, the update at the **server side** is $\theta_0^{k+1}=\theta_0^k-\eta_0^k\hat{g}_0^k$. For the **active local client** m_k at iteration k, its update is

$$\theta_{m_k}^{k+1}=\theta_{m_k}^k-\eta_{m_k}^k\hat{g}_{m_k}^k-\eta_{m_k}^k\nabla r(\theta_{m_k}^k), \tag{4a}$$

and for the **other clients** $m\neq m_k$, the update is

$$\theta_m^{k+1}=\theta_m^k, \tag{4b}$$

where η_m^k is the stepsize and m_k is the index of the client responsible for the kth update.

2.3. Types of flexible update rules

As shown in (2), the stochastic gradients are evaluated using delayed local embedding information $h_m^{k-\tau^k_{n_k,m}}$ from each client m, where $\tau^k_{n_k,m}$ is caused by both asynchronous communication and stochastic sampling.

To ensure convergence, we consider two reasonable settings on the flexible update protocols:
1. *Uniformly bounded delay* D. We can realize this by modifying the server behavior. During the training process, whenever the delay of $\tau^k_{n_k,m}$ exceeds $D(>0)$, the server immediately queries fresh $h_{n,m}$ from client m before continuing the server update process.
2. *Stochastic unbounded delay.* In this case, the activation of each client is a stochastic process. The delays is determined by the hitting times of the stochastic processes. For example, if the activation of all the clients follows independent Poisson processes, the delays will be geometrically distributed.
3. *t-synchronous update, $t>0$.* While fully asynchronous update is most flexible, t-synchronous update is also commonly adopted. In this case, the server computes the gradient w.r.t. θ_0 until receiving $\{h_{n,m}\}$ from t different clients, and then updates the server's model using the newly computed gradient. The t-synchronous updates have more stable performance empirically, which is listed in Algorithm 2.

3. Convergence analysis

We present the convergence results of our VAFL method for the nonconvex and strongly convex cases and under different update rules. Due to space limitation, this section mainly presents the convergence rates for fully asynchronous version of VAFL (Algorithm 1), and the convergence results for t-synchronous one (Algorithm 2) are similar, and thus will be given in the supplementary materials.

To analyze the performance of Algorithm 1, we first make the following assumptions on sampling and smoothness.

Assumption 1. *Sample indexes $\{n_k\}$ are i.i.d. And the variance of gradient follows $\mathbb{E}\left[\|g_m^k - \nabla_{\theta_m} F(\theta_0^k, \boldsymbol{\theta}^k]\|^2\right] \leq \sigma_m^2$, $\forall m$, where g_m^k is the stochastic gradient $\hat{g}_m^k$ without delay, e.g., $g_m^k := \nabla_{\theta_m}\ell\left(\theta_0^k, h^k_{n_k,1}, \ldots, h^k_{n_k,M}; y_{n_k}\right)$.*

Assumption 2. *The optimal loss is lower bounded $F^* > -\infty$. The gradient ∇F is L-Lipschitz continuous, and $\nabla_{\theta_m} F$ is L_m-Lipschitz continuous.*

Generally, assumption 2 cannot be satisfied under our general vertical FL formulation with *nonsmooth* local embedding functions such as neural networks. However, techniques that we call perturbed local embedding will be introduced to enforce smoothness in Section 4.

To handle asynchrony, we need the following assumption, which is often seen in the analysis of asynchronous BCD.

Assumption 3. *The uploading client m_k is independent of $m_0, \ldots, m_{k-1}$ and satisfies $\mathbb{P}(m_k = m) := q_m$.*

A simple scenario satisfying this assumptions is that the activation of all clients follows independent Poisson processes. That is, if the time difference between two consecutive activations of client m follows exponential distribution with parameter λ_m, then the activation of client m is a Possion process with $q_m = \lambda_m^{-1}/\sum_{j=1}^M \lambda_j^{-1}$.

We first present the convergence results for bounded $\tau^k_{n_k,m}$.

3.1. Convergence under bounded delay

We make the following assumption *only* for this subsection.

Assumption 4 (Uniformly bounded delay)**.** *For each client m and each sample n, the delay $\tau^k_{n,m}$ at iteration k is bounded by a constant D, i.e., $\tau^k_{n,m} \leq D$.*

We first present the convergence for the nonconvex case.

Theorem 1. *Under Assumptions 1 – 4, if $\eta_0^k = \eta_m^k = \min\{\frac{1}{4(1+D)L}, \frac{c_\eta}{\sqrt{K}}\}$ with $c_\eta > 0$, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] = \mathcal{O}\left(1/\sqrt{K}\right). \tag{5}$$

Under the additional assumption of strong convexity, the convergence rate is improved.

Theorem 2. *Under Assumptions 1 – 4, and the additional assumption that F is μ-strongly convex in $(\theta_0, \boldsymbol{\theta})$, if $\eta^k = \frac{4}{\mu \min_m \sqrt{q_m}(k+K_0)}$ with the constant $K_0 > 0$, then*

$$\mathbb{E}F\left(\theta_0^K, \boldsymbol{\theta}^K\right) - F^* = \mathcal{O}\left(1/K\right). \tag{6}$$

3.2. Convergence under stochastic unbounded delay

We make the following assumption *only* for this subsection.

Assumption 5 (Stochastic unbounded delay)**.** *For each client m, the delay $\tau^k_{n_k,m}$ is an random variable with unbounded support. And there exists $\bar{p}_m, \rho > 0$ such that $\mathbb{P}\left(\tau^k_{n_k,m} = d\right) \leq \bar{p}_m \rho^d := p_{m,d}$.*

Under Assumption 5, we obtain the convergence rates of the same order as those the under bounded delay assumption.

Theorem 3. *Under Assumptions 1-3 and 5, if $\eta_0^k = \eta_m^k = \min\left\{\frac{1}{4(1+\min_m \sqrt{c_m})L}, \frac{c_\eta}{\sqrt{K}}\right\}$, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] = \mathcal{O}\left(1/\sqrt{K}\right). \tag{7}$$

Under the additional assumption of strong convexity, the convergence rate is improved.

Theorem 4. *Assume that F is μ-strongly convex in $(\theta_0, \boldsymbol{\theta})$. Then under Assumptions 1-3 and 5, if $\eta_0^k = \eta_m^k = \frac{2}{\nu(k+K_0)}$ where K_0 is a positive constant, then it follows that*

$$\mathbb{E}F(\theta_0^K, \boldsymbol{\theta}^K) - F^* = \mathcal{O}(1/K). \tag{8}$$

It worth mentioning that under the assumption of bounded delay and unbounded but stochastic delay, without even coordinating clients' gradient samples and local model updates, our algorithm achieves the same order of convergence as that of block-wise SGD in both cases [41].

4. Perturbed local embedding: Enforcing differential privacy and smoothness

In this section, we introduce a local perturbation technique that is applied by each client to enforce the differential privacy of local information, which also smoothes the otherwise nonsmooth mapping of local embedding.

4.1. Local perturbation

Recall that h_m denotes a local embedding function of client m with the parameter θ_m which embeds the information of local data $x_{n,m}$ into its outputs $h_{n,m} := h_m(\theta_m; x_{n,m})$. When h_m is linear embedding, it is as simple as $h_m(\theta_m; x_{n,m}) = x_{n,m}^\top \theta_m$. To further account for nonlinear embedding such as neural networks, we represent $h_{n,m}$ in the following composite form

$$u_0 = x_{n,m} \tag{9a}$$
$$u_l = \sigma_l(w_l u_{l-1} + b_l), \quad l = 1, \dots, L \tag{9b}$$
$$h_{n,m} = u_L \tag{9c}$$

where σ_l is a linear or nonlinear function, and w_l, b_l corresponds to the parameter θ_m of h_m, e.g., $\theta_m := [w_1, \cdots, w_L, b_1, \cdots, b_L]^\top$. Here we assume that σ_l is $L_{\sigma_l}^0$-Lipschitz continuous. Specially, when h_m is linear, the composite embedding (9) corresponds to $L = 1, \sigma_1(z) = z$.

We perturb the local embedding function h_m by adding a random neuron with output Z_l at each layer l (cf. (9b))

$$u_l = \sigma_l(w_l u_{l-1} + b_l + Z_l), \quad l = 1, \dots, L \tag{10}$$

where $Z_1, \dots, Z_L$ are independent random variables. With properly chosen distributions of $Z_l, l = 1, \dots, L$, we show below h_m is smooth and enables differential privacy. While it does not exclude other options, our choice of the perturbation distributions is

$$Z_L \sim \mathcal{N}(0, c^2) \tag{11a}$$
$$Z_l \sim \mathcal{U}[-\sqrt{3}c_l, \sqrt{3}c_l], \quad l = 1, \cdots, L-1 \tag{11b}$$

where $\mathcal{N}(0, c^2)$ denotes the Gaussian distribution with zero mean and variance c^2, and $\mathcal{U}[-\sqrt{3}c_l, \sqrt{3}c_l]$ denotes the uniform distribution over $[-\sqrt{3}c_l, \sqrt{3}c_l]$.

4.2. Enforcing smoothness

The convergence results in Section 3 hold under Assumption 2 which requires the smoothness of the overall loss function. Inspired by the randomized smoothing technique [9; 27], we are able to smooth the objective function by taking expectation with respect to the random neurons. Intuitively this follows the fact that the smoothness of a function can be increased by convolving with proper distributions. By adding random neuron Z_l, the landscape of σ_l will be smoothed in expectation with respect to Z_l. And by induction, we can show the smoothness of local embedding vector h_m. Then so long as the loss function ℓ is smooth w.r.t. the local embedding vector h_m, the global objective F is smooth by taking expectation with respect to all the random neurons.

We formally establish this result in the following theorem.

