Secure Bilevel Asynchronous Vertical Federated Learning with Backward Updating

Qingsong Zhang\textsuperscript{1,2}, Bin Gu\textsuperscript{3,4}, Cheng Deng\textsuperscript{1,*}, and Heng Huang\textsuperscript{3,5,*}

\textsuperscript{1}School of Electronic Engineering, Xidian University, Xi’an 710071, China; \textsuperscript{2}JD Tech, Beijing 100176, China.
\textsuperscript{3}JD Finance America Corporation, Mountain View, CA, USA; \textsuperscript{4}MBZUAI, United Arab Emirates
\textsuperscript{5}Electrical and Computer Engineering, University of Pittsburgh, PA, USA
\{qszhang1995, jsgubin, chdeng.xd, henghuanghh\}@gmail.com

Abstract
Vertical federated learning (VFL) attracts increasing attention due to the emerging demands of multi-party collaborative modeling and concerns of privacy leakage. In the real VFL applications, usually only one or partial parties hold labels, which makes it challenging for all parties to collaboratively learn the model without privacy leakage. Meanwhile, most existing VFL algorithms are trapped in the synchronous computations, which leads to inefficiency in their real-world applications. To address these challenging problems, we propose a novel VFL framework integrated with new backward updating mechanism and bilevel asynchronous parallel architecture (VF\textsuperscript{B2}), under which three new algorithms, including VFB\textsuperscript{2}-SGD, -SVRG, and -SAGA, are proposed. We derive the theoretical results of the convergence rates of these three algorithms under both strongly convex and nonconvex conditions. We also prove the security of VFB\textsuperscript{2} under semi-honest threat models. Extensive experiments on benchmark datasets demonstrate that our algorithms are efficient, scalable and lossless.

Introduction
Federated learning (McMahan et al. 2016; Smith et al. 2017; Kairouz et al. 2019) has emerged as a paradigm for collaborative modeling with privacy-preserving. A line of recent works (McMahan et al. 2016; Smith et al. 2017) focus on the horizontal federated learning, where each party has a subset of samples with complete features. There are also some works (Gascón et al. 2016; Yang et al. 2019b; Dang et al. 2020) studying the vertical federated learning (VFL), where each party holds a disjoint subset of features for all samples. In this paper, we focus on VFL that has attracted much attention due to its wide applications to emerging multi-party collaborative modeling with privacy-preserving.

Currently, there are two mainstream methods for VFL, including homomorphic encryption (HE) based methods and exchanging the raw computational results (ERCR) based methods. The HE based methods (Hardy et al. 2017; Cheng et al. 2019) leverage HE techniques to encrypt the raw data and then use the encrypted data (ciphertext) for training model with privacy-preserving. However, there are two major drawbacks of HE based methods. First, the complexity of homomorphic mathematical operation on ciphertext field is very high, thus HE is extremely time consuming for modeling (Liu, Ng, and Zhang 2015; Liu et al. 2019). Second, approximation is required for HE to support operations of non-linear functions, such as Sigmoid and Logarithmic functions, which inevitably causes loss of the accuracy for various machine learning models using non-linear functions (Kim et al. 2018; Yang et al. 2019a). Thus, the inefficiency and inaccuracy of HE based methods dramatically limit their wide applications to realistic VFL tasks.

ERCR based methods (Zhang et al. 2018; Gu et al. 2020) leverage labels and the raw intermediate computational results transmitted from the other parties to compute stochastic gradients, and thus use distributed stochastic gradient descent (SGD) methods to train VFL models efficiently. Although ECR based methods circumvent aforementioned drawbacks of HE based methods, existing ECR based methods are designed with only considering that all parties have labels, which is not usually the case in real-world VFL tasks. In realistic VFL applications, usually only one or partial parties (denoted as active parties) have the labels, and the other parties (denoted as passive parties) can only provide extra feature data but do not have labels. When these ECR based methods are applied to the real situation with both active and passive parties, the algorithms even cannot guarantee the convergence because only active parties can update the gradient of loss function based on labels but the passive parties cannot, i.e. partial model parameters are not optimized during the training process. Thus, it comes to the crux of designing the proper algorithm for solving real-world VFL tasks with only one or partial parties holding labels.

