Machine Learning (2024) 113:6413–6451
https://doi.org/10.1007/s10994-024-06541-y

Secure and fast asynchronous Vertical Federated Learning via cascaded hybrid optimization

Ganyu Wang[1] · Qingsong Zhang[2] · Xiang Li[1] · Boyu Wang[1] · Bin Gu[3] · Charles X. Ling[1]

Received: 18 September 2023 / Revised: 7 January 2024 / Accepted: 5 March 2024 /
Published online: 27 June 2024

Abstract
Vertical Federated Learning (VFL) is gaining increasing attention due to its ability to enable multiple parties to collaboratively train a privacy-preserving model using vertically partitioned data. Recent research has highlighted the advantages of using zeroth-order optimization (ZOO) in developing practical VFL algorithms. However, a significant drawback of ZOO-based VFL is its slow convergence rate, which limits its applicability in handling large modern models. To address this issue, we propose a cascaded hybrid optimization method for VFL. In this method, the downstream models (clients) are trained using ZOO to ensure privacy and prevent the sharing of internal information. Simultaneously, the upstream model (server) is updated locally using first-order optimization, which significantly improves the convergence rate. This approach allows for the training of large models without compromising privacy and security. We theoretically prove that our VFL method achieves faster convergence compared to ZOO-based VFL because the convergence rate of our framework is not limited by the size of the server model, making it effective for training large models. Extensive experiments demonstrate that our method achieves faster convergence than ZOO-based VFL while maintaining an equivalent level of privacy protection. Additionally, we demonstrate the feasibility of training large models using our method.

Keywords Vertical Federated Learning · Zeroth order optimization · Computation-communication efficiency · Privacy

1 Introduction

Data availability is essential for machine learning, however, privacy concerns often prevent the direct sharing of data among different parties. Federated learning (FL) addresses this issue by facilitating collaborative model training without sharing private data. This approach allows multiple parties to leverage their data while adhering to privacy protection measures and government regulations, such as the General Data Protection Regulation (GDPR) (Commission, 2016).

Editor: Bo Han.

Extended author information available on the last page of the article

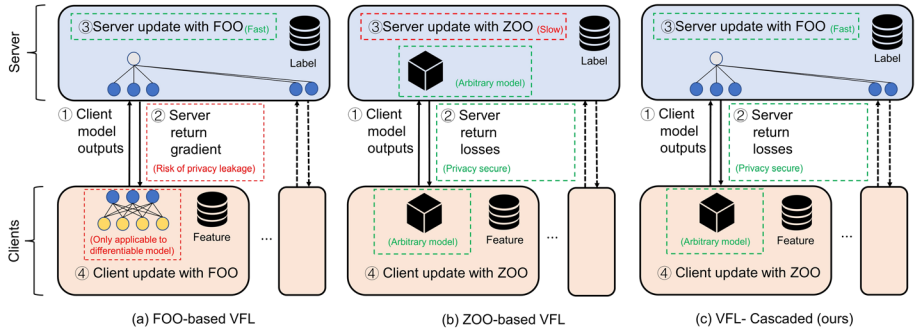


Fig. 1 The intuition of our VFL framework

Federated Learning (FL) algorithms have evolved into two mainstream subtypes, including Horizontal Federated Learning (HFL) (McMahan et al., 2017; Li et al., 2020, 2021; Karimireddy et al., 2020; Mishchenko et al., 2022; Shi et al., 2021; Casado et al., 2023; Badar et al., 2023; Ahmad et al., 2023; Li et al., 2022; Sabater et al., 2022) and Vertical Federated Learning (VFL) (Li et al., 2020; Vepakomma et al., 2018; Chen et al., 2020; Yang et al., 2019; Hu et al., 2019; Wei et al., 2022; Gu et al., 2021). HFL involves clients holding a subset of data points with a full feature set (horizontally distributed), while VFL involves clients holding all data points but with a non-intersecting subset of features (vertically distributed).

We focus on VFL, which is applicable to practical learning scenarios in various industries, such as hospitals, banks, and insurance companies. For example, a government agency (server) collaborates with multiple banks (clients) to develop a model for estimating customers' credit scores (Wei et al., 2022), where each bank holds a distinct set of customer features. In VFL, the client trains a feature extraction model that maps its local data sample to embeddings. The server then collects the embeddings from all clients and uses them as input for the server model to make a prediction.

To build a practical VFL, it is essential to meet the following fundamental requirements: model applicability (Castiglia et al., 2022; Makhija et al., 2022; Zhang et al., 2021), privacy security (Zhou et al., 2020; Hardy et al., 2017; Fang et al., 2021), computational efficiency (Chen et al., 2020; Hu et al., 2019; Zhang et al., 2021), and communication efficiency (Zhang et al., 2021; Castiglia et al., 2022; Wang et al., 2022).

In a typical VFL framework optimized with FOO (Chen et al., 2020; Vepakomma et al., 2018),nas illustrated in Fig. 1a, both the server and clients utilize FOO to optimize the model, which is fast. However, sharing the gradient with the client poses a serious risk of privacy leakage (Fu et al., 2022; Fredrikson et al., 2015; He et al., 2016; Zhao et al., 2020), and the framework is only applicable to differentiable models.

A recent study (Zhang et al., 2021) found that applying ZOO on VFL, as depicted in Fig. 1b, offers several advantages in building practical VFL. Firstly, it enhances model applicability by eliminating the requirement for an explicit gradient to update the model. Secondly, it improves privacy security by transmitting black-box information (losses) to the client instead of internal information (gradients). Besides, the client retains the perturbation direction, preventing third parties from obtaining the gradient. As a result, both the server and client can maintain the confidentiality of gradient

information during training. However, relying solely on ZOO for model optimization can lead to slow convergence, especially when dealing with large models.

Both frameworks mentioned above do not meet the requirement of practical VFL. Although FOO converges rapidly and dependably, the privacy risk associated with transmitting the gradient is a significant drawback. On the other hand, ZOO provides high model applicability and privacy security but suffers from a slow convergence problem.

Then, it comes to the question: *How to improve the convergence speed while preserving the advantages of ZOO to make a practical VFL?*

In this paper, we provide a solution to this problem by proposing a cascaded hybrid optimization method in the asynchronous VFL which maximizes the benefits of both optimization methods.

As depicted in Fig. 1c, we utilized distinct optimization methods for the upstream (server) and downstream (client) of the global model in a cascaded manner. This approach ensures privacy preservation, as the downstream models update with ZOO, which guarantees that no gradient is transmitted through the network. Additionally, the upstream model is updated with FOO locally, which converges fast and does not compromise privacy.

Our contributions can be summarized as follows:

- We propose a practical asynchronous VFL framework that cascades two different optimization methods (FOO &ZOO), where the advantages of both optimization methods are maximized. Our VFL framework satisfies the fundamental requirements of model applicability, privacy security, computational efficiency, and communication efficiency to a significant degree.
- We theoretically prove that the convergence of our VFL framework is faster than the ZOO-based VFL by demonstrating that the convergence is solely limited by the size of the client's parameters. Additionally, our VFL framework can feasibly train a large parameterized model with the majority part on the server.
- We conduct extensive experiments on the Multi-Layer Perception (MLP), Convolutional Neural Network (CNN), and Large Language Model (LLM) to demonstrate the privacy and applicability of our framework in the latest deep learning tasks.

Justification of the Application Scenario: In our VFL setting, the server uses a larger model compared with the clients. We provide our justification for this application scenario below.

In VFL, the server is typically the initiator and primary beneficiary of the model training process. The client, on the other hand, acts as a follower and only provides the embedding of their local features without disclosing the raw data (Wei et al., 2022). Besides, the server usually possesses more computational resources than the clients, making it more suitable for training large models. Therefore, using a larger model on the server side can lead to better data predictions and reduce the computational burden for all participants in the VFL, making it a more preferable and economical option.

2 Related work

There are several basic metrics to consider when developing a VFL framework:

Model Applicability dictates the VFL framework can fit heterogeneous models. The heterogeneity of the model mainly determines whether the model is differentiable.

For example, most of the VFL approaches explicitly apply gradient (Vepakomma et al., 2018; Chen et al., 2020), which forces each party to use a differentiable model. However, this approach may not always be practical, especially when the participants have non-differentiable model architectures. In such cases, when the gradient is not available, the main solution is to apply proximal-term (Castiglia et al., 2022) or to use ZOO (Zhang et al., 2021).