Theorem 5. *For each embedding function h_m, if the activation functions follow $\sigma_l = \sigma, \forall l$, and the variances of the random neurons follow (11), and assume $\|w_l\|$ is bounded, then with $\mathbf{Z} := [Z_1^\top, \cdots, Z_L^\top]^\top$, the perturbed loss satisfies Assumption 2, which is given by*

$$F_c(\theta_0, \boldsymbol{\theta}) := \mathbb{E}_{\mathbf{Z}}[F(\theta_0, \boldsymbol{\theta}; \mathbf{Z})]. \tag{12}$$

Starting from $L_{b_L}^h = L_\sigma^0 d/c$, the smoothness constants of the local model θ_m denoted as $L_{\theta_m}^{F_c}$ satisfy the following recursion ($l = 1, \cdots, L-1$)

$$\begin{aligned} L_{b_l}^h &= L_{b_{l+1}}^h \|w_{l+1}\| (L_\sigma^0)^2 + L_\sigma^0 \|w_L\| \cdots L_\sigma^0 \|w_{l+1}\| L_{\bar{\sigma}}(c_l) \\ L_{w_l}^h &= \mathbb{E}[\|u_{l-1}\|] L_{b_l}^h \\ L_{\theta_m}^{F_c} &= L_{h_m}^\ell (L_{h_m}^0)^2 + L_\ell^0 \sum_{l=1}^{L} (L_{w_l}^h + L_{b_l}^h) + L_{\theta_m}^r \end{aligned} \tag{13}$$

where $L_{\theta_m}^r$ is the smoothness constant of the regularizer w.r.t. θ_m; $L_{b_l}^h$ and $L_{w_l}^h$ are the smoothness constants of the perturbed local embedding h w.r.t. the bias b_l and weight w_l; and $L_{\bar{\sigma}}(c_l) := 2\sqrt{d} L_\sigma^0 / c_l$ is the smoothness constant of the neuron at lth layer under the uniform perturbation.

Theorem 5 implies that the perturbed loss is smooth w.r.t. the local model θ_m, and a large perturbation (large c_l or c) will lead to a smaller smoothness constant.

4.3. Enforcing differential privacy

We now connect the perturbed local embedding technique with the private information exchange in Algorithms 1-2.

As local clients keep sending out embedded information, it is essential to prevent any attacker to trace back to a specific individual via this observation. Targeting a better trade-off between the privacy and the accuracy, we leverage the Gaussian differential privacy (GDP) developed in [8].

Definition 1 ([8]). *A mechanism $\mathcal{M}$ is said to satisfy μ-GDP if for all neighboring datasets S and S', we have*

$$T(\mathcal{M}(S), \mathcal{M}(S')) \geq T(\mathcal{N}(0,1), \mathcal{N}(\mu, 1)) \tag{14}$$

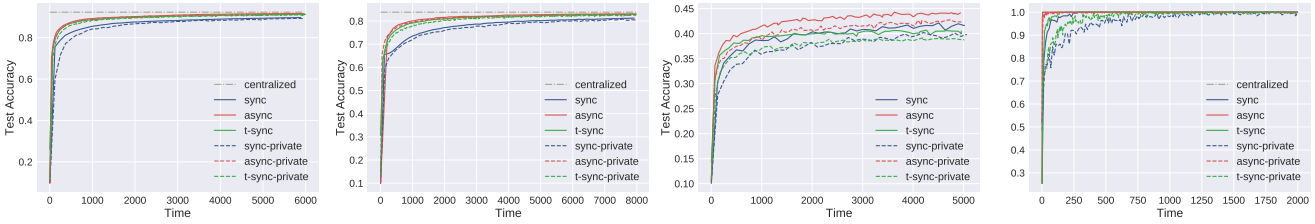

Figure 2. Testing accuracy versus clock time (sec) in MNIST, Fashion-MNIST, CIFAR10 and Parkinson datasets (from left to right).

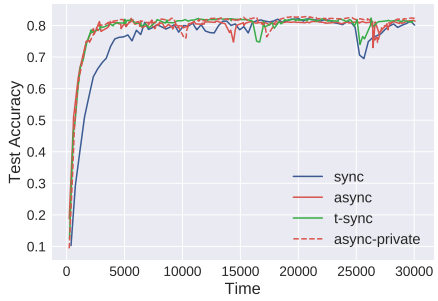

Figure 3. Testing accuracy of VAFL with nonlinear local embedding on *ModelNet40* dataset.

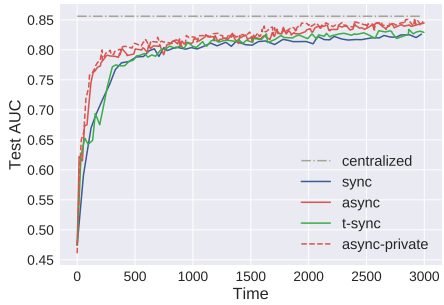

Figure 4. AUC curve of VAFL with local LSTM embedding on *MIMIC-III* clinical care dataset.

where the trade-off function $T(P,Q)(\alpha) = \inf\{\beta_\phi : \alpha_\phi \leq \alpha\}$*, and* α_ϕ, β_ϕ *are type I and II errors given a threshold* ϕ*.*

Intuitively, μ-GDP guarantees that distinguishing two adjacent datasets via information revealed by $\mathcal{M}$ is at least as difficult as distinguishing the two distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu,1)$. Smaller μ means less privacy loss.

To characterize the level of privacy of our local embedding approaches, we build on the moments accountant technique originally developed in [1] to establish that adding random neurons endows Algorithm 1 with GDP.

Theorem 6. *Under the same set of assumptions as those in Theorem 5, for client* m*, if we set the variance of the Gaussian random neuron at the* L*-th layer as*

$$c = \mathcal{O}\left(N_m\sqrt{K}/(\mu N)\right) \tag{15}$$

where N_m *is the size of minibatch used at client* m*,* N *is the size of the whole batch,* K *is the number of queries (i.e. the number of data samples processed by* h_m *at client* m*), then VAFL satisfies* μ*-GDP for the dataset of client* m*.*

Theorem 6 demonstrates the trade-off between accuracy and privacy. To increase privacy, i.e., decrease μ in (14), the variance of random neurons needs to be increased (cf. (15)). However, as the variance of random neurons increases, the variance of the stochastic gradient (2) also increases, which will in turn lead to slower convergence.

5. Numerical tests and remarks

We benchmark the fully asynchronous version of VAFL (**async**) in Algorithm 1, and t-synchronous version of VAFL (**t-sync**) in Algorithm 2 with the synchronous block-wise SGD (**sync**), which requires synchronization and sample index coordination among clients in each iteration. We also include private versions of these algorithms via perturbed local embedding technique in Section 4.

VAFL for federated logistic regression. We first conduct logistic regression on MNIST, Fashion-MNIST, CIFAR10 and Parkinson disease [32] datasets. The l_2-regularizer coefficient is set as 0.001. We select $M = 7$ for MNIST and MNIST, $M = 8$ for CIFAR10 and $M = 3$ for PD dataset. The testing accuracy versus wall-clock time is reported in Figures 2. The dashed horizontal lines represent the results trained on the centralized (non-federated) model, and the dashed curves represent private variants of considered algorithms with variance $c = 0.1$. In all cases, VAFL learns a federated model with accuracies comparable to that of the centralized model that requires collecting raw data.

VAFL for federated deep learning. We first train a neural network modified from MVCNN with 12-view data [39]. We use $M = 4$ clients, and each client has 3 views of each object and use a 7-layer CNN as local embedding functions, and server uses a fully connected network to aggregate the local embedding vectors. Results are plotted in Figure 3.

We also test our VAFL algorithm in MIMIC-III — an open dataset comprising deidentified health data [16]. We perform the in-hospital mortality prediction as in [14] among $M = 4$ clients. Each client uses LSTM as the embedding function. In Figure 4, we can still observe that async and t-sync VAFL learn a federated model with accuracies comparable to that of the centralized model, and requires less time relative to the synchronous FL algorithm.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, Vienna, Austria, October 2016.

[2] Aji, A. F. and Heafield, K. Sparse communication for distributed gradient descent. In *Proc. Conf. Empirical Methods Natural Language Process.*, pp. 440–445, Copenhagen, Denmark, Sep 2017.

[3] Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 1709–1720, Long Beach, CA, Dec 2017.

[4] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *Proc. ACM Conf. on Comp. and Comm. Security*, pp. 1175–1191, Dallas, TX, October 2017.

[5] Cannelli, L., Facchinei, F., Kungurtsev, V., and Scutari, G. Asynchronous parallel algorithms for nonconvex big-data optimization: Model and convergence. *arXiv preprint:1607.04818*, July 2016.

[6] Chen, T., Giannakis, G., Sun, T., and Yin, W. LAG: Lazily aggregated gradient for communication-efficient distributed learning. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 5050–5060, Montreal, Canada, Dec 2018.

[7] Chen, T., Sun, Y., and Yin, W. LASG: Lazily aggregated stochastic gradients for communication-efficient distributed learning. *arXiv preprint:2002.11360*, February 2020.

[8] Dong, J., Roth, A., and Su, W. J. Gaussian differential privacy. *arXiv preprint:1905.02383*, May 2019.

[9] Duchi, J. C., Bartlett, P. L., and Wainwright, M. J. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[10] Dutta, S., Joshi, G., Ghosh, S., Dube, P., and Nagpurkar, P. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. In *Proc. Intl. Conf. on Artif. Intell. and Stat.*, Lanzarote, Spain, 2018.

[11] Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[12] Hamm, J., Cao, Y., and Belkin, M. Learning privately from multiparty data. In *Proc. Intl. Conf. Machine Learn.*, pp. 555–563, New York, NY, June 2016.

[13] Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., and Thorne, B. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint:1711.10677*, November 2017.

[14] Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., and Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019. ISSN 2052-4463. doi: 10.1038/s41597-019-0103-9. URL https://doi.org/10.1038/s41597-019-0103-9.

[15] Hu, Y., Niu, D., Yang, J., and Zhou, S. Fdml: A collaborative machine learning framework for distributed features. In *Proc. of ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining*, pp. 2232–2240, 2019.

[16] Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3 (160035), 2016.

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

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

[19] Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint:1610.02527*, October 2016.

[20] Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint:1610.05492*, Oct 2016.

[21] Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 5330–5340, Long Beach, CA, Dec 2017.

[22] Liang, P. P., Liu, T., Ziyin, L., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint:2001.01523*, January 2020.

[23] Liu, Y., Kang, Y., Zhang, X., Li, L., Cheng, Y., Chen, T., Hong, M., and Yang, Q. A communication efficient vertical federated learning framework. *arXiv preprint arXiv:1912.11187*, 2019.

[24] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proc. Intl. Conf. Artificial Intell. and Stat.*, pp. 1273–1282, Fort Lauderdale, FL, April 2017.

[25] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proc. Intl. Conf. on Artif. Intell. and Stat.*, pp. 1273–1282, Fort Lauderdale, Florida, Apr 2017.

[26] Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *Proc. Intl. Conf. Machine Learn.*, pp. 4615–4625, Long Beach, CA, June 2019.