Moreover, algorithms using synchronous computation (Gong, Fang, and Guo 2016; Zhang et al. 2018) are inefficient when applied to real-world VFL tasks, especially, when computational resources in the VFL system are unbalanced. Therefore, it is desired to design the efficient asynchronous algorithms for real-world VFL tasks. Although there have been several works studying asynchronous VFL algorithms (Hu et al. 2019; Gu et al. 2020b), it is still an open problem to design asynchronous algorithms for solving real-world VFL tasks with only one or partial parties holding labels.

In this paper, we address these challenging problems by proposing a novel framework (VF\textsuperscript{B2}) integrated with the novel backward updating mechanism (BUM) and bilevel
asynchronous parallel architecture (BAPA). Specifically, the BUM enables all parties, rather than only active parties, to collaboratively update the model securely and also makes the final model lossless; the BAPA is designed for efficiently asynchronous backward updating. Considering the advantages of SGD-type algorithms in optimizing machine learning models, we thus propose three new SGD-type algorithms, i.e., VFB²-SGD, -SVRG and -SAGA, under that framework. We summarize the contributions of this paper as follows.

- We are the first to propose the novel backward updating mechanism for ERCR based VFL algorithms, which enables all parties, rather than only parties holding labels, to collaboratively learn the model with privacy-preserving and without hampering the accuracy of final model.
- We design a bilevel asynchronous parallel architecture that enables all parties asynchronously update the model through backward updating, which is efficient and scalable.
- We propose three new algorithms for VFL, including VFB²-SGD, -SVRG, and -SAGA under VFB². Moreover, we theoretically prove their convergence rates for both strongly convex and nonconvex problems.

**Notations.** $\bar{w}$ denotes the inconsistent read of $w$, $\bar{w}$ denotes $w$ to compute local stochastic gradient of loss function for collaborators, which may state due to communication delay. $\psi(t)$ is the corresponding party performing the $t$-th global iteration. Given a finite set $S$, $|S|$ denotes its cardinality.

**Problem Formulation**

Given a training set $\{x_i, y_i\}_{i=1}^n$, where $y_i \in \{-1, +1\}$ for binary classification task or $y_i \in \mathbb{R}$ for regression problem and $x_i \in \mathbb{R}^d$, we consider the model in a linear form of $w^T x$, where $w \in \mathbb{R}^d$ corresponds to the model parameters. For VFL, $x_i$ is vertically distributed among $q \geq 2$ parties, i.e., $x_i = [(x_i)_{q_1}; \cdots; (x_i)_{q_q}]$, where $(x_i)_{q_i} \in \mathbb{R}^{d_i}$ is stored on the $t$-th party and $\sum_{i=1}^q d_i = d$. Similarly, there is $w = [w_{q_1}; \cdots; w_{q_q}]$. Particularly, we focus on the following regularized empirical risk minimization problem.

$$
\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{n} \sum_{i=1}^n \mathcal{L}(w^T x_i, y_i) + \lambda \sum_{i=1}^q g(w_{q_i}), \quad (P)
$$

where $w^T x_i = \sum_{i=1}^q w_{q_i}^T (x_i)_{q_i}$, $\mathcal{L}$ denotes the loss function, $\sum_{i=1}^q g(w_{q_i})$ is the regularization term, and $f_i: \mathbb{R}^d \to \mathbb{R}$ is smooth and possibly nonconvex. Examples of problem P include models for binary classification tasks (Conroy and Sajda 2012; Wang et al. 2017) and models for regression tasks (Shen et al. 2013; Wang et al. 2019).

In this paper, we introduce two types of parties: **active party** and **passive party**, where the former denotes data provider holding labels while the latter does not. Particularly, in our problem setting, there are $m$ (1 ≤ $m$ ≤ $q$) active parties. Each active party can play the role of dominator in model updating by actively launching updates. All parties, including both active and passive parties, passively launching updates play the role of collaborator. To guarantee the model security, only active parties know the form of the loss function. Moreover, we assume that the labels can be shared by all parties finally. Note that this does not obey our intention that only active parties hold the labels before training. The problem studied in this paper is stated as follows:

**Given:** Vertically partitioned data $\{x_{q_i}\}_{i=1}^q$ stored in $q$ parties and the labels only held by active parties.

**Learn:** A machine learning model $M$ collaboratively learned by both active and passive parties without leaking privacy.

**Lossless Constraint:** The accuracy of $M$ must be comparable to that of model $M'$ learned under non-federated learning.