Privacy is a critical consideration for any VFL algorithm. In VFL, there are two types of private data: the features held by the clients and the labels held by the server. Depending on the target of the attack, privacy inference attacks in VFL can be classified as feature inference attacks (Luo et al., 2021; Jin et al., 2021; Zhu et al., 2019; Fredrikson et al., 2015; Weng et al., 2020) or label inference attacks (Fu et al., 2022; Sun et al., 2022; Zhu et al., 2019; Zhao et al., 2020; Jin et al., 2021).

The mainstream privacy protection scheme is applying privacy computing on VFL. For example, Liu et al. (2020) and Hardy et al. (2017) have applied homomorphic encryption (HE) on the transmission data, where the participant in the VFL framework sends the ciphertext instead of plain text through the network. Other works have used differential privacy (DP) (Shokri & Shmatikov, 2015; Ranbaduge & Ding, 2022; Wei et al., 2020; Sabater et al., 2022) or secure multiparty computation (SMC) (Fang et al., 2021). Although these privacy computing methods have a provable security level, they have several disadvantages. For example, HE restricts the choice of model structure, DP reduces the performance of the global model, and HE and SMC have high communication or computation costs for participants, which limits their application.

Computational Efficiency dictates that the computation resource in VFL is efficiently used. The computational efficiency of synchronous VFL can be low due to the idle time for participants. In synchronous VFL, the server coordinates with all clients by sending a request to all clients for each batch of training data. The server must wait for all clients' responses to fulfill one global update step before sending the next request to all clients (Liu et al., 2019; Vepakomma et al., 2018; Castiglia et al., 2022; Fang et al., 2021). As a result, all participants must wait for the slowest one, leading to low computational efficiency in synchronous VFL.

Asynchronous VFL (Chen et al., 2020; Hu et al., 2019; Zhang et al., 2021) was proposed to reduce idle time for each participant and improve the computation efficiency. In asynchronous VFL, the client continuously sends its model output to the server without coordination from the server. When the server receives the output from the client, it replies with the necessary information (e.g., partial derivative) to assist the model update of the client. This scheme eliminates most of the idle time for the clients and improves computation efficiency. Our research focuses on asynchronous VFL.

Communication Efficiency is about reducing the communication cost between the parties of VFL. Research has focused on reducing communication rounds (Liu et al., 2019; Wang et al., 2022) or per-round communication overhead (Castiglia et al., 2022). Liu et al. (2019) propose multiple local updates on VFL participants to reduce communication rounds. However, multiple local updates consume more computational resources on clients, which is not favorable in VFL. Wang et al. (2022) apply a better optimization method to speed up convergence and reduce communication rounds. Castiglia et al. (2022) apply compression to the embeddings of client outputs to support efficient communication and multiple local updates, reducing per-round communication overhead and communication rounds.

3 Method

This section introduces the modeling of the VFL problem and proposes our framework that cascades different optimization methods. With a cascaded hybrid optimization method, the advantage of both ZOO and FOO is maximized in one VFL framework.

3.1 Problem definition

We consider a general form of VFL problem (Chen et al., 2020; Hu et al., 2019; Liu et al., 2019; Zhang et al., 2021), which involves a single server and M clients.

Each participant in the VFL possesses n samples within their respective databases. Specifically, each client holds a distinct set of features for each sample, denoted as $x_{i,m}$, while the server holds the corresponding labels for the i-th sample,[1] denoted as y_i.

Clients communicate with the server through the network. To preserve the privacy of the local data. Raw data $x_{i,m}$ and y_i should not be transmitted through the network. The client holds a local model $F_m(w_m;x_{i,m})$ parameterized by $w_m \in \mathbb{R}^{d_m}$ with sample $x_{i,m}$ as input and send the output $c_{i,m}$ of the model to the server through the network. The server holds a model $F_0(w_0;c_{i,1},\dots,c_{i,q})$ which is parameterized by $w_0 \in \mathbb{R}^{d_0}$ and take $c_{i,m}$ from all clients as inputs. The loss function is denoted as $\mathcal{L}(\hat{y}_i, y_i)$.

Ideally, all parties in the VFL framework collaborate to solve a finite-sum problem in the composition form:

$$f(w_0,\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\left[\mathcal{L}(F_0(w_0,c_{i,1},\dots,c_{i,M}),y_i) + \lambda\sum_{m=0}^{M} g(w_m)\right]}_{f_i(w_0,\mathbf{w})} \quad \text{with} \quad c_{i,m} = F_m(w_m;x_{i,m}) \quad \forall m \in [M] \tag{1}$$

where g is the regularization function for the party m, $[M] = \{1, 2, \cdots, M\}$ denote the set of all clients' indices, $\mathbf{w} = \{w_1, w_2, \cdots, w_M\}$ denotes the parameters from all clients, $f_i(w_0, \mathbf{w})$ denotes the loss function for the i-th sample.

3.2 Cascaded hybrid optimization (ZOO & FOO)

To leverage the advantage of ZOO and FOO in one VFL, we apply a cascaded hybrid optimization method, where the upstream (server) and the downstream (client) of the global model apply different optimization methods simultaneously. Specifically, the clients are updated with ZOO and the communication between the server and the client does not contain internal information, which protects privacy. The server is updated with FOO locally, which speeds up the convergence of the VFL without degrading the privacy security.

[1] For brevity, we use a single data sample i for discussion, however, the discussion can be easily generalized to a mini-batch version.

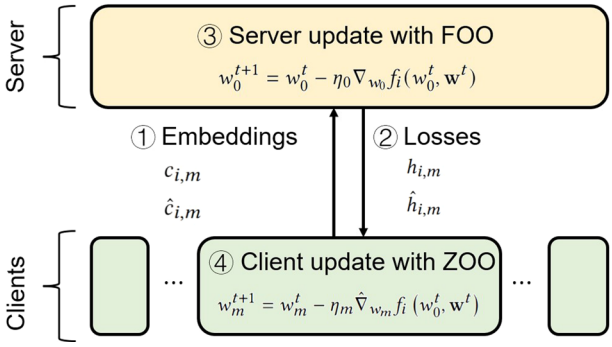


Fig. 2 One round of our VFL framework

3.2.1 Client update with ZOO to ensure privacy security

The models of the clients are trained with the ZOO. The two-point stochastic gradient estimator (Liu et al., 2020; Nesterov & Spokoiny, 2017) w.r.t. the client m's parameter w_m is defined as:

$$\hat{\nabla}_{w_m} f_i(w_0, \mathbf{w}) = \frac{\phi(d_m)}{\mu_m}\left[f_i(w_m + \mu_m u_{i,m}) - f_i(w_m)\right]u_{i,m} \tag{2}$$

where $u_{i,m} \sim p$ is a random direction vector drawn from distribution p. Typically, p is standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, or uniform distribution $\mathcal{U}(\mathcal{S}(\mathbf{0}, 1))$ over a unit sphere at $\mathbf{0}$, with the radius of 1. μ is the smoothing parameter. $f_i(w_m + \mu_m u_{i,m})$ is the simplified form of $f_i(w_0, w_1, w_2, \cdots, w_m + \mu_m u_{i,m}, \cdots, w_q)$, i.e. the loss of the i-th sample with the model parameter of client m changed to $w_m + \mu_m u_{i,m}$. $\phi(d_m)$ is a dimension-dependent factor that relates to the choice of p. To be more specific, if p is $\mathcal{N}(\mathbf{0}, \mathbf{I})$ then $\phi(d_m) = 1$ and if p is $\mathcal{U}(\mathcal{S}(\mathbf{0}, 1))$ then $\phi(d_m) = d_m$.

The clients are unable to compute the gradient of the loss function locally due to the fact that the label of the data is stored on the server. As illustrated in Fig. 2, the clients query the server for the necessary computation material. The active client then computes the model output with or without the perturbation $\mu_m u_{i,m}$ on its parameter and sends them to the server. Specifically, the client's outputs are:

$$c_{i,m} = F_m(w_m; x_{i,m})$$
$$\hat{c}_{i,m} = F_m(w_m + \mu_m u_{i,m}; x_{i,m})$$

Receiving the query from the client, the server replies to the client m with the corresponding loss values $h_{i,m}$ and $\hat{h}_{i,m}$:

$$h_{i,m} = \mathcal{L}(F_0(w_0, c_{i,1}, \ldots, c_{i,m}, \ldots, c_{i,M}), y_i)$$
$$\hat{h}_{i,m} = \mathcal{L}(F_0(w_0, c_{i,1}, \ldots, \hat{c}_{i,m}, \ldots, c_{i,M}), y_i)$$

When the client receives $h_{i,m}$ and $\hat{h}_{i,m}$ from the server, it is able to calculate the two-point gradient estimator via:

$$\hat{\nabla}_{w_m} f_i(w_0, \mathbf{w}) = \frac{\phi(d_m)}{\mu_m}\left[\hat{h}_{i,m} - h_{i,m}\right]u_{i,m} \tag{3}$$

Finally, the client m updates its parameter by gradient descent with the stochastic gradient estimator:

$$w_m^{t+1} = w_m^t - \eta_m \hat{\nabla}_{w_m} f_i\left(w_0^t, \mathbf{w}^t\right)$$

There are two parts of private data in the VFL framework that require protection: the features held by the clients and the labels held by the server. Our framework protects the privacy of the data by concealing the internal information of the participants. A comprehensive analysis of the privacy protection of our framework is presented in Sect. 5.