[27] Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[28] Niu, C., Wu, F., Tang, S., Hua, L., Jia, R., Lv, C., Wu, Z., and Chen, G. Secure federated submodel learning. *arXiv preprint:1911.02254*, November 2019.

[29] Peng, Z., Xu, Y., Yan, M., and Yin, W. Arock: an algorithmic framework for asynchronous parallel coordinate updates. *SIAM J. Sci. Comp.*, 38(5):2851–2879, September 2016.

[30] Razaviyayn, M., Hong, M., and Luo, Z.-Q. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, June 2013.

[31] Recht, B., Re, C., Wright, S., and Niu, F. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 693–701, Granada, Spain, December 2011.

[32] Sakar, C. O., Serbes, G., Gunduz, A., Tunc, H. C., Nizam, H., Sakar, B. E., Tutuncu, M., Aydin, T., Isenkul, M. E., and Apaydin, H. A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform. *Applied Soft Computing*, 74:255–263, 2019.

[33] Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Proc. Conf. Intl. Speech Comm. Assoc.*, Singapore, Sept 2014.

[34] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 4427–4437, Long Beach, CA, December 2017.

[35] Strom, N. Scalable distributed DNN training using commodity gpu cloud computing. In *Proc. Conf. Intl. Speech Comm. Assoc.*, Dresden, Germany, Sept 2015.

[36] Sun, C., Ippel, L., van Soest, J., Wouters, B., Malic, A., Adekunle, O., van den Berg, B., Mussmann, O., Koster, A., van der Kallen, C., et al. A privacy-preserving infrastructure for analyzing personal health data in a vertically partitioned scenario. *Studies in health technology and informatics*, 264:373–377, 2019.

[37] Sun, T., Hannah, R., and Yin, W. Asynchronous coordinate descent under more realistic assumptions. In *Proc. Advances in Neural Info. Process. Syst.*, pp. 6183–6191, Long Beach, CA, December 2017.

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

[39] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

[40] Xu, Y. and Yin, W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3): 1758–1789, 2013.

[41] Xu, Y. and Yin, W. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization*, 25(3):1686–1716, 2015.

[42] Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intelligent Systems and Technology*, 10(2), January 2019.

[43] Yao, A. C. Protocols for secure computations. In *Annual Symposium on Foundations of Computer Science*, pp. 160–164, Chicago, Illinois, 1982.

Supplementary materials for "VAFL: a Method of Vertical Asynchronous Federated Learning"

In this supplementary document, we first present some supporting lemmas that will be used frequently in this document, and then present the proofs of all the lemmas and theorems in the paper, which is followed by details on our experiments. The content of this supplementary document is summarized as follows.

Table of Contents

A. Supporting Lemmas

For notational brevity, we define

$$G_0^k := \nabla_{\theta_0} F(\theta_0^k, \boldsymbol{\theta}^k) \tag{16a}$$

$$G_m^k := \nabla_{\theta_m} F(\theta_0^k, \boldsymbol{\theta}^k) \tag{16b}$$

$$g_0^k := \nabla_{\theta_0} \ell\left(\theta_0^k, h_{n_k,1}^k, \ldots, h_{n_k,M}^k; y_{n_k}\right) \tag{16c}$$

$$g_m^k := \nabla_{\theta_m} \ell\left(\theta_0^k, h_{n_k,1}^k, \ldots, h_{n_k,M}^k; y_{n_k}\right) \tag{16d}$$

$$\hat{g}_0^k := \nabla_{\theta_0} \ell(\theta_0, h_{n_k,1}^{k-\tau_{n_k,1}^k}, \ldots, h_{n_k,M}^{k-\tau_{n_k,M}^k}; y_{n_k}) \tag{16e}$$

$$\hat{g}_m^k := \begin{cases} \nabla h_m(\theta_m^k; x_{n_k,m}) \nabla_{h_m} \ell(\theta_0^k, h_{n_k,1}^{k-\tau_{n_k,1}^k} \ldots, h_{n_k,M}^{k-\tau_{n_k,M}^k}; y_{n_k}) & \text{if } m = m_k \\ 0 & \text{else.} \end{cases} \tag{16f}$$

$$\hat{\boldsymbol{\theta}} := [\theta_1^{k-\tau_{n_k,1}^k}; \ldots; \theta_M^{k-\tau_{n_k,M}^k}]. \tag{16g}$$

To handle the delayed information, we leverage the following Lyapunov function for analyzing VAFL

$$V^k := F(\theta_0^k, \boldsymbol{\theta}^k) + \sum_{d=1}^{D} \gamma_d \|\boldsymbol{\theta}^{k-d+1} - \boldsymbol{\theta}^{k-d}\|^2 \tag{17}$$

where $\{\gamma_d\}$ are a set of constants to be determined later.

Lemma 1. *Under Assumptions 1 – 4, for $\eta_0^k \le \frac{1}{4L}, \eta_m^k \le \frac{1}{4(L+2\gamma_1)}$, it follows that (with $\gamma_{D+1} = 0$)*

$$\begin{aligned}
\mathbb{E}V^{k+1} - \mathbb{E}V^k \le & -\left(\frac{\eta_0^k}{2} - L(\eta_0^k)^2\right)\mathbb{E}[\|\nabla_{\theta_0} F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] \\
& -\sum_{m=1}^{M} q_m \left(\frac{\eta_m^k}{2} - L(\eta_m^k)^2 - 2\gamma_1(\eta_m^k)^2\right)\mathbb{E}[\|\nabla_{\theta_m} F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] \\
& + \sum_{d=1}^{D}\left(Dc^k + D\gamma_1 \max_m 2(\eta_m^k)^2 L_m^2 + \gamma_{d+1} - \gamma_d\right) \\
& \times \mathbb{E}[\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|^2] \\
& + L(\eta_0^k)^2\sigma_0^2 + \sum_{m=1}^{M} q_m (L + 2\gamma_1)(\eta_m^k)^2 \sigma_m^2.
\end{aligned} \tag{18}$$

If $\{\gamma_d\}$ are chosen properly as specified in the supplementary materials, the first three terms in the right hand side of (18) is negative. By carefully choosing $\{\eta_0^k, \eta_m^k\}$, we can ensure the convergence of Algorithm 1.

We first quantify the descent amount in the objective value.

Lemma 2. *Under Assumptions 1-3, the iterates $\{\theta_0^k, \boldsymbol{\theta}^k\}$ generated by Algorithm 1 satisfy*

$$\begin{aligned}
\mathbb{E}[F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1})|\Theta^k] \le & F(\theta_0^k, \boldsymbol{\theta}^k) + c_k \mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|\Theta^k] + L(\eta_0^k)^2\sigma_0^2 \\
& + \sum_{m=1}^{M} q_m L(\eta_m^k)^2\sigma_m^2 - (\frac{\eta_0^k}{2} - L(\eta_0^k)^2)\|G_0^k\|^2 - \sum_{m=1}^{M} q_m (\frac{\eta_m^k}{2} - L(\eta_m^k)^2)\|G_m^k\|^2
\end{aligned} \tag{19}$$

where Θ^k is the σ-algebra generated by $\{\theta_0^0, \boldsymbol{\theta}^0, \ldots, \theta_0^k, \boldsymbol{\theta}^k\}$, and c_k is defined as

$$c_k := \left(\frac{\eta_0^k}{2} + L(\eta_0^k)^2\right) L_0^2 + \max_m \left(\frac{\eta_m^k}{2} + L(\eta_m^k)^2\right) L_m^2. \tag{20}$$

Proof. By Assumption 2, we have

$$\begin{aligned}
& F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1}) \\
=& F(\theta_0^k - \eta_0^k \hat{g}_0^k, \ldots, \theta_{m_k}^k - \hat{g}_{m_k}^k, \ldots) \\
\le & F(\theta_0^k, \boldsymbol{\theta}^k) - \eta_0^k \langle G_0^k, \hat{g}_0^k\rangle - \eta_{m_k}^k \langle G_{m_k}^k, \hat{g}_{m_k}^k\rangle + \frac{L(\eta_0^k)^2}{2}\|\hat{g}_0^k\|^2 + \frac{L(\eta_{m_k}^k)^2}{2}\|\hat{g}_{m_k}^k\|^2 \\
\le & F(\theta_0^k, \boldsymbol{\theta}^k) - \eta_0^k \langle G_0^k, g_0^k\rangle - \eta_0^k \langle G_0^k, \hat{g}_0^k - g_0^k\rangle - \eta_{m_k}^k \langle G_{m_k}^k, g_{m_k}^k\rangle - \eta_{m_k}^k \langle G_{m_k}^k, \hat{g}_{m_k}^k - g_{m_k}^k\rangle + \frac{L(\eta_0^k)^2}{2}\|\hat{g}_0^k\|^2 + \frac{L(\eta_{m_k}^k)^2}{2}\|\hat{g}_{m_k}^k\|^2 \\
\le & F(\theta_0^k, \boldsymbol{\theta}^k) - \eta_0^k \langle G_0^k, g_0^k\rangle - \eta_{m_k}^k \langle G_{m_k}^k, g_{m_k}^k\rangle + \frac{\eta_0^k}{2}\|G_0^k\|^2 + \frac{\eta_{m_k}^k}{2}\|G_{m_k}^k\|^2 + L(\eta_0^k)^2\|g_0^k\|^2 + L(\eta_{m_k}^k)^2\|g_{m_k}^k\|^2 \\
& + (\frac{\eta_0^k}{2} + L(\eta_0^k)^2)\|\hat{g}_0^k - g_0^k\|^2 + (\frac{\eta_{m_k}^k}{2} + L(\eta_{m_k}^k)^2)\|\hat{g}_{m_k}^k - g_{m_k}^k\|^2.
\end{aligned} \tag{21}$$

Note that we have

$$\begin{aligned}
\mathbb{E}\left[\|g_{m_k}^k\|^2|\Theta^k\right] &= \mathbb{E}\left[\|g_{m_k}^k - G_{m_k}^k + G_{m_k}^k\|^2|\Theta^k\right] \\
&= \mathbb{E}\left[\|g_{m_k}^k - G_{m_k}^k\|^2|\Theta^k\right] + 2\mathbb{E}\left[\langle g_{m_k}^k - G_{m_k}^k, G_{m_k}^k\rangle|\Theta^k\right] + \|G_{m_k}^k\|^2 \\
&= \sigma_{m_k}^2 + \|G_{m_k}^k\|^2
\end{aligned} \tag{22}$$

where the last equality follows from Assumption 1.