**Algorithm 1** Safe algorithm of obtaining $w^T x_i$.

**Input:** $\{w_{q_i}\}_{i=1}^q$ and $\{(x_i)_{q_i}\}_{i=1}^q$ allocating at each party, index $i$.

**Do this in parallel**

1: for $\ell = 1, \cdots, q$ do
2: Generate a random number $\delta_{\ell}$ and calculate $w_{q_{\ell}}^T (x_i)_{q_{\ell}} + \delta_{\ell}$.
3: end for
4: Obtain $\xi_1 = \sum_{\ell=1}^q (w_{q_{\ell}}^T (x_i)_{q_{\ell}} + \delta_{\ell})$ through tree structure $T_1$.
5: Obtain $\xi_2 = \sum_{\ell=1}^q \delta_{\ell}$ through totally different tree structure $T_2 \neq T_1$.

**Output:** $w^T x_i = \xi_1 - \xi_2$

**VFB² Framework**

In this section, we propose the novel VFB² framework. VFB² is composed of three components and its systemic structure is illustrated in Fig. 1a. The details of these components are presented in the following.

The key of designing the proper algorithm for solving real-world VFL tasks with both active and passive parties is to make the passive parties utilize the label information for model training. However, it is challenging to achieve this because direct using the labels hold by active parties leads to privacy leakage of the labels without training. To address this challenging problem, we design the BUM with painstaking.

**Backward Updating Mechanism:** The key idea of BUM is to make passive parties indirectly use labels to compute stochastic gradient without directly accessing the raw label data. Specifically, the BUM embeds label $y_i$ into an intermediate value $\vartheta := \nabla \mathcal{L}(w^T x, y)$. Then $\vartheta$ and $i$ are distributed backward to the other parties. Consequently, the passive parties can also compute the stochastic gradient and update the model by using the received $\vartheta$ and $i$ (please refer to Algorithms 2 and 3 for details). Fig. 1b depicts the case where $\vartheta$ is distributed from party 1 to the rest parties. In this case, all parties, rather than only active parties, can collaboratively learn the model without privacy leakage.

For VFL algorithms with BUM, dominated updates in different active parties are performed in distributed-memory parallel, while collaborative updates within a party are performed in shared-memory parallel. The difference of parallelism fashion leads to the challenge of developing a new
parallel architecture instead of just directly adopting the existing asynchronous parallel architecture for VFL. To tackle this challenge, we elaborately design a novel BAPA.

**Bilevel Asynchronous Parallel Architecture**: The BAPA includes two levels of parallel architectures, where the upper level denotes the inner-party parallel and the lower one is the intra-party parallel. More specifically, the inner-party parallel denotes distributed-memory parallel between active parties, which enables all active parties to asynchronously launch dominated updates; while the intra-party one denotes the shared-memory parallel of collaborative updates within each party, which enables multiple threads within a specific party to asynchronously perform the collaborative updates. Fig. 1b illustrates the BAPA with m active parties.

To utilize feature data provided by other parties, a party need obtain \( w^T x_i = \sum_{\ell=1}^q w^T_{\ell, i} \). Many recent works achieved this by aggregating the local intermediate computational results securely (Hu et al. 2019; Gu et al. 2020a). In this paper, we use the efficient tree-structured communication scheme (Zhang et al. 2018) for secure aggregation, whose security was proved in (Gu et al. 2020a).

**Secure Aggregation Strategy**: The details are summarized in Algorithm 1. Specifically, at step 2, \( w^T_{\ell, i} \) is computed locally on the \( \ell \)-th party to prevent the direct leakage of \( w_{\ell, i} \) and \( (x_i)_G \). Especially, a random number \( \delta_\ell \) is added to \( w^T_{\ell, i} \) to mask the value of \( w^T_{\ell, i} \), which can enhance the security during aggregation process. At steps 4 and 5, \( \xi_1 \) and \( \xi_2 \) are aggregated through tree structures \( T_1 \) and \( T_2 \), respectively. Note that \( T_2 \) is totally different from \( T_1 \) that can prevent the random value being removed under threat model 1 (defined in section ). Finally, value of \( w^T x_i = \sum_{\ell=1}^q w^T_{\ell, i} \) is recovered by removing term \( \sum_{\ell=1}^q \delta_\ell \) from \( \sum_{\ell=1}^q (w^T_{\ell, i} + \delta_\ell) \) at the output step. Using such aggregation strategy, \( (x_i)_G \) and \( w_{G, i} \) are prevented from leaking during the aggregation.