3.2.2 Server update with FOO to speed up the convergence

The primary issue with ZOO in the context of machine learning is that the variance of the gradient estimation increases as the parameter dimension grows larger, leading to slow convergence of ZOO, particularly for large models. To address this issue, we implemented the FOO on the server to speed up the convergence. It is important to note that the server update is performed locally and does not affect communication with the client or the client's update steps. As a result, the privacy protection of the framework is not compromised while simultaneously accelerating convergence.

The server's model is trained with the first-order gradient. Whenever the server receives a message from the client, it performs one gradient descent step on its local model. Since the server can access the output embeddings $[c_{i,m}]_{m=1}^M$ from all clients and the label y_i, plus that the server naturally has full access to its own model F_0, the server can explicitly calculate the gradient via backpropagation. Specifically, the local gradient of the server is:

$$\nabla_{w_0} f_i(w_0, \mathbf{w}) = \frac{\partial\left[\mathcal{L}(F_0(w_0, c_{i,1}, \ldots, c_{i,M}), y_i) + \lambda g(w_0)\right]}{\partial w_0}$$

And the server's parameter is updated via gradient descent:

$$w_0^{t+1} = w_0^t - \eta_0 \nabla_{w_0} f_i(w_0^t, \mathbf{w}^t) \tag{4}$$

3.3 Asynchronous updates

The global model is trained without coordination among each party. We assume that all messages will be successfully transmitted, and no participants will withdraw during training. A schematic graph is shown in Fig. 2. At each round, only one client is activated and communicates with the server. After the communication, the activated client and the server update their model. The clients' update order can be modeled with a sequence of length T. In the t-th iteration, the client m_t is activated and picks the i-th sample for the update.

To model the delay of the clients, if the client m_t is activated at the t-th iteration, the client updates its parameter w_{m_t} and its delay for the i-th sample on the global model is reset. For all other clients $m \neq m_t$, the delay count is incremented by 1. Formally, the delay for the client m and sample i is updated using the following equation:

$$\tau_{i,m}^{t+1} = \begin{cases} 1, & m = m_t, i = i_t \\ \tau_{i,m}^t + 1, & \text{otherwise} \end{cases}$$

Taking the client delay $\tau_{i,m}^t$ into consideration, we can represent the set of parameters for the delayed clients as:

$$\tilde{\mathbf{w}}^t = \mathbf{w}^{t-\tau_i^t} = [w_1^{t-\tau_{i,1}^t}, \ldots, w_M^{t-\tau_{i,M}^t}]$$

3.4 Algorithm

By combining the ZOO on the client and FOO on the server, we designed an asynchronous VFL framework. The algorithm is presented in Algorithm 1, and the procedure of one update round is summarized in Fig. 2. The procedure of each training round can be summarized as follows: first, the client randomly selects one sample i, computes $c_{i,m}$ and $\hat{c}_{i,m}$, and sends them to the server. Upon receiving the query from client m, the server calculates the corresponding losses $h_{i,m}$ and $\hat{h}_{i,m}$ and sends them back to the client. The server updates its parameter using gradient descent (Eq. 4) immediately after sending the losses to the client. Finally, upon receiving $h_{i,m}$ and $\hat{h}_{i,m}$ from the server, the client updates its parameter using the stochastic gradient estimator given by Eq. 3.

Algorithm 1 Asyn. VFL with Cascaded Hybrid Optimization

Initialize variables for workers $m \in [M]$
while not convergent **do**
  when a **client** m is activated, **do**:
    Randomly select a sample $x_{i,m}$
    Compute $c_{i,m}$, $\hat{c}_{i,m}$ and upload them to the server
    Receive $h_{i,m}$ and $\hat{h}_{i,m}$ from the server (in a listen manner)
    Compute $\hat{\nabla}_{w_m} f_i(w_0, \tilde{\mathbf{w}})$ via Eq. 3 (ZOO)
    Update $w_m \leftarrow w_m - \eta_m \hat{v}_m$
  when server receives $c_{i,m}$ and $\hat{c}_{i,m}$, **do**:
    Compute and send $h_{i,m}$, $\hat{h}_{i,m}$ to client m
    Compute $\nabla_{w_0} f_i(w_0, \tilde{\mathbf{w}})$, (FOO)
    Update $w_0 \leftarrow w_0 - \eta_0 \nabla_{w_0} f_i(w_0, \tilde{\mathbf{w}})$
end while

4 Convergence analysis

4.1 Theoretical challenges and advantages

The theoretical difficulty of our work comes from the cascaded hybrid optimization in the VFL, where different optimization methods are simultaneously applied to the upstream and downstream parts of the VFL. To the best of our knowledge, all related works in VFL only considered a single type of optimization method in the entire VFL during one iteration, whose analytic result can be more easily derived via the same analytic steps on the entire framework. However, our work required different analytic procedures to be applied to different parts of the model to solve the problem, which posed a significant challenge.

Specifically, the analytic procedure for ZOO and FOO is vastly different, making it difficult to analyze these two different optimizations cascaded in a single model.

The theoretical advantage of our framework compared to the ZOO-based VFL (Zhang et al., 2021) is that the convergence rate of our framework is no longer limited by the server's parameter size, as stated in Remark 3. The complete proof of the convergence analysis is provided in "Appendix 1".

4.2 Assumptions

Assumptions 1–4 are the basic assumptions for solving the non-convex optimization problem with stochastic gradient descent (Ghadimi & Lan, 2013; Liu et al., 2019; Zhang et al., 2021). Assumption 1 tells that the global minima f^* is not $-\infty$ (Ghadimi & Lan, 2013; Liu et al., 2018; Zhang et al., 2021). Assumption 2 is used for modeling the smoothness of the loss function $f(\cdot)$, with which we can link the difference of the gradients with the difference of the input in the definition domain. Assumption 3 is a common assumption for stochastic gradient descent telling that the expectation of the estimation of the stochastic gradient of the sample i does not have a systematic error or bias (Ghadimi & Lan, 2013). Assumption 4 tells that the variance of the gradient estimation is bounded (Liu et al., 2018).

Assumption 1 (Feasible optimal solution) Function f is bounded below that is, there exist f^* such that,

$$f^* := \inf_{[w_0,\mathbf{w}]\in\mathbb{R}^d} f(w_0,\mathbf{w}) > -\infty.$$

Assumption 2 (Lipschitz gradient) ∇f_i is L-Lipschitz continuous w.r.t. all the parameter, i.e., there exists a constant L for $\forall\ [w_0,\mathbf{w}], [w_0',\mathbf{w}']$ such that

$$\left\|\nabla_{[w_0,\mathbf{w}]} f_i(w_0,\mathbf{w}) - \nabla_{[w_0,\mathbf{w}]} f_i(w_0',\mathbf{w}')\right\| \le L\left\|[w_0,\mathbf{w}] - [w_0',\mathbf{w}']\right\|$$

specifically there exists an $L_m > 0$ for all parties $m = 0,\cdots,M$ such that $\nabla_{w_m} f_i$ is L_m-Lipschitz continuous:

$$\left\|\nabla_{w_m} f_i(w_0,\mathbf{w}) - \nabla_{w_m} f_i(w_0',\mathbf{w}')\right\| \le L_m\left\|[w_0,\mathbf{w}] - [w_0',\mathbf{w}']\right\|$$

Assumption 3 (Unbiased gradient) For $m \in 0,1,\cdots M$ for every data sample i, the stochastic partial derivatives for all participants are unbiased, i.e.

$$\mathbb{E}_i \nabla_{w_m} f_i(w_0,\mathbf{w}) = \nabla_{w_m} f\big(w_0,\mathbf{w}\big)$$

Assumption 4 (Bounded variance) For $m = 0,1,\cdots,M$, there exist constants $\sigma_m \le \infty$ such that the variance of the stochastic partial derivatives are bounded:

$$\mathbb{E}_i\left\|\nabla_{w_m} f_i(w_0,\mathbf{w}) - \nabla_{w_m} f(w_0,\mathbf{w})\right\|^2 \le \sigma_m^2$$

Assumption 5 is a common assumption for analyzing VFL when bounding some terms for the entire model when the rest parts have been bounded (Castiglia et al., 2022; Gu et al., 2021; Zhang et al., 2021). We only apply this assumption in the parts of convergence analysis that do not affect the analytic result.