First we take expectation on (21) with respect to n_k, conditioned on Θ^k and $m_k = m$, we have

$$\begin{aligned}\mathbb{E}[F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1})|m_k = m, \Theta^k] \leq & F(\theta_0^k, \boldsymbol{\theta}^k) - (\frac{\eta_0^k}{2} - L(\eta_0^k)^2)\|G_0^k\|^2 - (\frac{\eta_m^k}{2} - L(\eta_m^k)^2)\|G_m^k\|^2 + L(\eta_0^k)^2\sigma_0^2 + L(\eta_m^k)^2\sigma_m^2 \\ & + (\frac{\eta_0^k}{2} + L(\eta_0^k)^2)\mathbb{E}[\|\hat{g}_0^k - g_0^k\|^2|m_k = m] + (\frac{\eta_m^k}{2} + L(\eta_m^k)^2)\mathbb{E}[\|\hat{g}_m^k - g_m^k\|^2|m_k = m].\end{aligned} \tag{23}$$

Then taking expectation with respect to m_k, it follows that

$$\begin{aligned}\mathbb{E}[F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1})|\Theta^k] \leq & F(\theta_0^k, \boldsymbol{\theta}^k) - (\frac{\eta_0^k}{2} - L(\eta_0^k)^2)\|G_0^k\|^2 - \sum_{m=1}^M q_m(\frac{\eta_m^k}{2} - L(\eta_m^k)^2)\|G_m^k\|^2 + L(\eta_0^k)^2\sigma_0^2 + \sum_{m=1}^M q_m L(\eta_m^k)^2\sigma_m^2 \\ & + (\frac{\eta_0^k}{2} + L(\eta_0^k)^2)L_0^2\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|\Theta^k] + \sum_{m=1}^M q_m(\frac{\eta_m^k}{2} + L(\eta_m^k)^2)L_m^2\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|m_k = m, \Theta^k] \\ \leq & F(\theta_0^k, \boldsymbol{\theta}^k) - (\frac{\eta_0^k}{2} - L(\eta_0^k)^2)\|G_0^k\|^2 - \sum_{m=1}^M q_m(\frac{\eta_m^k}{2} - L(\eta_m^k)^2)\|G_m^k\|^2 + L(\eta_0^k)^2\sigma_0^2 + \sum_{m=1}^M q_m L(\eta_m^k)^2\sigma_m^2 \\ & + \underbrace{\left(\left(\frac{\eta_0^k}{2} + L(\eta_0^k)^2\right)L_0^2 + \max_m\left(\frac{\eta_m^k}{2} + L(\eta_m^k)^2\right)L_m^2\right)}_{:=c^k}\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|\Theta^k]\end{aligned}$$

which completes the proof. □

B. Convergence under bounded delay

Recalling the definition of $\hat{\boldsymbol{\theta}}^k$ in (16g), if $\tau_{n_k,m}^k \leq D$, then it can be derived that

$$\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2 \leq \sum_{d=1}^D D\left\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\right\|^2. \tag{24}$$

B.1. Proof of Lemma 1

Recall the definition of V^k, that is

$$V^k = F(\theta_0^k, \boldsymbol{\theta}^k) + \sum_{d=1}^D \gamma_d\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|^2$$

where we initialize the algorithm with $\boldsymbol{\theta}^{-D+1} = \cdots = \boldsymbol{\theta}^{-1} = \boldsymbol{\theta}^0$. We first decompose $\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2$ as

$$\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 = (\eta_{m_k}^k)^2\|\hat{g}_{m_k}^k\|^2 \leq 2(\eta_{m_k}^k)^2\|g_{m_k}^k\|^2 + 2(\eta_{m_k}^k)^2\|\hat{g}_{m_k}^k - g_{m_k}^k\|^2. \tag{25}$$

Taking expectation on both sides of (25), and applying (22) leads to

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2|\Theta^k\right] &= \sum_{m=1}^M \mathbb{E}\left[\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2|m_k = m, \Theta^k\right]\mathbb{P}(m_k = m) \\ &\leq \sum_{m=1}^M 2q_m(\eta_m^k)^2\|G_m^k\|^2 + \sum_{m=1}^M 2q_m(\eta_m^k)^2\sigma_m^2 + 2\max_m(\eta_m^k)^2L_m^2\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|\Theta^k].\end{aligned} \tag{26}$$

Following Lemma 2 in Appendix A and (26), the Lyapunov function V^k satisfies

$$\begin{aligned}\mathbb{E}[V^{k+1}|\Theta^k] - V^k =& \mathbb{E}[F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1})|\Theta^k] - F(\theta_0^k, \boldsymbol{\theta}^k) + \gamma_1 \mathbb{E}[\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2|\Theta^k] \\ &+ \sum_{d=1}^{D-1} (\gamma_{d+1} - \gamma_d)\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|^2 - \gamma_D \|\boldsymbol{\theta}^{k+1-D} - \boldsymbol{\theta}^{k-D}\|^2 \\ \leq& -\left(\frac{\eta_0^k}{2} - L(\eta_0^k)^2\right)\|G_0^k\|^2 - \sum_{m=1}^{M} q_m \left(\frac{\eta_m^k}{2} - L(\eta_m^k)^2 - 2\gamma_1(\eta_m^k)^2\right)\|G_m^k\|^2 \\ &+ L(\eta_0^k)^2\sigma_0^2 + \sum_{m=1}^{M} q_m (L + 2\gamma_1)(\eta_m^k)^2\sigma_m^2 \\ &+ \sum_{d=1}^{D-1} \left(Dc^k + D\gamma_1 \max_m 2(\eta_m^k)^2 L_m^2 + \gamma_{d+1} - \gamma_d\right)\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|^2 \\ &+ \left(Dc^k + D\gamma_1 \max_m 2(\eta_m^k)^2 L_m^2 - \gamma_D\right)\|\boldsymbol{\theta}^{k+1-D} - \boldsymbol{\theta}^{k-D}\|^2. \end{aligned} \tag{27}$$

Since we choose $\eta_0^k, \eta_m^k \leq \bar{\eta} \leq \frac{1}{4(L+2\gamma_1)}$, it follows that $c^k \leq \frac{3}{2}\bar{\eta}L^2$. By taking expectation on both sides of (27), we have

$$\begin{aligned}\mathbb{E}V^{k+1} - \mathbb{E}V^k \leq& -\frac{1}{4}\min\{\eta_0^k, q_m\eta_m^k\}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] + L(\eta_0^k)^2\sigma_0^2 + (2\gamma_1 + L)\sum_{m=1}^{M} q_m(\eta_m^k)^2\sigma_m^2 \\ &- \sum_{d=1}^{D-1}\left(\gamma_d - \gamma_{d+1} - \frac{3}{2}D\bar{\eta}L^2 - 2D\gamma_1\bar{\eta}^2L^2\right)\mathbb{E}\|\boldsymbol{\theta}^{k+1-d} - \boldsymbol{\theta}^{k-d}\|^2 \\ &- \left(\gamma_D - \frac{3}{2}D\bar{\eta}L^2 - 2D\gamma_1\bar{\eta}^2L^2\right)\mathbb{E}\|\boldsymbol{\theta}^{k+1-D} - \boldsymbol{\theta}^{k-D}\|^2. \end{aligned} \tag{28}$$

B.2. Proof of Theorem 1

Define $\gamma_1 = \frac{\frac{3}{2}\bar{\eta}D^2L^2}{1-2D^2\bar{\eta}^2L^2} \leq \frac{1}{2}DL$, and $\eta_0^k = \eta_m^k = \eta = \min\{\frac{1}{4(D+1)L}, \frac{c_\eta}{\sqrt{K}}\}$. Select $\gamma_2, \ldots, \gamma_D$ as follows

$$\gamma_{d+1} = \gamma_d - \frac{3}{2}D\eta L^2 - 2D\gamma_1\eta^2L^2, \quad d = 1, \ldots, D-1.$$

It can be verified that $\gamma_D - \frac{3}{2}D\eta L^2 - 2D\gamma_1\eta^2L^2 \geq 0$. Then (28) reduces to

$$\mathbb{E}V^{k+1} - \mathbb{E}V^k \leq -\frac{1}{4}\min_m q_m\eta\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] + \eta^2L\sigma_0^2 + \eta^2(2\gamma_1 + L)\sum_{m=1}^{M} q_m\sigma_m^2. \tag{29}$$

By summing over $k = 0, \cdots, K-1$ and using $\eta \leq \frac{c_\eta}{\sqrt{K}}$, it follows that

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] \leq& \frac{F^0 - F^* + K\eta^2L\sigma_0^2 + K\eta^2(2\gamma_1 + L)\sum_{m=1}^{M} q_m\sigma_m^2}{\frac{1}{4}\min_m q_m\eta K} \\ \leq& \frac{16DL(F^0 - F^*)}{\min_m q_m K} + \frac{4c_\eta(F^0 - F^*)}{\min_m q_m\sqrt{K}} + \frac{4c_\eta L\sigma_0^2 + c_\eta(8\gamma_1 + 4L)\sum_{m=1}^{M} q_m\sigma_m^2}{\min_m q_m\sqrt{K}}. \end{aligned}$$

B.3. Proof of Theorem 2

By the μ-strong convexity of $F(\theta_0, \boldsymbol{\theta})$, we have

$$2\mu(F(\theta_0, \boldsymbol{\theta}) - F^*) \leq \|\nabla F(\theta_0, \boldsymbol{\theta})\|^2. \tag{30}$$

Choose γ_d such that

$$\gamma_d - \gamma_{d+1} - \frac{3}{2}D\bar{\eta}L^2 - 2D\gamma_1\bar{\eta}^2L^2 = \frac{\mu}{2}\min_m q_m\bar{\eta}\gamma_1, \quad d = 1, \ldots, D-1$$
$$\gamma_D - \frac{3}{2}D\bar{\eta}L^2 - 2D\gamma_1\bar{\eta}L^2 = \frac{\mu}{2}\min_m q_m\bar{\eta}\gamma_1.$$

Solve the above linear equations above and get

$$\gamma_1 = \frac{\frac{3}{2}\bar{\eta}D^2L^2}{1 - 2D^2\bar{\eta}^2L^2 - \frac{\mu}{2}\min_m q_m D\bar{\eta}}, \quad \gamma_d = (D+1-d)(\frac{3}{2}D\bar{\eta}L^2 + 2D\gamma_1\bar{\eta}L^2 + \frac{\mu}{2}\min_m q_m\bar{\eta}\gamma_1), \quad d = 1, \ldots, D-1.$$