**Secure Bilevel Asynchronous VFL Algorithms with Backward Updating**

SGD (Bottou 2010) is a popular method for learning machine learning (ML) models. However, it has a poor convergence rate due to the intrinsic variance of stochastic gradient. Thus, many popular variance reduction techniques have been proposed, including the SVRG, SAGA, SPIDER (Johnson and Zhang 2013; Defazio, Bach, and Lacoste-Julien 2014; Wang et al. 2019) and their applications to other problems (Huang, Chen, and Huang 2019; Hu et al. 2020; Zhang et al. 2020; Yang et al. 2020; Li et al. 2020; Wei et al. 2019). In this section we raise three SGD-type algorithms, *i.e.* the SGD, SVRG and SAGA, which are the most popular ones among SGD-type methods for the appealing performance in practice. We summarize the detailed steps of VFB^2-SGD in Algorithms 2 and 3. For VFB^2-SVRG and -SAGA, one just needs to replace the update rule with corresponding one.

As shown in Algorithm 2, at each dominated update, the dominator (an active party) calculates \( \vartheta \) and then distributes \( \vartheta \) together with \( i \) to the collaborators (the rest \( q - 1 \) parties). As shown in Algorithm 3, for party \( \ell \), once it has received the \( \vartheta \) and \( i \), it will launch a new collaborative update asynchronously. As for backtracking, it computes the local
Assumption 3. Bounded Delay: Following conditions hold:

\[ \text{delay} \leq \tau_2. \]

Given \( \tilde{w} \) as the inconsistent read of \( w \), which is used to compute the stochastic gradient in dominated updates, following the analysis in (Gu et al. 2020b), we have

\[ \tilde{w}_t - w_t = \gamma \sum_{u \in D(t)} U_{\psi(u)} \tilde{u}^\psi(u), \]

where \( D(t) = \{ t - 1, \ldots, t - \tau_0 \} \) is a subset of non-overlapped previous iterations with \( \tau_0 \leq \tau_2 \). Given \( \tilde{w} \) as the parameter used to compute the \( \nabla \tilde{g} \) in collaborative updates, which is the steal state of \( \tilde{w} \) due to the communication delay between the specific dominator and its corresponding collaborators. Then, following the analyses in (Huo and Huang 2017), there is

\[ \tilde{w}_t = \tilde{w}_{t-\tau_0} + \gamma \sum_{t' \in D'(t)} U_{\psi(t')} \tilde{u}^\psi(t'), \]

where \( D'(t) = \{ t - 1, \ldots, t - \tau_0 \} \) is a subset of previous iterations performed during the communication and \( \tau_0 \leq \tau_2 \).

Convergence Analysis for Strongly Convex Problem

Assumption 4. Each function \( f_i, i = 1, \ldots, n \), is \( \mu \)-strongly convex, i.e., \( \forall w, w' \in \mathbb{R}^d \) there exists a \( \mu > 0 \) such that

\[ f_i(w) \geq f_i(w') + \langle \nabla f_i(w'), w - w' \rangle + \frac{\mu}{2} ||w - w'||^2. \]

For strongly convex problem, we introduce notation \( k(t) \) that denotes a minimum set of successive iterations fully visiting all coordinates from global iteration number \( t \). Note that this is necessary for the asynchronous convergence analyses of the global model. Moreover, we assume that the size of \( k(t) \) is upper bounded by \( \eta_1 \), i.e., \( |k(t)| \leq \eta_1 \). Based on \( k(t) \), we introduce the epoch number \( v(t) \) as follow.

Definition 1. Let \( P(t) \) be a partition of \( \{0, 1, \ldots, t - \delta' \} \), where \( \delta' \geq 0 \). For any \( \kappa \subseteq P(t) \) we have that there exists \( t' \leq t \) such that \( k(t') = k(t) \), and \( \kappa \subseteq P(t) \) such that \( k(0) = \kappa_1 \), i.e., \( v(t) \) is defined as the maximum cardinality of \( P(t) \).

Given the definition of epoch number \( v(t) \), we have the following theoretical results for \( \mu \)-strongly convex problem.