Assumption 5 (Bounded block-coordinate gradient) The gradient w.r.t. all the client is bounded, i.e. there exist positive constants $\mathbf{G}_m$ for the client $m = 1, \cdots, M$ the following inequalities hold:

$$\left\|\nabla_{w_m} h_m(w_m; x_{m,i})\right\| \leq \mathbf{G}_m$$

Assumption 6–7 are fundamental assumptions for analyzing the asynchronous VFL (Zhang et al., 2021; Chen et al., 2020; Gu et al., 2021).

Assumption 6 states that the activation of each client in asynchronous VFL is independent, without which the convergence result cannot be further simplified. Assumption 7 states that the delay on the clients is bounded, without which the convergence cannot be achieved.

Assumption 6 (Independent client) The activated client m_t for the global iteration t is independent of $m_0, \cdots, m_{t-1}$ and satisfies $\mathbb{P}(m_t = m) := p_m$

Assumption 7 (Uniformly bounded delay) For each client m, and each sample i, the delay at each global iteration t is bounded by a constant τ. i.e. $\tau_{m,i}^t \leq \tau$

4.3 Theorems

Theorem 1 *Under Assumptions 1–7, to solve the Problem 1 with Algorithm 1 the following inequality holds.*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \leq \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} + \eta\left(4p_*L_*\sigma_*^2 + 8p_*L_*d_*\sigma_*^2 + p_*L_*^3\mu_*^2d_*^2\right)$$
$$+ \eta^2\left(18p_*\tau^2L_*^2d_*\mathbf{G}_*^2 + 5p_*\tau^2L_*^2\mu_*^2L_*^2d_*^2\right) + \mu_*^2\left(p_*L_*^3d_*^2\right)$$

where $L_* = \max_m\left\{L, L_0, L_m\right\}$, $d_* = \max_m\left\{d_m\right\}$, $\eta_0 = \eta_m = \eta \leq \frac{1}{4L_*d_*}$, $\frac{1}{p_*} = \min_m p_m$, $\mu_* = \max_m\left\{\mu_m\right\}$, $\mathbf{G}_* = \max_m\left\{\mathbf{G}_m\right\}$, *and T is the number of iterations.*

Remark 1 Theorem 1 tells that the major factors that affect the convergence are the learning rate η, the smoothing coefficient μ for the ZOO, and the biggest parameter size d_* among the clients.

Corollary 1 *If we choose* $\eta = \frac{1}{\sqrt{T}}$, $\mu = \frac{1}{\sqrt{T}}$, *we can derive*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \leq \frac{1}{\sqrt{T}}\left[4p_*\mathbb{E}\left(f^0 - f^*\right) + 4p_*L_*\sigma_*^2 + 8p_*L_*d_*\sigma_*^2\right]$$
$$+ \frac{1}{T}\left(18p_*\tau^2L_*^2d_*\mathbf{G}_*^2 + 5p_*\tau^2\mu_*^2L_*^4d_*^2 + p_*L_*^3d_*^2\right)$$
$$+ \frac{1}{T^{\frac{3}{2}}}\left(p_*L_*^3d_*^2\right)$$

where the parameters are the same as that in Theorem 1.

Remark 2 Corollary 1 demonstrates the convergence of our cascaded hybrid optimization framework and shows that it converges in $\mathcal{O}\left(\frac{d_*}{\sqrt{T}}\right)$, where $d_* = \max_m \left\{d_m\right\}$ represents the largest model size among the clients, and T denotes the number of iterations.

Remark 3 Comparing our convergence analysis result and ZOO-VFL (Zhang et al., 2021), our result does not include the parameter size of the server (d_0) in the constant terms, which demonstrates that the convergence of the global model is not limited by the size of the server's parameter. Therefore, in our framework, the server can apply a larger model without impacting the convergence of the global model.

5 Security analysis

5.1 Threat model

We discuss the privacy protection of our framework under the "honest-but-curious" and "honest-but-colluded" models.

5.1.1 Honest-but-curious

The "honest-but-curious" threat model refers to a scenario in which a participant is honest and adheres to the protocol, but is curious about the data of other parties. This party may attempt to gain more knowledge about the data of other parties through communication between participants. Specifically, in VFL, clients seek to infer the label from the server, while the server aims to derive the feature from the client.

5.1.2 Honest-but-colluded

The "honest-but-colluded" threat model involves multiple participants colluding to gain more knowledge about the private data from other participants. Specifically, in VFL, clients may work together to infer the label from the server, or the server may collude with some clients to infer the feature from the remaining clients.

5.2 Theorem

Theorem 2 *Our framework can defend against existing privacy inference attacks on VFL under the "honest-but-curious" and "honest-but-colluded" scenarios.*

Proof **Defend Against Label Inference Attack**: Our framework protects the label on the server by concealing its internal information from clients. Specifically, the server responds to the client with the losses of the model, which are limited to a single value for each batch, without revealing the domain of the target task. Moreover, the server keeps the internal details of its model and the domain information associated with the labels confidential from clients. This approach guarantees that the server acts as a black box to clients,

allowing them to collaborate with the server without having access to any task-specific information.

In the context of the "honest-but-curious" model, one client in the VFL system attempts to infer the label from the server.

The "direct label inference" attack from Fu et al. (2022) is based on the gradient information provided by the server and relies on strong assumptions about both the attacker and the victim. Specifically, the attack assumes that the server simply sums the output from all clients and that the attacker has explicit knowledge of this fact. By exploiting this information, the label can be directly inferred from the sign of the element in the gradient provided by the server. However, this attack is not feasible for our framework, as we do not transmit gradients to the client and the server model is agnostic, rather than a simple summation.

The "model completion attack" from Fu et al. (2022) and the "forward embeddings leakage" from Sun et al. (2022) utilize the client's local model and feature to predict the label on the server. For these attacks to be successful, the local model and local feature must be well-represented on the target task. Besides, a certain label for the sample cannot be guaranteed with those attacks. Additionally, these attacks assume that the client has knowledge of the target task, which can be avoided by using our proposed framework.

Deep leakage from gradient and its variant (Zhu et al., 2019; Zhao et al., 2020; Jin et al., 2021) utilize the gradient provided by the server as the optimization objective to reconstruct the true labels of the sample. However, these attacks assume the attacker has access to the server's model, which is not applicable to our current framework.

Under the "honest-but-colluded" model, some clients collude to infer the label from the server, the attacker can access more information in this scenario.

If all clients colluded, the "direct label inference attack", from Fu et al. (2022) still assumes that the client knows that the server uses a simple summation model, which is not applicable to our framework. The "model completion attack" from Fu et al. and the "forward embeddings" attack from Sun et al. (2022) can have better representation on the global task if some client colluded. However, the clients still cannot access the task information from the server, which is not applicable to our model. In the "honest-but-colluded" model, the "deep leakage from the gradient" (Zhu et al., 2019), still requires the gradient information from the server and assumes a simple summation model on the server, which can be avoided with our framework.

Defend Against Feature Inference Attack: Our framework protects the client's features by concealing their internal information from other participants. Clients send the model's output for each batch to the server without revealing the feature's domain. Additionally, the server is unable to access the client's model information. As a result, adversaries view the client as a black box, only able to receive outputs from it. This makes it difficult to infer the feature from the client.

In the "honest-but-curious" model, the server attempts to infer the feature from the clients.

The "deep leakage from gradient" (Zhu et al., 2019) leverages the gradient as the optimization target to infer the feature from the client. However, this method assumes that the server, as the attacker, can access the client's model, which is not possible through the protocol in our framework.

The model inversions attack (Fredrikson et al., 2015) uses the model's output to recover the input of a machine-learning model, which has the potential to be used for feature inference attacks in VFL. However, this attack requires the attacker to have the ability to adaptively query the target model, which the server does not possess this capability in our framework.

The "honest-but-colluded" model allows the server to collude with certain clients to infer features from the remaining clients. Luo et al. (2021) consider a feature inference scenario with two participants, where one participant takes the role of server and client and attempts to infer the feature from the remaining client. They assume that the client uses a logistic regression model, which allows them to reverse the model with the output. However, this method is not applicable to our framework because the client model is agnostic to the attacker. Weng et al. (2020) consider a similar VFL with an extra HE scheme, and they assume that the coordinator with the private key also colludes, enabling the attacker to decrypt the communication. However, this approach is not applicable to our framework as they also assume a specific model on the client. □

6 Experiments

In this section, we did extensive experiments to demonstrate the security of our framework, the convergence of our framework and the feasibility of applying our framework to deep learning tasks.