If we choose $\eta_0^k = \eta_m^k = \eta^k \leq \bar{\eta} \leq \frac{1}{4(D+1)L + 2\mu\min_m q_m D}$, $\gamma_1 \leq 2\bar{\eta}D^2L^2$, then (28) reduces to

$$\mathbb{E}V^{k+1} \leq (1 - \frac{\mu}{2}\eta^k\min_m q_m)\mathbb{E}V^k + (\eta^k)^2\left(L\sigma_0^2 + (2\gamma_1 + L)\sum_{m=1}^M q_m\sigma_m^2\right).$$

Defining $R := \left(L\sigma_0^2 + (2\gamma_1 + L)\sum_{m=1}^M q_m\sigma_m^2\right)$ and $\eta^k = \frac{4}{\mu\min_m q_m(k+K_0)}$, where $K_0 = \frac{4(4(D+1)L + 2\mu\min_m q_m D)}{\mu\min_m q_m}$, we have

$$\begin{aligned}
\mathbb{E}V^k &\leq V^0\prod_{k=0}^{K-1}(1 - \frac{\mu}{2}\min_m q_m\eta^k) + R\sum_{k=0}^{K-1}(\eta^k)^2\prod_{j=k+1}^{K-1}(1 - \frac{\mu}{2}\min_m q_m\eta^j) \\
&= V^0\prod_{k=0}^{K-1}\frac{k + K_0 - 2}{k + K_0} + \frac{16R}{\mu^2\min_m q_m}\sum_{k=0}^{K-1}\frac{1}{(k+K_0)^2}\prod_{j=k+1}^{K-1}\frac{j + K_0 - 2}{j + K_0} \\
&\leq \frac{(K_0-2)(K_0-1)}{(K+K_0-2)(K+K_0-1)}V^0 + \frac{16R}{\mu^2\min_m q_m}\sum_{k=0}^{K-1}\frac{1}{(k+K_0)^2}\frac{(k+K_0-1)(k+K_0)}{(K+K_0-2)(K+K_0-1)} \\
&\leq \frac{(K_0-1)^2}{(K+K_0-1)^2}(F(\theta_0^0, \boldsymbol{\theta}^0) - F^*) + \frac{16RK}{\mu^2\min_m q_m(K+K_0-1)^2}.
\end{aligned}$$

C. Convergence under stochastic unbounded delay

We first present a useful fact. Given the definition of $\bar{p}_m, p_{m,d}$ in Assumption 5, it can be shown that

$$\begin{aligned}
\sum_{s=d}^{\infty} sp_{m,s} &= \bar{p}_m\left(\frac{d\rho^d}{1-\rho} + \frac{\rho^{d+1}}{(1-\rho)^2}\right) := c_{m,d} \\
\sum_{s=d}^{\infty} c_{m,s} &= \bar{p}_m\left(\frac{d\rho^d}{(1-\rho)^2} + \frac{2\rho^{d+1}}{(1-\rho)^3}\right) \\
\sum_{d=1}^{\infty} c_{m,d} &= \bar{p}_m\left(\frac{\rho}{(1-\rho)^2} + \frac{2\rho^2}{(1-\rho)^3}\right) := c_m.
\end{aligned}$$

For unbounded delay, we have the following relation

$$\begin{aligned}\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k-\boldsymbol{\theta}^k\|^2|\Theta^k]&=\sum_{m=1}^{M}\mathbb{E}[\|\theta_m^{k-\tau_{n_k,m}^k}-\theta_m^k\|^2|\Theta^k]\\&=\sum_{m=1}^{M}\sum_{s=1}^{\infty}\mathbb{E}[\|\theta_m^{k-s}-\theta_m^k\||\Theta^k]\mathbb{P}(\tau_{n_k,m}^k=s)\\&\le\sum_{m=1}^{M}\sum_{s=1}^{\infty}\sum_{d=1}^{s}sp_{m,s}\|\theta_m^{k+1-d}-\theta_m^{k-d}\|^2\\&=\sum_{m=1}^{M}\sum_{d=1}^{\infty}c_{m,d}\|\theta_m^{k+1-d}-\theta_m^{k-d}\|^2.\end{aligned}$$

Similar to (26), we can decompose the difference term as

$$\mathbb{E}\left[\|\theta_m^{k+1}-\theta_m^k\|^2|\Theta^k\right]\le 2q_m(\eta_m^k)^2\|G_m^k\|^2+2q_m(\eta_m^k)^2\sigma_m^2+2q_m(\eta_m^k)^2L^2\mathbb{E}[\|\hat{\theta}_m^k-\theta_m^k\|^2|\Theta^k]. \tag{31}$$

Following Lemma 2 and (31), we have

$$\begin{aligned}&\mathbb{E}[V^{k+1}|\Theta^k]-V^k\\=&\mathbb{E}[F(\theta_0^{k+1},\boldsymbol{\theta}^{k+1})|\Theta^k]-F(\theta_0^k,\boldsymbol{\theta}^k)+\sum_{m=1}^{M}\gamma_{m,1}\mathbb{E}[\|\theta_m^{k+1}-\theta_m^k\|^2|\Theta^k]+\sum_{m=1}^{M}\sum_{d=1}^{\infty}(\gamma_{d+1}-\gamma_d)\|\theta_m^{k+1-d}-\theta_m^{k-d}\|^2\\\le&-\left(\frac{\eta_0^k}{2}-L(\eta_0^k)^2\right)\|G_0^k\|^2-\sum_{m=1}^{M}q_m\left(\frac{\eta_m^k}{2}-(2\gamma_{m,1}+L)(\eta_m^k)^2\right)\|G_m^k\|^2+L(\eta_0^k)^2\sigma_0^2\\&+\sum_{m=1}^{M}q_m(\eta_m^k)^2(2\gamma_{m,1}+L)\sigma_m^2+\sum_{m=1}^{M}\sum_{d=1}^{k}\left(\left(c^k+2q_m\gamma_{m,1}(\eta_m^k)^2L^2\right)c_{m,d}+\gamma_{m,d+1}-\gamma_{m,d}\right)\|\theta_m^{k+1-d}-\theta_m^{k-d}\|^2.\end{aligned}$$

If we choose $\eta_0^k,\eta_m^k\le\bar{\eta}\le\frac{1}{4(L+2\max_m\gamma_{m,1})}$, then $c^k\le\frac{3}{2}\bar{\eta}L^2$. By direct calculation, we have

$$\begin{aligned}\mathbb{E}V^{k+1}-\mathbb{E}V^k\le&-\frac{1}{4}\min\{\eta_0^k,q_m\eta_m^k\}\mathbb{E}[\|\nabla F(\theta_0^k,\boldsymbol{\theta}^k)\|^2]+L(\eta_0^k)^2\sigma_0^2+\sum_{m=1}^{M}q_m(\eta_m^k)^2(2\gamma_{m,1}+L)\sigma_m^2\\&-\sum_{m=1}^{M}\sum_{d=1}^{\infty}\left(\gamma_{m,d}-\gamma_{m,d+1}-c_{m,d}\left(\frac{3}{2}\bar{\eta}L^2+2q_m\gamma_{m,1}\bar{\eta}^2L^2\right)\right)\mathbb{E}[\|\theta_m^{k+1-d}-\theta_m^{k-d}\|^2].\end{aligned} \tag{32}$$

If we select $\gamma_{m,d}$ such that

$$\left(\frac{3}{2}\bar{\eta}L^2+2q_m\gamma_{m,1}\bar{\eta}^2L^2\right)c_{m,d}+\gamma_{m,d+1}-\gamma_{m,d}=-\xi_mc_{m,d},\quad m=1,\dots,M,\ d=1,\dots,\infty$$

then it remains that

$$\gamma_{m,d}=\sum_{s=d}^{\infty}c_{m,s}\left(\frac{3}{2}\bar{\eta}L^2+2q_m\gamma_{m,1}\bar{\eta}^2L^2+\xi_m\right).$$

C.1. Proof of Theorem 3

We set the parameters as

$$\xi_m=0,\qquad c_m=\sum_{d=1}^{\infty}c_{m,d},\qquad \gamma_{m,1}=\frac{\frac{3}{2}c_m\bar{\eta}L^2}{1-2c_mq_m\bar{\eta}^2L^2}\le 2\bar{\eta}c_mL^2\le\frac{1}{2}\sqrt{c_m}L \tag{33}$$

and

$$\eta_0^k = \eta_m^k = \eta = \min\left\{\frac{1}{4(1+\max_m\sqrt{c_m})L}, \frac{c_\eta}{\sqrt{K}}\right\}. \tag{34}$$

Plugging these constants into (32), we have

$$\mathbb{E}V^{k+1} - \mathbb{E}V^k \le \frac{1}{4}\min_m q_m\eta\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] + \eta^2\sum_{m=1}^M q_m(2\gamma_{m,1}+L)\sigma_m^2. \tag{35}$$

By summing (35) over $k = 0, \cdots, K-1$, it follows that

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] &\le \frac{F^0 - F^* + K\eta^2\sum_{m=1}^M(2\gamma_{m,1}+L)q_m\sigma_m^2}{\frac{1}{4}\min_m q_m\eta K} \\ &\le \frac{16(1+\max_m\sqrt{c_m})L(F^0-F^*)}{\min_m q_m K} + \frac{4c_\eta(F^0-F^*)}{\min_m q_m\sqrt{K}} + \frac{4c_\eta\sum_{m=1}^M(2\gamma_{m,1}+L)q_m\sigma_m^2}{\min_m q_m\sqrt{K}}.\end{aligned}$$

C.2. Proof of Theorem 4

If we set $\xi_m = \frac{1}{4}c_m\bar{\eta}L^2$ and $\eta^k \le \bar{\eta} = \frac{1}{4(1+\max_m\sqrt{c_m})L}$, then

$$\gamma_{m,1} = \frac{\frac{3}{2}c_m\bar{\eta}L^2 + \xi_m}{1 - 2c_m\bar{\eta}^2L^2} = 2c_m\bar{\eta}L^2 \le \frac{1}{2}\sqrt{c_m}L.$$

Plugging the parameters in (32) and using the strong convexity in (30), we have

$$\mathbb{E}V^{k+1} \le (1-\nu\eta^k)\mathbb{E}V^k + (\eta^k)^2R$$

where $\nu = \inf_{m,d}\{\frac{\xi_m c_{m,d}}{\bar{\eta}\gamma_{m,d}}, \frac{\mu q_m}{2}\}$ and $R := \big(L\sigma_0^2 + (2\gamma_{m,1}+L)\sum_{m=1}^M q_m\sigma_m^2\big)$.