Theorem 1. Under Assumptions 1-3 and 4, to achieve the accuracy \( \epsilon \) of problem \( P \) for \( \text{VFB2-SGD} \), i.e.,

\[ E(f(w_t) - f(w^*)) \leq \epsilon, \]

let \( \gamma \leq \frac{\epsilon^{1/3}}{\tau (G^2 L^2)^{1/3}}, \)

where \( \tau \leq \min \{ \epsilon^{-1/3} \left( \frac{G^2 L^2}{\mu} \right)^{1/3}, \frac{1}{\epsilon} \} \), the epoch number \( v(t) \) should satisfy

\[ v(t) \geq \frac{44(G^2 L^2)^{1/3}}{\epsilon \mu^{1/3}} \log \left( \frac{2(f(w_0)) - f(\tilde{w})}{\epsilon} \right), \]

where \( L_* = \max \{ L_{\ell} \}_{\ell=1}^q, \) \( \tau = \tau_1, \tau_2, \eta_1 \), \( \tilde{w} \) and \( w^* \) denote the initial point and optimal point, respectively.

Theorem 2. Under Assumptions 1-3 and 4, to achieve the accuracy \( \epsilon \) of problem \( P \) for \( \text{VFB2-SVRG} \), let \( C = (L^2 \gamma + \frac{L^2}{2}) \) and \( \rho = \frac{2\gamma}{\mu} - \frac{16L_2 \eta_1 C}{\mu} \), we can carefully choose \( \gamma \) such that

\[ 1) \ 1 - 2L^2 \gamma^2 \tau > 0; \ 2) \ \rho > 0; \ 3) \ \frac{8L_0^2 \gamma^{1/2} C}{\mu \rho} \leq 0.05; \]

4) \[ L^2 \gamma^2 \tau^{3/2} (28C + 5\gamma) \frac{36G}{\rho} (1 - 2L^2 \gamma^2 \tau) \leq \epsilon, \]

where \( L^0 \gamma \) and \( \rho \) are constants.
where \( v(t) \) should satisfy \( v(t) \geq \frac{\log(0.25)}{\log(1 - \rho)} \) and the outer loop number \( S \) should satisfy \( S \geq \frac{1}{\log(2)} \frac{\log(2)}{\epsilon} \).

**Theorem 3.** Under Assumptions 1-3 and 4, to achieve the accuracy \( \epsilon \) of problem \( P \) for VFB\(^2\)-SAGA, let \( c_0 = (2\gamma^3 + \beta^2 + \beta) \), \( c_1 = 2L^2 \rho(2\gamma^3 + \beta), c_2 = 4(L^2 \gamma^3 + \beta) \), and \( \rho \in (1 - \frac{1}{n}, 1) \), we can choose \( \gamma \) such that

1. \( 1 - 72L^2 \gamma^2 \tau > 0 \);
2. \( 0 < 1 - \frac{\gamma \mu}{4} < 1 \);
3. \( \frac{4c_0}{\gamma(1 - \rho)} \frac{\rho}{4} - 2c_1 - c_2 \) \leq \( \frac{\epsilon}{2} \);
4. \( -\frac{\gamma \mu^2}{4} + 2c_1 + c_2 \left( 1 + \frac{1 - 1 - \frac{1}{\rho}}{\rho} \right) \) \leq 0;
5. \( -\frac{\gamma \mu^2}{4} + c_2 + c_1 \left( 2 + \frac{1 - 1 - \frac{1}{\rho}}{\rho} \right) \leq 0 (8) \)

the epoch number \( v(t) \) should satisfy \( v(t) \geq \frac{1}{\log(2)} \frac{\log(2)}{\epsilon} \).

**Remark 1.** For strongly convex problems, given the assumptions and parameters in corresponding theorems, the convergence rate of VFB\(^2\)-SGD is \( O\left(\frac{1}{n^2 \log(\frac{1}{\epsilon})}\right) \), and those of VFB\(^2\)-SVRG and VFB\(^2\)-SAGA are \( O\left(\log(\frac{1}{\epsilon})\right) \).

**Convergence Analysis for Nonconvex Problem**

**Assumption 5.** Nonconvex function \( f(w) \) is bounded below; \( f^* := \inf_{w \in \mathbb{R}^d} f(w) > -\infty \).

Assumption 5 guarantees the feasibility of nonconvex problem (P). For nonconvex problem, we introduce the notation \( K'\) that denotes a set of iterations fully visiting all coordinates, i.e., \( K'(t) = \{ t + t_1, \cdots, t + t_q - 1 : \psi(t, t + t_1, \cdots, t + t_q - 1) = \{ 1, \cdots, q \} \} \), where the \( t \)th global iteration denotes a dominated update. Moreover, these iterations are performed respectively on a dominator \( q \) and \( q - 1 \) different collaborators receiving \( \psi \) calculated at the \( t \)th global iteration. Moreover, we assume that \( K'(t) \) can be completed in \( n^2 \) global iterations, i.e., for all \( t' \in A(t) \), there is \( n^2 \) max\{\( w_i \in K'(t') \) \} \( t' \). Note that, different from \( K(t) \), there is \( |K'(t)| = q \) and the definition of \( K'(t) \) does not emphasize on “successive iterations” due to the difference of analysis techniques between strongly convex and nonconvex problems. Based on \( K'(t) \), we introduce the epoch number \( v'(t) \) as follow.