6.1 Experiment setups

6.1.1 Datasets

We vertically partitioned the dataset among M clients, with each client holding an equal amount of features. The server held the labels. Both clients and the server knew the sample IDs, enabling them to coordinate training on each sample. For the base experiment, we used the MNIST dataset (LeCun et al., 2010), the features of the image were flattened and equally distributed among the clients. For the image classification task, we used the CIFAR-10 dataset (Krizhevsky, 2009), with each client holding half of each image. For the natural language processing (NLP) task, we used the IMDb dataset (McAuley & Leskovec, 2013) where the client held the review text data.

6.1.2 Models

We used a Multi-Layer Perceptron (MLP) for the base experiment to demonstrate the convergence rate of our framework. Although simple, it showed the advantage of our framework.

The base model for clients was a single-layer Fully Connected Layer (FCL) with an input size equal to the feature size of the client's data and an output size of 128 by default. The activation function was ReLU.

The base model for the server was a two-layer FCL whose input was the concatenation of all the clients' outputs $[c_{i,1}, \ldots, c_{i,M}]$. Since the client updated asynchronously, the server held a table of $[c_{i,1}, \ldots, c_{i,M}]$. When the server received an update from client m, it would update the corresponding $c_{i,m}$ in the table and use the table as input of the model. The embedding size of the first layer was 128 by default and the output size of the second layer was the number of classes.

For the image classification task, we applied a split ResNet-18 model (He et al., 2016) on the VFL framework. There were two clients and one server. Each client held half of each image while the server held the labels. The clients preprocessed the images and passed them through the first convolutional layer of ResNet-18. The model on the server comprised the remaining parts of the ResNet-18 model.

For the NLP task, we applied a split distilBERT (Devlin et al., 2018) model on the VFL framework. The network consisted of one client and one server, the client holding the embedding layer of the transformer and the server holding the remaining parts of the model.

6.1.3 The frameworks for comparison

We conducted a comparative analysis of our asynchronous VFL framework with four baseline methods: VAFL (Chen et al., 2020), ZOO-VFL (Zhang et al., 2021), Split-Learning (Vepakomma et al., 2018), and Syn-ZOO-VFL.[2] All baselines employ a single optimization method across the entire VFL, and we applied the same base models to all frameworks. While ZOO-VFL and Syn-ZOO-VFL share the same message transmission content as our framework, VAFL and Split-Learning transmit partial derivatives through the network, which poses a privacy risk. It is worth noting that our framework offers the same level of privacy security as ZOO-VFL and Syn-ZOO-VFL, whereas VAFL is privacy risky. Therefore, we consider the experiment on VAFL and Split-Learning as an upper bound for convergence rate comparison among these frameworks, but it is not practical due to the privacy risk.

6.1.4 Training procedures

We employed different learning rates for the server and clients in our experiments, as their update times differ. The optimal learning rate η was selected from the range [0.020, 0.015, 0.010, 0.005, 0.001] for all frameworks. We chose this range because $\eta = 0.001$ was too small, resulting in slow convergence, while $\eta = 0.020$ was too large for ZOO to achieve satisfactory test accuracy. We set μ to 0.001 for all experiments, which was the optimal parameter selected from the range [0.1, 0.01, 0.001, 0.0001, 0.00001] through preliminary experiments. To make a fair comparison, we applied the vanilla SGD strategy to all VFL frameworks. The number of training epochs was 100 by default to ensure model convergence.

For training the split ResNet-18 on distributed CIFAR-10, we trained the model for 40 epochs. To determine the optimal learning rate η for the framework, we searched η within the range [0.03, 0.01 0.003, 0.001] for the framework. We selected the one with the highest test accuracy. For the ZOO-VFL and Syn-ZOO-VFL, we searched for the optimal learning rate in an exponential manner, i.e., $[\cdots, 3 \times 10, 10, 3, 1, 0.3, 0.1, \cdots]$. The upper limit for the search was where the loss kept increasing, and the lower limit was where the model training accuracy did not increase for every epoch. We selected the learning rate that allowed the model to train the fastest.

For the NLP task, we finetuned the pre-trained distil-BERT model. Since the model is pre-trained, we set the number of training epochs to 10. The hyperparameter tuning scheme

[2] This is the synchronous version of ZOO-VFL and the algorithm is in the "Appendix 2".

Table 1 Demonstration with Direct Label Inference Attack

	FOO frameworks	ZOO frameworks
Curious Client	$100_{\pm 0.0}$	$11.7_{\pm 0.07}$
Eavesdropper	$100_{\pm 0.0}$	$10.0_{\pm 0.1}$

was the same as that used for the CIFAR-10 task. All of the test accuracy presented in this paper (including the Appendix) is derived from five independent runs.

6.2 A demonstration on defending against label inference attack

In this experiment, we aimed to demonstrate the security levels of ZOO-based VFL (ZOO-VFL, Syn-ZOO-VFL, and ours) and FOO-based VFL (Split-Learning and VAFL) against a direct label inference attack from Fu et al. (2022). The attack is only effective for the "model without split" VFLs where the server simply sums up the output from all clients. The threat model involves a curious client aiming to infer labels from the victim server. The client can design the query for the server to acquire partial derivative w.r.t. the global model's output layer, i.e., $\frac{\partial \mathcal{L}(y;y_i)}{\partial y^c}$, where y represents the probability output for all classes, y^c is the probability for the c-th class predicted by the model, and there are C classes in total. The label can be directly inferred with the sign of $\frac{\partial \mathcal{L}(y;y_i)}{\partial y^c}$, i.e., if the sign of it is negative, then the label for sample i is c; otherwise, the sign is positive. Note that this attack scenario where the server model simply sums the output of the clients is very strong (the server is too vulnerable). However, it has effectively demonstrated the vulnerability of transmitting gradients in VFL.

To simulate a curious client who wanted to infer the label from the server, we designed a dummy client that directly generated a random vector $c_{i,m} \in \mathcal{R}^C$, with elements sampled from $\mathcal{N}(0,1)$. The client then randomly selected a $u \in \mathcal{R}^C$ to compute $\hat{c}_{i,m} = c_{i,m} + u$. The server then responded with the corresponding losses $\hat{h}_{i,m}$ and $h_{i,m}$, and the curious client estimated $\frac{\partial \mathcal{L}(y;y_i)}{\partial y^c}$ using gradient estimation, i.e. $\hat{\nabla}_y \mathcal{L}(y;y_i) = \frac{\phi(d)}{\mu}(\hat{h}_{i,m} - h_{i,m})u$. In addition to the curious client, eavesdroppers also sought to infer labels from the server. However, when clients are benign, eavesdroppers cannot obtain the client's u value. Therefore, in the experiment, they randomly generated a u to estimate the gradient.

We conducted the label inference attack using the MNIST dataset, using a batch size of 64. The attack success rate was calculated by dividing the number of correctly predicted samples by the total number of samples. The VFL framework was run for a single epoch, during which the attacker predicted the label of all samples based on the information they obtained. The VFL framework consisted of two clients and one server, where the server model summed up the output from the clients and replied with the losses value w.r.t. the client's output. In the trial involving the curious client, there was one curious client and one benign client. In the trial involving the eavesdropper, both clients were benign.

The results are present in Table 1, where each experiment consists of 5 independent trials. The table indicates that the use of FOO in VFL poses a serious privacy vulnerability, as both curious clients and eavesdroppers can infer certain labels. On the other hand, when ZOO is applied to VFL, the malicious client who dedicated designed the query only gains

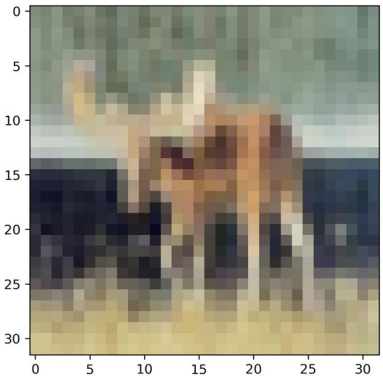


Fig. 3 The target data, with the victim client holding the left half

a slight advantage with one query. Additionally, eavesdroppers were unable to infer the label from the messages due to the lack of gradient information on the server.