Choosing $\eta^k = \frac{2}{\nu(k+K_0)}$ with $K_0 = \frac{4(1+\max_m\sqrt{c_m})L}{\nu}$, it follows that

$$\begin{aligned}\mathbb{E}V^k &\le \prod_{k=0}^{K-1}(1-\nu\eta^k)V^0 + R\sum_{k=0}^{K-1}(\eta^k)^2\prod_{j=k+1}^{K-1}(1-\nu\eta^j) \\ &= V^0\prod_{k=0}^{K-1}\frac{k+K_0-2}{k+K_0} + \frac{16R}{\mu^2\min_m q_m}\sum_{k=0}^{K-1}\frac{1}{(k+K_0)^2}\prod_{j=k+1}^{K-1}\frac{j+K_0-2}{j+K_0} \\ &\le \frac{(K_0-2)(K_0-1)}{(K+K_0-2)(K+K_0-1)}V^0 + \frac{16R}{\mu^2\min_m q_m}\sum_{k=0}^{K-1}\frac{1}{(k+K_0)^2}\frac{(k+K_0-1)(k+K_0)}{(K+K_0-2)(K+K_0-1)} \\ &\le \frac{(K_0-1)^2}{(K+K_0-1)^2}(F^0-F^*) + \frac{16RK}{\mu^2\min_m q_m(K+K_0-1)^2}.\end{aligned}$$

Remark 1. *To verify the existence of $\nu > 0$, we have*

$$\begin{aligned}\frac{\xi_m c_{m,d}}{\bar{\eta}\gamma_{m,d}} &= \frac{c_{m,d}}{\sum_{s=d}^\infty c_{m,s}}\frac{1}{\bar{\eta}}\frac{\xi_m}{\frac{3}{2}\bar{\eta}L^2 + 2q_m\gamma_{m,1}\bar{\eta}^2L^2 + \xi_m} \ge \frac{1}{\bar{\eta}}\frac{(1-\rho)\xi_m}{3\bar{\eta}L^2 + 4q_m\gamma_{m,1}\bar{\eta}^2L^2 + 2\xi_m} \\ &\ge \frac{1}{\bar{\eta}}\frac{(1-\rho)\xi_m}{5\bar{\eta}L^2 + \xi_m} = \frac{1}{\bar{\eta}}\frac{(1-\rho)c_m}{20+c_m}\end{aligned}$$

where we use the fact that $\frac{c_{m,d}}{\sum_{s=d}^\infty c_{m,s}} \ge \frac{1-\rho}{2}$. Then $\nu = \min\{\frac{1}{\bar{\eta}}\frac{(1-\rho)c_m}{20+c_m}, \frac{\mu q_m}{2}\}$.

D. Convergence results of vertical t-synchronous federated learning

In the t-synchronous, we use $\mathcal{M}^k$ to denote the set of clients that upload at iteration k. For notational brevity, we define

$$\hat{g}_0^k = \frac{1}{t}\sum_{m\in\mathcal{M}^k} \nabla_{\theta_0}\ell(\theta_0, h_{n_k,1}^{k-\tau^k_{n_k(m),1}}, \ldots, h_{n_k(m),M}^{k-\tau^k_{n_k(m),M}}; y_{n_k})$$

$$\hat{g}_m^k = \begin{cases} \nabla h_m(\theta_m^k; x_{n_k(m),m})\nabla_{h_m}\ell(\theta_0^k, h_{n_k(m),1}^{k-\tau^k_{n_k(m),1}}, \ldots, h_{n_k(m),M}^{k-\tau^k_{n_k(m),M}}; y_{n_k(m)}), & \text{if } m\in\mathcal{M}^k; \\ 0, & \text{else.} \end{cases}$$

Similar to Assumption 3, we assume that

Assumption 6. *The probability of client m in the set of uploading clients $\mathcal{M}_k$ at iteration k is independent of $\mathcal{M}^{k-1}, \cdots, \mathcal{M}^1$, and it satisfies*

$$\mathbb{P}(m\in\mathcal{M}^k) := q_m.$$

D.1. Connecting with asynchronous case

Similar to the previous analysis, the objective value satisfies the following inequality

$$\begin{aligned} F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1}) &\le F(\theta_0^k, \boldsymbol{\theta}^k) + \langle G_0^k, \theta_m^{k+1} - \theta_0^k\rangle + \langle G_m^k, \theta_m^{k+1} - \theta_m^k\rangle + \frac{L}{2}\|\theta_0^k - \boldsymbol{\theta}^k\|^2 + \sum_{m=1}^M \frac{L}{2}\|\theta_m^{k+1} - \theta_m^k\|^2 \\ &= F(\theta_0^k, \boldsymbol{\theta}^k) + \langle G_0^k, \hat{g}_0^k\rangle + \sum_{m\in\mathcal{M}^k}\langle G_m^k, \hat{g}_m^k\rangle + \frac{L(\eta_0^k)^2}{2}\|\hat{g}_0^k\|^2 + \frac{L}{2}\sum_{m\in\mathcal{M}^k}(\eta_m^k)^2\|\hat{g}_m^k\|^2 \\ &\le F(\theta_0^k, \boldsymbol{\theta}^k) - \eta_0^k\langle G_0^k, g_0^k\rangle - \sum_{m\in\mathcal{M}^k}\eta_m^k\langle G_m^k, g_m^k\rangle + \frac{\eta_0^k}{2}\|G_0^k\|^2 + \sum_{m\in\mathcal{M}}\frac{\eta_m^k}{2}\|G_m^k\|^2 + L(\eta_0^k)^2\|G_0^k\|^2 \\ &\quad + \sum_{m\in\mathcal{M}^k} L(\eta_m^k)^2\|G_m^k\|^2 + (\frac{\eta_0^k}{2} + L(\eta_0^k)^2)\|\hat{g}_0^k - g_0^k\|^2 + \sum_{m\in\mathcal{M}^k}(\frac{\eta_m^k}{2} + L(\eta_m^k)^2)\|\hat{g}_m^k - g_m^k\|^2. \end{aligned}$$

And by taking expectation with respect to $\mathcal{M}^k, n_k(m)$, it follows that (with $t = \sum_{m=1}^M q_m$)

$$\begin{aligned} &\mathbb{E}[F(\theta_0^{k+1}, \boldsymbol{\theta}^{k+1})|\Theta^k] \\ \le& F(\theta_0^k, \boldsymbol{\theta}^k) - (\frac{\eta_0^k}{2} - L(\eta_0^k)^2)\|G_0^k\|^2 - \sum_{m=1}^M q_m(\frac{\eta_m^k}{2} - L(\eta_m^k)^2)\|G_m^k\|^2 + \frac{1}{t^2}L(\eta_0^k)^2\sigma_0^2 + \sum_{m=1}^M q_m L(\eta_m^k)^2\sigma_m^2 \\ &+ \left(\left(\frac{\eta_0^k}{2} + L(\eta_0^k)^2\right)L_0^2 + t\max_m\left(\frac{\eta_m^k}{2} + L(\eta_m^k)^2\right)L_m^2\right)\mathbb{E}[\|\hat{\boldsymbol{\theta}}^k - \boldsymbol{\theta}^k\|^2|\Theta^k]. \end{aligned}$$

Following the Lyapunov analysis of the asynchronous case, it can be shown that the vertical t-synchronous federated learning achieves the same order of convergence rate as in Theorems 1-4.

D.2. Convergence results

For completeness, we state the convergence results for the vertical t-synchronous federated learning as follows.

Theorem 7 (Bounded delay, nonconvex)**.** *Under Assumptions 1,2,4 and 6, if $\eta_0^k = t\eta_m^k = \min\{\frac{1}{4(1+D)L}, \frac{c_\eta}{\sqrt{K}}\}$ with $c_\eta > 0$, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] = \mathcal{O}\left(1/\sqrt{K}\right). \tag{37}$$

Theorem 8 (Bounded delay, strongly convex)**.** *Assume that F is μ-strongly convex in $(\theta_0, \boldsymbol{\theta})$. Then under the same assumptions of Theorem 7, if $\eta^k = \frac{4}{\mu\min_m\sqrt{q_m}(k+K_0)}$ with $K_0 = \frac{4(4(D+1)L + \mu\min_m\sqrt{q_m}D)}{\mu t\min_m\sqrt{q_m}}$, then*

$$\mathbb{E}F\left(\theta_0^K, \boldsymbol{\theta}^K\right) - F^* = \mathcal{O}\left(1/K\right). \tag{38}$$

Theorem 9 (Unbounded stochastic delay, nonconvex)**.** *Under Assumptions 1,2,5 and 6, if we choose* $\eta_0^k = t\eta_m^k = \min\left\{\frac{1}{4(1+\min_m\sqrt{c_m})L}, \frac{c_\eta}{\sqrt{K}}\right\}$*, then we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\theta_0^k, \boldsymbol{\theta}^k)\|^2] = \mathcal{O}\left(1/\sqrt{K}\right). \tag{39}$$

Theorem 10 (Unbounded stochastic delay, strongly convex)**.** *Assume that* F *is* μ*-strongly convex in* $(\theta_0, \boldsymbol{\theta})$*. Then under the same assumptions of Theorem 9, if* $\eta_0^k = \eta_m^k = \frac{2}{\nu(k+K_0)}$ *where* $K_0 = \frac{4(1+\max_m\sqrt{c_m})L}{t\nu}$ *and* ν *is a positive constant depending on* $\mu, L, \bar{p}_m, \rho$*, then it follows that*

$$\mathbb{E}F\left(\theta_0^K, \boldsymbol{\theta}^K\right) - F^* = \mathcal{O}\left(1/K\right). \tag{40}$$

E. Proof of Theorem 5

Before proceeding to the proof of Theorem 5, we first present the smoothness of a single neuron in the following lemma.