**Definition 2.** \( A(t) \) denotes a set of global iterations, where for all \( t' \in A(t) \) there is \( t' \)-th global iteration denoting a dominated update and \( \bigcup_{t' \in A(t)} K'(t') = \{ 0, 1, \cdots, t \} \). The epoch number \( v'(t) \) is defined as \( |A(t)| \).

Give the definition of epoch number \( v'(t) \), we have the following theoretical results for nonconvex problem.

**Theorem 4.** Under Assumptions 1-3 and 5, to achieve the \( \epsilon \)-first-order stationary point of problem \( P \), i.e., \( \mathbb{E}[\|\nabla f(w)\|] \leq \epsilon \) for stochastic variable \( w \), for VFB\(^2\)-SGD, let \( \gamma = \frac{5\mu}{\rho} \), \( \tau = \frac{1}{\rho} \), \( f(w^0) \) is the initial function value and \( f^* \) is defined in Eq. (9).

\[
T \geq \frac{\epsilon}{\rho(1 + \frac{2}{\rho})} \left( \frac{\rho}{4} - 2c_1 - c_2 \right) - \frac{\epsilon}{\rho(1 + \frac{2}{\rho})} \left( \frac{\rho}{4} - 2c_1 - c_2 \right).
\]

**Theorem 5.** Under Assumptions 1-3 and 5, to solve problem \( P \) with VFB\(^2\)-SVRG, let \( \gamma = \frac{\mu}{\rho} \), \( \rho \) is a small value independent of \( n \), \( 0 < \alpha \leq 1 \), if epoch number \( N \) in an outer loop satisfies \( N \leq \left| \frac{n^2}{\sigma \rho} \right| \), and \( \tau < \min\{\frac{1}{\rho} \}, \) there is

\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_{t+1})\|^2] \leq \frac{L_n \sigma^{2} \mathbb{E}[f(w_{t+1}) - f(w^*)]}{T},
\]

where \( T \) is the total number of epochs, \( t_0 \) is the start iteration of epoch \( t, \) and \( \sigma \) is a small value independent of \( n \).

**Theorem 6.** Under Assumptions 1-3 and 5, to solve problem \( P \) with VFB\(^2\)-SAGA, let \( \gamma = \frac{\mu}{\rho} \), \( 0 < \mu_0 < \frac{1}{20} \), \( 0 < \alpha \leq 1 \), if epoch number \( T \) satisfies \( T \leq \left| \frac{n^2}{4 \sigma \rho} \right| \), and \( \tau < \min\{\frac{1}{\rho} \}, \) there is

\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(w_{t+1})\|^2] \leq \frac{L_n \sigma^{2} \mathbb{E}[f(w_{t+1}) - f(w^*)]}{T},
\]

**Remark 2.** For nonconvex problems, given conditions in the theorems, the convergence rate of VFB\(^2\)-SGD is \( O\left(1/\sqrt{T}\right) \), and those of VFB\(^2\)-SVRG and VFB\(^2\)-SAGA are \( O\left(1/T\right) \).

**Security Analysis**

We discuss the data security and model security of VFB\(^2\) under two semi-honest threat models commonly used in security analysis (Cheng et al. 2019; Xu et al. 2019; Gu et al. 2020a). Specially, these two threat models have different threat abilities, where threat model 2 allows collusions between parties while threat model 1 does not.

- **Honest-but-curious** (threat model 1): All workers will follow the algorithm to perform the correct computations. However, they may use their own retained records of the intermediate computation result to infer other worker’s data and model.
- **Honest-but-colluding** (threat model 2): All workers will follow the algorithm to perform the correct computations. However, some workers may collude to infer other worker’s data and model by sharing their retained records of the intermediate computation result.
Theorem 7. Under two semi-honest threat models, VF can simulate the environment with multiple machines (or parties) to refer to the equation (12) with only one being known, there are infinite different solutions to this equation.

The proof of lemma 1 is shown in the arXiv version. Based on lemma 1, we obtain the following theorem.

Theorem 7. Under two semi-honest threat models, VF2 can prevent the inference attack.