6.3 A demonstration on defending against feature inference attack

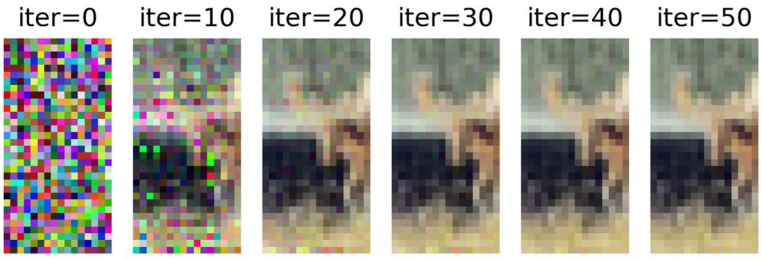
In this experiment, we demonstrate the capability of our framework in defending against the feature inference attacks based on "deep leakage from gradient" (DLG) (Zhu et al., 2019). Besides, we highlight the vulnerability of gradient-based VFL in the context of such attacks.

We designed an experiment where the VFL involved two clients, each equipped with a Convolutional Neural Network (CNN). In the CNN architecture, the first two layers are convolutional layers, employing the Sigmoid activation function. The final layer is a fully connected layer. The server aggregates the logits generated by each client through a summation process. Each client possessed half of each image from CIFAR-10 as their private dataset.

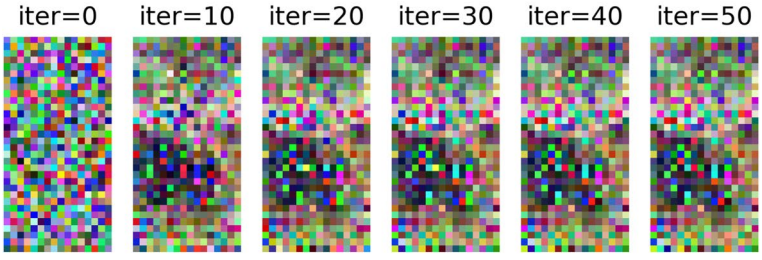
Without loss of generality, we assume that client 1 is the victim, and the server is the curious party. We assume that at some stage of the training, the attacker obtained a snapshot of the model parameters from Client 1 and the corresponding gradient w.r.t. the sample i. The gradient information obtained by the attack is $\nabla_{w_1} f_i(w_0, \mathbf{w}) = \frac{\partial f_i(w_0, \mathbf{w})}{\partial w_1}$ under the FOO case (VAFL and Split-learning), or $\hat{\nabla}_{w_1} f_i(w_0, \mathbf{w}) = \frac{\phi(d_1)}{\mu_1}\left(\hat{h}_{i,1} - h_{i,1}\right) u_{i,1}$ under the ZOO case (ZOO-VFL and ours). Having obtained the model parameter and gradient information, the attacker aims to reconstruct the private data $x_{i,1}$ maintained by Client 1.

We randomly selected an image from the CIFAR-10 dataset, specifically choosing the image at index 28, which belongs to the class "deer". Figure 3 shows the original private data from the two clients, where client 1 (victim) held the left half of the picture. Figure 4a depicts the DLG attack on the First Order Optimization (FOO)-based model, while Fig. 4b showcases the DLG attack on the FOO-based model with Gaussian Noise $\mathcal{N}(0, 0.03)$ added to each dimension of the gradient. Lastly, Fig. 4c illustrates the DLG attack on the ZOO-based model.

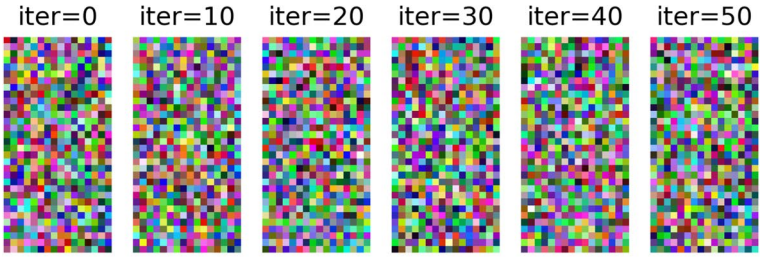
Our observations indicate that DLG successfully infiltrated the VFL model when an accurate gradient and the model snapshot were acquired. However, the DLG attack proved ineffective against our framework trained with ZOO. This outcome is likely attributed to the randomness introduced by the ZOO, which hinders the attacker from obtaining accurate gradient information for the attack.



(a) FOO-based VFL



(b) FOO-based VFL with Gaussian Noise in Gradient



(c) ZOO-based VFL

Fig. 4 DLG attack on the VFL framework

6.4 The convergence for different numbers of clients

In this experiment, we compared the convergence curve between our framework and others, with varying numbers of clients. With the base model, we set the number of clients to {4, 6, 8} and plotted the epoch-training accuracy curve in Fig. 5. As illustrated in the figure, our framework exhibited a more stable convergence rate than ZOO-VFL. The curve for ZOO-VFL displayed significant vibration between the fifth and tenth epoch, primarily due to client delay. This phenomenon was less obvious in our framework. Table 2 shows the test accuracy achieved after the training procedure. Our framework demonstrated a slight test accuracy loss compared to VAFL, which was a trade-off for improving the privacy and security of the framework. In contrast, our framework

Table 2 Test accuracy (%) for the convergence of different number of clients experiments

	Number of clients		
	4	6	8
Split-Learning	$97.7_{\pm 0.1}$	$97.7_{\pm 0.1}$	$97.5_{\pm 0.2}$
VAFL	$97.7_{\pm 0.2}$	$97.8_{\pm 0.1}$	$97.7_{\pm 0.2}$
Syn-ZOO-VFL	$87.4_{\pm 0.3}$	$87.4_{\pm 0.2}$	$87.7_{\pm 0.3}$
ZOO-VFL	$89.0_{\pm 0.3}$	$89.4_{\pm 0.4}$	$89.2_{\pm 0.4}$
VFL-Cascaded (Ours)	$\mathbf{96.4}_{\pm 0.3}$	$\mathbf{96.5}_{\pm 0.4}$	$\mathbf{96.4}_{\pm 0.3}$

The best test accuracyamong the three privacy-protected baselines are given in bold

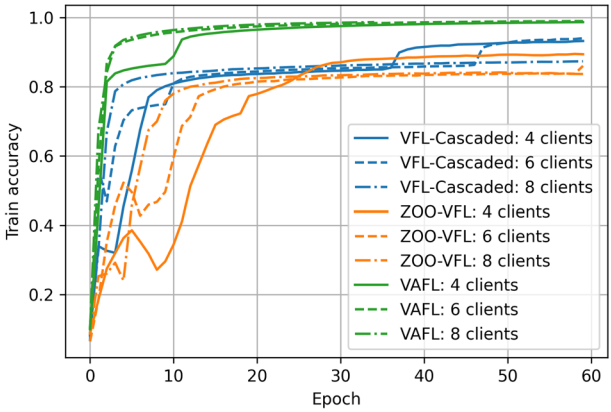


Fig. 5 Learning curve for different numbers of clients

achieved a much higher test accuracy than ZOO-VFL, indicating that ZOO-VFL does not possess good convergence characteristics.

6.4.1 More robust hyperparameter tuning

When searching for the optimal learning rate, we observed that the selection of the learning rate for ZOO-VFL was more sensitive compared to VFL-Cascaded. This sensitivity is an undesirable characteristic for hyperparameter tuning, especially in federated learning, which introduces more hyperparameters than centralized training (Kairouz et al., 2019).

Assuming that we have obtained the optimal learning rate for ZOO-VFL, it is worth noting that even a slight increase in the learning rate can lead to a significant reduction in test accuracy. Conversely, a minor decrease in the learning rate can also slow the convergence and decrease test accuracy. In contrast, our framework demonstrates greater resilience in learning rate selection, resulting in a more stable performance with less deviation in hyperparameters.

To demonstrate the resilience of our framework, we reported the test accuracy at a different learning rate for comparing the ZOO-VFL and VFL-Cascaded. We selected the server learning rate from [0.020, 0.015, 0.010, 0.005, 0.001], and trained the model for 200 epochs to make sure the model converges. The test accuracy is presented in Fig. 6. Our findings indicate that the deviation from the optimal learning rate had a more significant impact on ZOO-VFL than VFL-Cascaded.

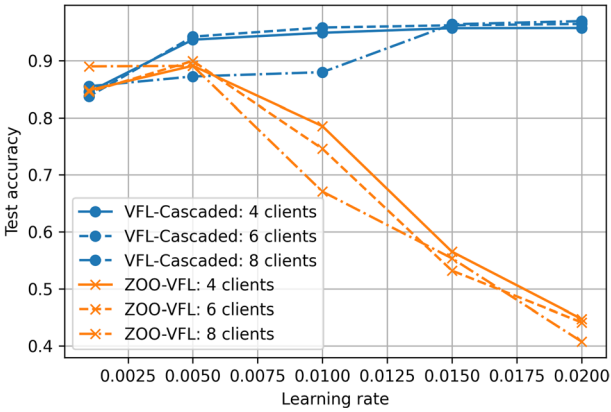


Fig. 6 Robustness of the hyperparameter

In VFL, a more robust hyperparameter is favorable as it requires less tuning and computational resources. This is particularly important as communication between the server and clients in VFL is costly.