Lemma 3. *If* $\sigma(x)$ *is* L_σ^0*-Lipschitz continuous and differentiable almost everywhere,* Z *is a continuous random variable with pdf* $\mu(Z)$*, then* $\bar{\sigma}(x) := \mathbb{E}\sigma(x+Z)$ *is differentiable with Lipschitz continuous gradient* $\nabla\bar{\sigma}(x) = \mathbb{E}\nabla\sigma(x+Z)$*.*

Proof. We first prove that $\bar{\sigma}(x)$ is smooth and $\mathbb{E}\nabla\sigma(x+Z) = \nabla\bar{\sigma}(x)$.

$$\frac{\mathbb{E}_Z\sigma(x+\delta v+Z) - \mathbb{E}_Z\sigma(x+Z)}{\delta} = \int_{\mathbb{R}^d}\frac{\sigma(x+\delta v+Z)-\sigma(x+Z)}{\delta}\mu(Z)dZ$$

Since σ is differentiable almost everywhere, for any fixed $x\in\mathbb{R}^d$ and directional vector $v\in\mathbb{R}^d$, we have

$$\lim_{\delta\to 0}\frac{\sigma(x+\delta v+Z)-\sigma(x+Z)}{\delta} = v^\top\nabla\sigma(x+Z) = \sum_{i=1}^n\frac{\partial\sigma}{\partial x_i}(x+Z)v_i \qquad a.e.$$

and

$$\int_{\mathbb{R}^d}\left|\frac{\sigma(x+\delta v+Z)-\sigma(x+Z)}{\delta}\right|\mu(Z)dZ \le \int_{\mathbb{R}^d}L_\sigma^0\mu(Z)dZ = L_\sigma^0.$$

Then by dominated convergence theorem, when taking $\delta\to 0$, it follows that

$$\begin{aligned}
\frac{\partial\bar{\sigma}}{\partial x_i}(x) &= \int_{\mathbb{R}^d}\frac{\partial\sigma}{\partial x_i}(x+Z)\mu(Z)dZ,\\
\frac{\partial\bar{\sigma}}{\partial v}(x) &= \int_{\mathbb{R}^d}\sum_{i=1}^d\frac{\partial\sigma}{\partial x_i}(x+Z)v_i\mu(Z)dZ = \sum_{i=1}^n\frac{\partial\bar{\sigma}}{\partial x_i}(x)v_i.
\end{aligned}$$

Therefore, $\bar{\sigma}(x)$ is differentiable, that is

$$\nabla\bar{\sigma}(x) = \int_{\mathbb{R}^d}\nabla\sigma(x+Z)\mu(Z)dZ = \mathbb{E}_Z\nabla\sigma(x+Z).$$

Next we derive the smoothness constant of $\bar{\sigma}(x)$. We focus on the uniform distribution and the Gaussian distribution.

Case I. Assume that the uniform distribution $Z\sim\mathcal{U}[-\frac{c}{2},\frac{c}{2}]^d$, i.e., $\mu(Z) = \frac{1}{c^d}\mathbb{1}_{\{-\frac{c}{2}\le Z_i\le\frac{c}{2}, 1\le i\le d\}}(Z)$.

$$\begin{aligned}
\|\nabla\bar{\sigma}(x)-\nabla\bar{\sigma}(x')\| &= \left\|\int_{\mathbb{R}^d}\nabla\sigma(x+y)\mu(Z)dZ - \int_{\mathbb{R}^d}\nabla\sigma(x'+Z)\mu(Z)dZ\right\|\\
&= \left\|\int_{\mathbb{R}^d}\nabla\sigma(y)(\mu(y-x)-\mu(y-x'))dy\right\| \le L_\sigma^0\int_{\mathbb{R}^d}|\mu(y-x)-\mu(y-x')|\,dy\\
&\le \frac{2\sqrt{d}L_\sigma^0}{c}\|x-x'\| := L_{\bar{\sigma}}(c)\|x-x'\|
\end{aligned}$$

where the smoothness constant is defined as $L_{\bar{\sigma}}(c) := \frac{2\sqrt{d}L_\sigma^0}{c}$.

Case II. Assume that the Gaussian distribution $Z \sim \mathcal{N}(0, c^2 I_d)$, i.e., $\mu(Z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|Z\|^2}{2\sigma^2}}$.

$$\bar{\sigma}(x) = \int_{\mathbb{R}^d} \sigma(x+Z)\mu(Z)dZ = \int_{Z\in\mathbb{R}^d} \sigma(y)\mu(y-x)dy.$$

By the Leibniz rule, we have

$$\nabla\bar{\sigma}(x) = -\int_{\mathbb{R}^d} \sigma(y)\nabla\mu(y-x)dy = -\int_{\mathbb{R}^d} \sigma(y+x)\nabla\mu(y)dy.$$

Then it follows

$$\begin{aligned}\|\nabla\bar{\sigma}(x) - \nabla\bar{\sigma}(x')\| &= \left\|\int_{\mathbb{R}^d} (\sigma(y+x) - \sigma(y+x'))\nabla\mu(y)dy\right\| \\ &\le L_\sigma^0 \left(\int_{\mathbb{R}^d} \|\nabla\mu(y)\|dy\right)\|x-x'\| \\ &= \frac{L_\sigma^0 d}{c}\|x-x'\| := L_{\bar{\sigma}}(c)\|x-x'\|\end{aligned}$$

where the smoothness constant is $L_{\bar{\sigma}}(c) := \frac{L_\sigma^0 d}{c}$. □

E.1. Proof of Theorem 5

Building upon Lemma 3, we next prove Theorem 5. For simplicity, we assume that all the activation functions are same, e.g., $\sigma_l = \sigma$, $\forall l = 1, \cdots, L$. We use L_f to denote the lipschitz constant of a function f. In the following proof, we change the order of differentiation and integration (expectation) as it is supported by Leibniz integral rule. We also let $\bar{f} = \mathbb{E}f$.

Since $\nabla_{b_L}\bar{h} = \mathbb{E}[\nabla\bar{\sigma}], \nabla_{w_L}\bar{h} = \mathbb{E}[\nabla\bar{\sigma}]u_{L-1}^\top, \nabla_{u_{L-1}}\mathbb{E}_{Z_L}h = w_L^\top\mathbb{E}[\nabla\bar{\sigma}]$. The smoothness of $\bar{\sigma}$ implies that $\nabla_{b_L}\bar{h}, \nabla_{w_L}\bar{h}, \nabla_{u_{L-1}}\bar{h}$ are $L_{b_L}^{\bar{h}}, L_{w_L}^{\bar{h}}, L_{u_{L-1}}^{\bar{h}}$-Lipschitz continuous respectively, with

$$\begin{aligned}&L_{b_L}^{\bar{h}} := L_{\bar{\sigma}}(c), \\ &L_{w_L}^{\bar{h}} := L_{b_L}^{\bar{h}}\mathbb{E}[\|u_{L-1}\|], \\ &L_{u_{L-1}}^{\bar{h}} := L_{b_L}^{\bar{h}}\|w_L\|, \\ &\|\nabla_{u_{L-1}}\bar{h}\| \le L_\sigma^0\|w_L\|.\end{aligned}$$

Since σ is differentiable almost everywhere, $\bar{\sigma}(w_L\sigma(\cdot + Z_{L-1}) + b_L)$ is differentiable almost everywhere and thus is smooth in expectation of Z_{L-1}. By some calculation, we can show that

$$\begin{aligned}L_{b_{L-1}}^{\bar{h}} &= L_{u_{L-1}}^{\bar{h}}(L_\sigma^0)^2 + L_\sigma^0\|w_L\|L_{\bar{\sigma}}(c_l) \\ L_{w_{L-1}}^{\bar{h}} &= L_{b_{L-1}}^{\bar{h}}\mathbb{E}[\|u_{L-1}\|] \\ L_{u_{L-2}}^{\bar{h}} &= L_{b_{L-1}}^{\bar{h}}\|w_{L-1}\|.\end{aligned}$$

Following the similar steps, we can obtain that

$$\begin{aligned}L_{b_l}^{\bar{h}} &= L_{u_l}^{\bar{h}}(L_\sigma^0)^2 + L_\sigma^0\|w_L\|\cdots L_\sigma^0\|w_{l+1}\|L_{\bar{\sigma}}(c_l), \\ L_{w_l}^{\bar{h}} &= L_{b_l}^{\bar{h}}\mathbb{E}[\|u_{l-1}\|], \\ L_{u_{l-1}}^{\bar{h}} &= L_{b_l}^{\bar{h}}\|w_l\|.\end{aligned}$$

As long as the overall loss $\ell(\theta_0, h_1, \ldots, h_M; y)$ is smooth w.r.t. $\theta_0, h_1, \ldots, h_M$, we can extend our results to show that it is smooth in the local parameters $\theta_1, \ldots, \theta_M$. Taking u_l from h_m as example, that is

$$L_{\theta_m}^{\bar{\ell}} = L_{h_m}^{\bar{\ell}}(L_{h_m}^0)^2 + L_\ell^0 L_{\theta_m}^{\bar{h}_m}$$

we can extend our results to show that $F_c(\theta_0, \boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}\ell(\theta_0, h_{n,1}, \ldots, h_{n,M}; y_n) + \sum_{m=1}^{M} r(\theta_m)$ is smooth, where the expectation is taken with respect to all the random neurons in local embedding vectors $h_1, \ldots, h_M$. Specifically, the smoothness of F_c is given by

$$L_{\theta_m}^{F_c} = L_{h_m}^{\bar{\ell}}(L_{\bar{h}_m}^{0})^2 + L_{\ell}^{0}\sum_{l=1}^{L}(L_{w_l}^{\bar{h}} + L_{b_l}^{\bar{h}}) + L_{\theta_m}^{r} \tag{41}$$

where $L_{\theta_m}^{r}$ is the smoothness constant of the regularizer w.r.t. θ_m; $L_{b_l}^{\bar{h}}$ and $L_{w_l}^{\bar{h}}$ are the smoothness constants of the perturbed local embedding h w.r.t. the bias b_l and weight w_l.