Feature and model security: During the aggregation, the value of \( o_i = w_{xG}^T(x_i)G + \delta_i \) is masked by \( \delta_i \) and just the value of \( w_{xG}^T(x_i)G \) is transmitted. Under threat model 1, one even can not access the true value of \( o_i \), let alone using relation \( o_i = w_{xG}^T(x_i)G \) to refer \( w_{xG}^T \) and \( (x_i)G \). Under threat model 2, the random value \( \delta_i \) has risk of being removed from term \( w_{xG}^T(x_i)G + \delta_i \) by colluding with other parties. Applying lemma 1 to this circumstance, and we have that even if the random value is removed it is still impossible to exactly refer \( w_{xG}^T \) and \( (x_i)G \). Thus, the aggregation process can prevent inference attack under two semi-honest threat models.

Label security: When analyze the security of label, we do not consider the collusion between active parties and passive parties, which will make preventing labels from leaking meaningless. In the backward updating process, if a passive party \( \ell \) wants to infer \( y_i \) through the received \( \vartheta \), it must solve the equation \( \vartheta = \frac{\partial L(\hat{\omega}^T x_i, y_i)}{\partial (\hat{\omega}^T x_i)} \). However, only \( \vartheta \) is known to the active party \( \ell \). Thus, following from lemma 1, we have that it is impossible to exactly infer the labels. Moreover, the collusion between passive parties has no threats to the security of labels. Therefore, the backward updating can prevent inference attack under two semi-honest threat models.

From above analyses, we have that the feature security, label security and model security are guaranteed in VF2.

**Experiments**

In this section, extensive experiments are conducted to demonstrate the efficiency, scalability and losslessness of our algorithms. More experiments are presented in the arXiv version.

**Experiment Settings:** All experiments are implemented on a machine with four sockets, and each socket has 12 cores. To simulate the environment with multiple machines (or parties), we arrange an extra thread for each party to schedule its \( k \) threads and support communication with (threads of) the other parties. We use MPI to implement the communication scheme. The data are partitioned vertically and randomly into \( q \) non-overlapped parts with nearly equal number of features. The number of threads within each parties, i.e. \( k \), is set as \( m \). We use the training dataset or randomly select 80% samples as the training data, and the testing dataset or the rest as the testing data. An optimal learning rate \( \gamma \) is chosen from \( \{5e^{-1}, 1e^{-1}, 5e^{-2}, 1e^{-2}, \ldots \} \) with regularization coefficient \( \lambda = 1e^{-4} \) for all experiments.

**Datasets:** We use four classification datasets summarized in Table 1 for evaluation. Especially, \( D_1 \) (UCICreditCard) and \( D_2 \) (GiveMeSomeCredit) are the real financial datasets from the Kaggle website\(^*\), which can be used to demonstrate the ability to address real-world tasks; \( D_3 \) (news20) and \( D_4 \) (webslap) are the large-scale ones from the LIB-SVM (Chang and Lin 2011) website\(^1\). Note that we apply one-hot encoding to categorical features of \( D_1 \) and \( D_2 \), thus the number of features become 90 and 92, respectively.

**Problems:** We consider \( \ell^2 \)-norm regularized logistic regression problem for \( \mu \)-strong convex case

\[
\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i w^T x_i}) + \frac{\lambda}{2} ||w||^2, \quad (13)
\]

and the nonconvex logistic regression problem

\[
\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i w^T x_i}) + \frac{\lambda}{2} \sum_{i=1}^{d} \frac{w_i^2}{1 + w_i^2}.
\]

**Evaluations of Asynchronous Efficiency and Scalability**

To demonstrate the asynchronous efficiency, we introduce the synchronous counterparts of our algorithms (i.e., synchronous VFL algorithms with BUM, denoted as VFB) for comparison. When implementing the synchronous algorithms, there is a synthetic straggler party which may be 30% to 50% slower than the faster party to simulate the real application scenario with unbalanced computational resource.

**Asynchronous Efficiency:** In these experiments, we set \( q = 8, m = 3 \) and fix the \( \gamma \) for algorithms with a same SGD-type but in different parallel fashions. As shown in Figs. 3 and 4, the loss v.s. run time curves demonstrate that our algorithms consistently outperform their synchronous counterparts regarding the efficiency.

Moreover, from the perspective of loss v.s. epoch number, we have that algorithms based on SVRG and SAGA have better convergence rate than that of SGD-based algorithms which is consistent to the theoretical results.