6.5 The convergence for different server model sizes

6.5.1 Base model

In this experiment, we conducted a comparison of the convergence rates between our framework and other frameworks, using a variety of server model sizes. The frameworks were applied to four clients and one server, and we tested it on different widths of the server model, specifically the embedding size of the first layer. We varied the embedding size of the first layer of the server from the default value of 128 to 256 and 512, resulting in server model parameter counts of 66954, 133898, and 267786, respectively.

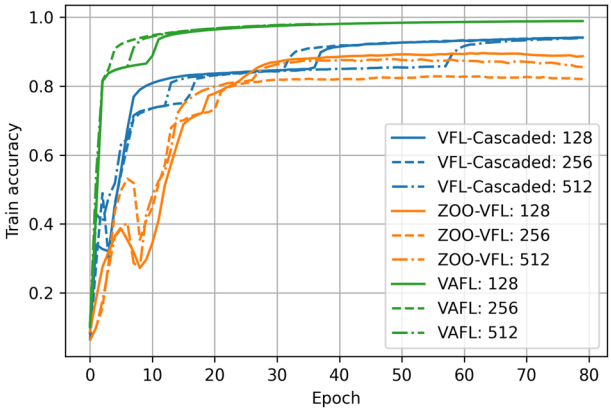
The training curve is presented in Fig. 7a. As shown in the figure, for all different sizes of models, our framework has a more stable convergence than ZOO-VFL, where the vibration between the fifth and tenth epoch is less obvious. Table 3 presents the test accuracy achieved after the training procedure. For all model sizes, our model has a significantly higher test accuracy than ZOO-VFL. However, when compared to VAFL, our framework incurs a trade-off of approximately 1% in test accuracy for privacy security.

To demonstrate the superiority of our framework in training larger models, we conducted tests on deep learning tasks, including image classification and text classification (NLP).

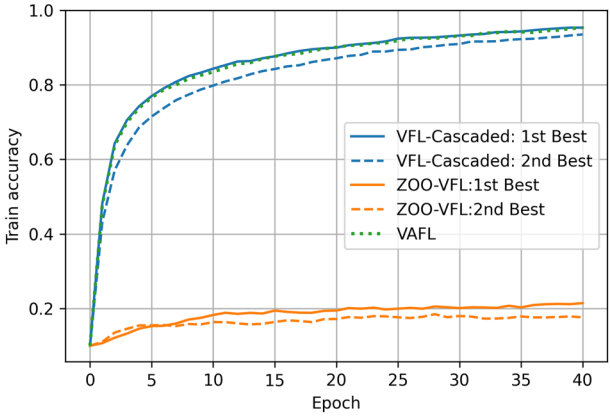
6.5.2 Image classification

The training curve for the image classification task on CIFAR-10 using the split ResNet-18 model is presented in Fig. 7b. As depicted in the figure, our framework maintains a reasonable convergence rate and is robust for the best two learning rates, where the best curve almost overlaps the training curve for VAFL. The training accuracy for ZOO-VFL gradually increases from 0.10 to 0.22 during the training process, indicating the slow convergence problem of ZOO-VFL with the large model. Table 3 shows the test accuracy. By

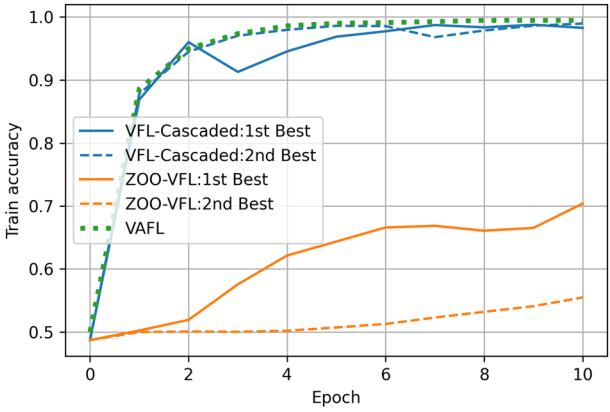
Fig. 7 Learning curve for different server model size



(a) MNIST - Base Model (Small)



(b) CIFAR10 - ResNet18 (Medium)



(c) IMDb - DistillBERT (Larger)

Table 3 The test accuracy (%) for the different model size experiments

	MNIST			CIFAR-10	IMDb
	MLP—server embedding size			ResNet-18	Distil-BERT
	128	256	512		
Split-Learning	$97.7_{\pm 0.1}$	$98.1_{\pm 0.2}$	$98.1_{\pm 0.1}$	$84.7_{\pm 0.2}$	$90.5_{\pm 0.1}$
VAFL	$97.7_{\pm 0.2}$	$97.8_{\pm 0.2}$	$97.8_{\pm 0.1}$	$88.1_{\pm 0.1}$	$90.5_{\pm 0.1}$
Syn-ZOO-VFL	$87.5_{\pm 0.4}$	$88.7_{\pm 0.2}$	$88.2_{\pm 0.3}$	–	
ZOO-VFL	$89.0_{\pm 0.3}$	$85.3_{\pm 0.8}$	$86.0_{\pm 0.7}$	–	–
VFL-Cascaded (Ours)	$\mathbf{96.4}_{\pm 0.3}$	$\mathbf{96.5}_{\pm 0.4}$	$\mathbf{96.2}_{\pm 0.3}$	$\mathbf{87.2}_{\pm 0.6}$	$\mathbf{89.6}_{\pm 0.2}$

The best test accuracyamong the three privacy-protected baselines are given in bold

applying our framework, we can achieve a reasonable test accuracy in 40 training epochs using a modified split ResNet18 model.

6.5.3 Natural language processing

We also demonstrated that a more complex transformer-based model for NLP can be trained with our VFL framework. The training curve is depicted in Fig. 7c. The dataset comprises of two classes, therefore, the training accuracy commences at around 50%.

The difference in convergence speed becomes more noticeable when using a large model. In our framework, the training accuracy reached 94% in the second epoch, which took approximately 45 min. In contrast, ZOO-VFL's training accuracy only rose from 50% to 70% in 10 epochs, requiring around 6 h of training time, and the model's performance remained close to random guessing. Besides, the learning rate was more robust for VFL-Cascaded, with most of the parameters we tuned proving to be effective. In contrast, ZOO-VFL's second-best learning rate exhibited much slower convergence, and the third-best learning rate failed to converge altogether. The test accuracy of our model is presented in Table 3. Since training for around 6 h is contrary to the basic idea of fine-tuning, we test the model after 2 epochs of training. The results demonstrate that our framework is capable of training an extremely large deep-learning model.

7 Limitations and discussions

In our framework, we utilized ZOO and FOO strategically to address the demanding aspects of the VFL framework. Specifically, we employed ZOO on the client to maximize model applicability and privacy protection, and FOO on the server to accelerate convergence. We carefully balanced the advantages and disadvantages of ZOO and FOO in different parts of the VFL model to ensure that our framework meets all requirements for practical VFL. A detailed comparison of the frameworks is presented in Table 4 ("S" for the server and "C" for the client, "**F**" for the entire framework). It is important to note that the inherent limitations of ZOO and FOO were not eliminated. That is, ZOO's slow convergence makes it unsuitable for dealing with large models on the client side, while the server can only handle differentiable models.

However, our framework is more suitable for real-world application scenarios for several reasons. Firstly, in VFL, the server is the initiator and sole beneficiary of the

Table 4 Comparison with typical VFL frameworks

	VAFL			ZOO-VFL			Ours		
	S	C	**F**	S	C	**F**	S	C	**F**
Model Applicability	✗	✗	✗	✓	✓	✓	✗	✓	✓
Fast Convergence	✓	✓	✓	✗	✗	✗	✓	✗	✓
Privacy Security	✗	✗	✗	✓	✓	✓	✓	✓	✓
Comp. Efficiency	✓	✓	✓	✓	✓	✓	✓	✓	✓

framework, with all clients acting as collaborators. As such, it is more cost-effective for the server to train a larger model to achieve better prediction results, as only the server obtains the prediction. Secondly, the server typically has more computational resources than the clients, making it computationally efficient for the server to train a larger model. Thirdly, as the server is the initiator and has the ability to select its model, the model applicability of the server is not as critical in VFL. Conversely, for clients, their models are unknown to the initiator of the VFL, making the model-agnostic characteristic important. Therefore, our framework is more suitable for real-world applications than other frameworks that use a unified optimization method.