E.2. The objective difference after local perturbation

Now we evaluate the difference between $F_c(\theta_0, \boldsymbol{\theta})$ and $F(\theta_0, \boldsymbol{\theta})$. Note that

$$\begin{aligned}
|F_c(\theta_0, \boldsymbol{\theta}) - F(\theta_0, \boldsymbol{\theta})|^2 &= \left|\frac{1}{N}\sum_{n=1}^{N}(\mathbb{E}_{\mathbf{Z}}\ell(\theta_0, h'_{n,1}, \ldots, h_{n,M}; \mathbf{Z}) - \ell(\theta_0, h_{n,1}, \ldots, h_{n,M}))\right|^2 \\
&\leq \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{\mathbf{Z}}[|\ell(\theta_0, h'_{n,1}, \ldots, h'_{n,M}; \mathbf{Z}) - \ell(\theta_0, h_{n,1}, \ldots, h_{n,M})|^2] \\
&\leq \frac{M}{N}\sum_{n=1}^{N} L_{\ell}^2 \mathbb{E}_{\mathbf{Z}}[\|h'_{n,m} - h_{n,m}\|^2]
\end{aligned} \tag{42}$$

where $h_{n,m}$ and $h'_{n,m}$ correspond to the outputs of (9) and (10), respectively. Since we have that

$$\|h'_{n,m} - h_{n,m}\| = \|u'_L - u_L\| \leq L_{\sigma_L}(\|w_L\|\|u'_{L-1} - u_{L-1}\| + \|Z_L\|) \leq \cdots \leq \sum_{j=1}^{L}\left(\prod_{l=j}^{L} L_{\sigma_l}\|w_l\|\right)\|Z_l\|$$

and thus it follows that

$$\|h'_{n,m} - h_{n,m}\|^2 \leq \left(\sum_{j=1}^{L}\prod_{l=j}^{L} L_{\sigma_l}^2\|w_l\|^2\right)\left(\sum_{j=1}^{L}\|Z_j\|^2\right).$$

Taking expectation on both side, we have

$$\mathbb{E}[\|h'_{n,m} - h_{n,m}\|^2] \leq \left(\sum_{j=1}^{L}\prod_{l=j}^{L} L_{\sigma_l}^2\|w_l\|^2\right)\left(\sum_{j=1}^{L-1} c_l^2 + c^2\right).$$

Plugging into (42), we arrive at

$$|F_c(\theta_0, \boldsymbol{\theta}) - F(\theta_0, \boldsymbol{\theta})| \leq M\left(\sum_{j=1}^{L}\prod_{l=j}^{L} L_{\sigma_l}^2\|w_l\|^2\right)^{\frac{1}{2}}\left(\sum_{j=1}^{L-1} c_l^2 + c^2\right)^{\frac{1}{2}}.$$

F. Proof of Theorem 6

Let u_l, u'_l denote the the outputs of l-th layer with inputs $u_0 = x, x'$. Under the assumptions that $Z_l \sim \mathcal{U}[-c_l/2, c_l/2]$,$\sigma_l$ is L_{σ_l}-Lipschitz continuous for $l = 1, \ldots, L-1$, we can derive that

$$\begin{aligned}
\|w_L u_{L-1} - w_L u'_{L-1}\| &\leq \|w_L\| L_{\sigma_{L-1}}(\|w_{L-1}\|\|u_{L-2} - u'_{L-2}\| + \sqrt{d_{L-1}}c_{L-1}) \\
&\leq \|w_L\|\prod_{l=1}^{L-1} L_{\sigma_l}\|w_l\|\|x - x'\| + \|w_L\|\sum_{l=1}^{L-1}\left(\prod_{j=1}^{l} L_{\sigma_j}\sqrt{d_j}\right)c_l \\
&:= \bar{B}.
\end{aligned}$$

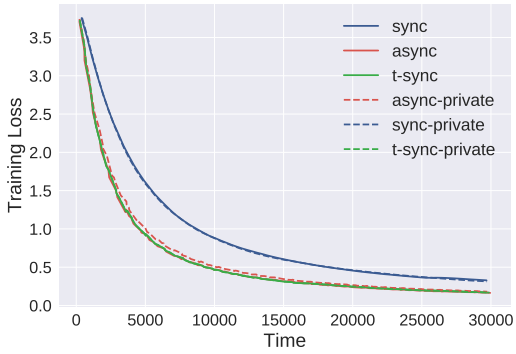

Figure 5. Training loss of VAFL with nonlinear local embedding on *ModelNet40* dataset.

Consider the linear operation of L-th layer $\mathcal{M}(u_{L-1}) = w_L u_{L-1} + b_L + Z_L$ which is a random mechanism defined by $Z_L \sim \mathcal{N}(0, \nu^2)$. Since differential privacy is immune to post-processing [11], $\sigma_L \circ \mathcal{M}$ does not increase the privacy loss compared with $\mathcal{M}$. According to Theorem 1 in [1], Algorithm 1 is (ε, δ)-differentially private if $\nu = c\frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}$.

G. Simulation details

In this section, we present the details of our simulations, and provide the additional test results.

G.1. Simulation environment

We conducted our simulations on a deep learning workstation with 2 Nvidia Titan V and 2 Nvidia GeForce RTX 2080 Ti GPUs. Codes are written using Python 3.6 and Tensorflow 2.0.

G.2. VAFL for federated logistic regression

Data allocation. The datasets we choose are CIFAR-10, Parkinson Disease, MNIST and Fashion MNIST. The batch size is selected to be approximate 0.01 fraction of the entire training dataset. The data are uniformly distributed among $M = 8$ clients for CIFAR-10, $M = 3$ for Parkinson Disease, and $M = 7$ for both MNIST and Fashion MNIST.

Stepsize. The stepsize is $\eta = 1 \times 10^{-2}$ for Parkinson Disease, $\eta = 2 \times 10^{-4}$ for CIFAR-10, and $\eta = 1 \times 10^{-4}$ for both MNIST and Fashion MNIST.

Random delay. The random delay follows a Poisson distribution with client-specific parameters to reflect heterogeneity. The delay on each worker m follows the Poisson distribution with parameter $2m$ and scaled by $1/2M$, where M is the number of workers and m is the worker index. The expectation of maximum worker delay is one second.

Perturbation. The noise added to the output of each local client follows the Gaussian distribution of each task is $\mathcal{N}(0, 0.01)$ for CIFAR-10, MNIST and Fashion MNIST and $\mathcal{N}(0, 1)$ for Parkinson Disease.

For each task, we run the algorithms sufficiently many epochs and record the training loss. Testing accuracy and wall clock time are recorded at the end of each epoch.

G.3. VAFL for federated deep learning

G.3.1. Training on ModelNet40 dataset

Local embedding structure. We train a convolutional neural network-based model consisting of two parts: the local embedding models and the server model. Each local model is a 7-layer convolutional neural network. The server part is a centralized 3-layer fully connect neural network.

Vertical data allocation. The data we choose is ModelNet40 and we vertically distributed images of the objects in the dataset from 12 angles and assign to each local client. Each local client deals with the data assigned by their local

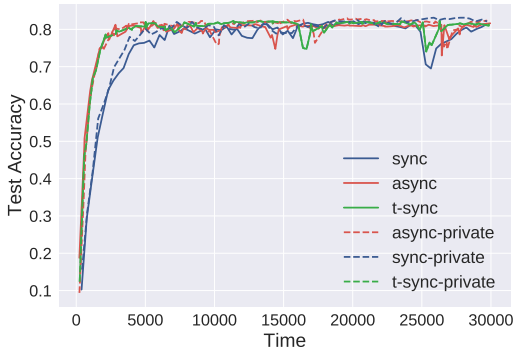

Figure 6. Testing accuracy of VAFL with nonlinear local embedding on *ModelNet40* dataset.

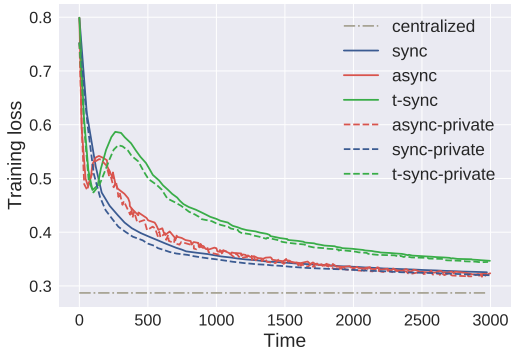

Figure 7. Training loss of VAFL with local LSTM embedding on *MIMIC-III* critical care dataset.

convolutional network and generate a vector whose dimension is 512 as the local output.

Random delay. The random delay follows exponential distribution with client-specific parameters to reflect heterogeneity. For each worker m, the delay follows the exponential distribution with parameter m.

Random perturbation. We use ReLU as the local embedding activation function. We add a random noise on the output of each local embedding convolutional layer. The noises follow the following distributions: $\mathcal{U}(-0.1, 0.1)$ (the first two layers), $\mathcal{U}(-0.01, 0.01)$ (the other convolutional layers except the last layer) and $\mathcal{N}(0, 1)$ (the last convolutional layers).

Server structure. The server then combines the 12 vectors linearly and pass them into the three-layer fully connected neural network and classify into 40 classes. The number of nodes of each layer is 256, 100 and 40.

Learning rate. The stepsize of the local embedding update η_m is 10^{-3} and the server stepsize $\eta_0 = \frac{\eta_m}{M}$ where M is the number of workers.

G.3.2. Training on MIMIC-III dataset

MIMIC is an open dataset comprising deidentified health data associated with 60,000 intensive care unit admissions [16]. The data are allocated into 4 workers having different feature dimensions.

Local embedding structure. The local embedding part is a two layer LSTM models and the server part is a fully connected layer. The first layer is a bidirectional LSTM and the number of units is 16. The second layer is a normal LSTM layer and the number of units is also 16.

Random delay. The random delay follows an exponential distribution with client-specific parameters to reflect heterogeneity. The delay on each worker m follows an exponential distribution with parameter m.

Random perturbation. A random noise following Gaussian distribution $\mathcal{N}(0, 10^{-4})$ is also added on the output of each

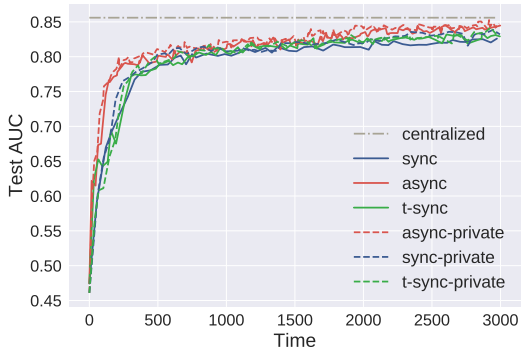

Figure 8. AUC curve of VAFL with local LSTM embedding on *MIMIC-III* clinical care dataset.

local embedding layer.

We have also added simulation results that compare all the algorithms with their private counterparts on both ModelNet40 and MIMIC-III datasets.