**Asynchronous Scalability:** We also consider the asynchronous speedup scalability in terms of the number of total parties \( q \). Given a fixed \( m \), \( q \)-parties speedup is defined as

\[
q\text{-parties speedup} = \frac{\text{Run time of using 1 party}}{\text{Run time of using } q \text{ parties}}, \quad (14)
\]

\(^*\)https://www.kaggle.com/datasets

\(^1\)https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/
To demonstrate the losslessness of our algorithms, we compare VFB\(^2\)-SVRG with its non-federated (NonF) counterpart (all data are integrated together for modeling) and ERCR based algorithm but without BUM, i.e., AFSVRG-VP proposed in (Gu et al. 2020b). Especially, AFSVRG-VP also uses distributed SGD method but can not optimize the parameters corresponding to passive parties due to lacking labels. When implementing AFSVRG-VP, we assume that only half parties have labels, i.e., parameters corresponding to the features held by the other parties are not optimized. Each comparison is repeated 10 times with \( m = 3, q = 8 \), and a same stop criterion, e.g., \( 1e^{-5} \) for \( D_1 \). As shown in Table 2, the accuracy of our algorithms are the same with those of NonF algorithms and are much better than those of AFSVRG-VP, which are consistent to our claims.

### Evaluation of Losslessness

To demonstrate the losslessness of our algorithms, we compare VFB\(^2\)-SVRG with its non-federated (NonF) counterpart (all data are integrated together for modeling) and ERCR based algorithm but without BUM, i.e., AFSVRG-VP proposed in (Gu et al. 2020b). Especially, AFSVRG-VP also uses distributed SGD method but can not optimize the parameters corresponding to passive parties due to lacking labels. When implementing AFSVRG-VP, we assume that only half parties have labels, i.e., parameters corresponding to the features held by the other parties are not optimized. Each comparison is repeated 10 times with \( m = 3, q = 8 \), and a same stop criterion, e.g., \( 1e^{-5} \) for \( D_1 \). As shown in Table 2, the accuracy of our algorithms are the same with those of NonF algorithms and are much better than those of AFSVRG-VP, which are consistent to our claims.

### Conclusion

In this paper, we proposed a novel backward updating mechanism for the real VFL system where only one or partial parties have labels for training models. Our new algorithms enable all parties, rather than only active parties, to collaboratively update the model and also guarantee the algorithm convergence, which was not held in other recently proposed ERCR based VFL methods under the real-world setting. Moreover, we proposed a bilevel asynchronous parallel architecture to make ERCR based algorithms with backward updating more efficient in real-world tasks. Three practical SGD-type of algorithms were also proposed with theoretical guarantee.
Acknowledgments

Q.S. Zhang and C. Deng were supported in part by the National Natural Science Foundation of China under Grant 62071361, the National Key R&D Program of China under Grant 2017YFE0104100, and the China Research Project under Grant 6141B07270429.

References


Adversarial Fine-Grained Composition Learning for Unseen 
International Conference on Computer Vision, 3741–3749.

Xu, R.; Baracaldo, N.; Zhou, Y.; Anwar, A.; and Ludwig, 
H. 2019. Hybridalpha: An efficient approach for privacy-
preserving federated learning. In Proceedings of the 12th 
ACM Workshop on Artificial Intelligence and Security, 13–23.

A Quasi-Newton Method Based Vertical Federated Learn-
ing Framework for Logistic Regression. arXiv preprint 

Yang, M.; Deng, C.; Yan, J.; Liu, X.; and Tao, D. 2020a. 
Learning Unseen Concepts via Hierarchical Decomposition 
and Composition. In Proceedings of the IEEE/CVF Confer-
ence on Computer Vision and Pattern Recognition, 10248– 
10256.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019b. Federated 
machine learning: Concept and applications. ACM Trans-
actions on Intelligent Systems and Technology (TIST) 10(2): 
12.

Yang, X.; Deng, C.; Wei, K.; Yan, J.; and Liu, W. 2020b. 
Adversarial Learning for Robust Deep Clustering. Advances 
in Neural Information Processing Systems 33.

Zhang, G.-D.; Zhao, S.-Y.; Gao, H.; and Li, W.-J. 2018. 
Feature-Distributed SVRG for High-Dimensional Linear 

Zhang, Q.; Huang, F.; Deng, C.; and Huang, H. 2020. 
Faster Stochastic Quasi-Newton Methods. arXiv preprint 