8 Conclusions

We proposed a novel VFL framework where different optimization methods were applied to the upstream (server) and the downstream (client) of the VFL cascaded. This approach maximized the benefits of both optimization methods. The clients are optimized with ZOO to protect privacy, while the server is optimized with FOO to accelerate convergence without compromising the framework's privacy. Theoretical results demonstrated that our framework with cascaded hybrid optimization converges faster than the ZOO-based VFL, and that applying a large model on the server does not hinder convergence. Extensive experiments demonstrated that our framework achieves better convergence characteristics compared with the ZOO-based VFL while maintaining the same level of privacy security.

Appendix 1: Convergence analysis

Notation

Table 5 summarizes the parameters used in the convergence analysis.

Lemmas

Lemma 1 (Zeroth-order optimization) *For arbitrary* $f \in C_L^1(\mathcal{R}^d)$, *we have*:

(1) $f_\mu(x)$ *is continuously differentiable*, *its gradient is Lipschitz continuous with* $L_\mu \leq L$:

$$\nabla f_\mu(x) = \mathbb{E}_{\mathbf{u}}\left[\hat{\nabla} f(x)\right] \tag{5}$$

Table 5 Notation table

Basic	
w_0	The parameter for the server
w_m	The parameter for the client m
$\mathbf{w} = [w_1, w_2, \cdots, w_M]$	The grouped parameters for all the clients
$f(w_0, \mathbf{w}) = f(w_0, \mathbf{w}, X, y)$	The global loss function
$f_i(w_0, \mathbf{w}) = f_i(w_0, w_1, \ldots, w_M)$	The loss function for the sample i
Notation with timestep (t), *clients' delay* $(\tilde{\mathbf{w}})$, *ZOO gradient estimator* $(\hat{\nabla})$	
w_m^t	The client m's parameter, at global timestep t
$\mathbf{w}^t = [w_1^t, \ldots, w_M^t]$	The clients' parameter at global timestep t
$\tilde{\mathbf{w}}^t = \mathbf{w}^{t-\tau_i^t} = [w_1^{t-\tau_{1,i}^t}, \ldots, w_M^{t-\tau_{M,i}^t}]$	The delayed parameter for all the clients at global time step t (and the local timestep is 0 for all w)
$\hat{\nabla}_{w_m} f_i(w_0, \mathbf{w}) = \frac{\phi(d_{h_m})}{\mu_m}\left[f_i(w_m + \mu_m u_{m,i}) - f_i(w_m)\right]u_{m,i}$	The ZO gradient estimator w.r.t. the client m's parameter w_m

where $\mathbf{u}$ *is drawn from the uniform distribution over the unit Euclidean sphere, and* $\hat{\nabla} f(x) = \frac{d}{\mu}\left[f(x+\mu\mathbf{u}) - f(x)\right]u$ *is the gradient estimator,* $f_\mu(x) = \mathbb{E}_{\mathbf{u}}\left[f(x+\mu\mathbf{u})\right]$ *is the smooth approximation of* f.

2) *For any* $x \in \mathbb{R}^d$,

$$|f_\mu(x) - f(x)| \leq \frac{L\mu^2}{2} \tag{6}$$

$$\left\|\nabla f_\mu(x) - \nabla f(x)\right\|^2 \leq \frac{\mu^2 L^2 d^2}{4} \tag{7}$$

$$\frac{1}{2}\left\|\nabla f(x)\right\|^2 - \frac{\mu^2 L^2 d^2}{4} \leq \left\|\nabla f_\mu(x)\right\|^2 \leq 2\left\|\nabla f(x)\right\|^2 + \frac{\mu^2 L^2 d^2}{2} \tag{8}$$

3) *For any* $x \in \mathbb{R}^d$,

$$\mathbb{E}_{\mathbf{u}}\left[\left\|\hat{\nabla} f(x)\right\|^2\right] \leq 2d\left\|\nabla f(x)\right\|^2 + \frac{\mu^2 L^2 d^2}{2} \tag{9}$$

Lemma 1 helps build a connection between $f(\cdot)$ and its smooth approximation $f_{\mu_m}(\cdot)$ of the convergence analysis. Proof of this lemma is provided in Liu et al. (2018); Gao et al. (2018).

Bound the global update round

In one global round during training, the client m_t is activated, and the server and the client m_t update one step.

Taking expectations w.r.t. the sample i and the random direction u for the zeroth-order optimization in one global update round.

$$
\begin{aligned}
&\mathbb{E}_{i,u}\left[f\left(w_0^{t+1}, w_1^t, \cdots, w_{m_i}^{t+1}, \cdots, w_M^t\right) - f\left(w_0^t, w_1^t, \cdots, w_{m_i}^t, \cdots, w_M^t\right)\right] \\
&\overset{1)}{\leq} \underbrace{-\eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right)\right\rangle}_{a)} \\
&\underbrace{+\frac{1}{2} L \eta_0^2 \mathbb{E}_i \left\| \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right)\right\|^2}_{b)} \\
&\underbrace{-\eta_{m_i} \mathbb{E}_{i,u} \left\langle \nabla_{w_{m_i}} f\left(w_0^t, \mathbf{w}^t\right), \hat{\nabla}_{m_i} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right)\right\rangle}_{c)} \\
&\underbrace{+\frac{1}{2} L \eta_{m_i}^2 \mathbb{E}_{i,u} \left\| \hat{\nabla}_{m_i} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right)\right\|^2}_{d)} \\
&\overset{2)}{\leq} -\frac{1}{2}\eta_0 \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + \frac{1}{2}\eta_0 L_0^2 \mathbb{E}_i \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t\right\|^2 \\
&+ L\eta_0^2 L_0^2 \mathbb{E}_i \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 + L\eta_0^2 \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + L\eta_0^2 \sigma_0^2 \\
&- \frac{1}{2}\eta_{m_i} \mathbb{E}_{i,u} \left\| \nabla_{w_{m_i}} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + \frac{1}{4}\eta_{m_i} \mu_{m_i}^2 L_{m_i}^2 d_{m_i}^2 + \eta_{m_i} L_{m_i}^2 \mathbb{E}_{i,u} \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t\right\|^2 \\
&+ 2L\eta_{m_i}^2 d_{m_i} \mathbb{E}_{i,u} \left\| \nabla_{w_{m_i}} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + 2L\eta_{m_i}^2 d_{m_i} L_{m_i}^2 \mathbb{E}_{i,u} \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t\right\|^2 + 2L\eta_{m_i}^2 d_{m_i} \sigma_{m_i}^2 \\
&+ \frac{1}{4} L \eta_{m_i}^2 \mu_{m_i}^2 L_{m_i}^2 d_{m_i}^2 \\
&\leq -\left(\frac{1}{2}\eta_0 - L\eta_0^2\right) \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \\
&+ \left(\frac{1}{2}\eta_0 + L\eta_0^2\right) L_0^2 \mathbb{E}_i \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 + L\eta_0^2 \sigma_0^2 \\
&- \left(\frac{1}{2}\eta_{m_i} - 2L\eta_{m_i}^2 d_{m_i}\right) \mathbb{E}_{i,u} \left\| \nabla_{w_{m_i}} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + \left(\eta_{m_i} + 2L\eta_{m_i}^2 d_{m_i}\right) L_{m_i}^2 \mathbb{E}_{i,u} \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t\right\|^2 \\
&+ 2L\eta_{m_i}^2 d_{m_i} \sigma_{m_i}^2 + \frac{1}{4}\left(L\eta_{m_i}^2 + \eta_{m_i}\right) \mu_{m_i}^2 L_{m_i}^2 d_{m_i}^2 \\
&\overset{3)}{\leq} -\left(\frac{1}{2}\eta_0 - L\eta_0^2\right) \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 - \left(\frac{1}{2}\eta_{m_i} - 2L\eta_{m_i}^2 d_{m_i}\right) \mathbb{E}_{i,u} \left\| \nabla_{w_{m_i}} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \\
&+ \left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right) L_0^2 + \left(\eta_{m_i} + 2L\eta_{m_i}^2 d_{m_i}\right) L_{m_i}^2\right] \mathbb{E}_i \|\mathbf{w} - \tilde{\mathbf{w}}\|^2 \\
&+ L\eta_0^2 \sigma_0^2 + 2L\eta_{m_i}^2 d_{m_i} \sigma_{m_i}^2 + \frac{1}{4}\left(L\eta_{m_i}^2 + \eta_{m_i}\right) \mu_{m_i}^2 L_{m_i}^2 d_{m_i}^2
\end{aligned}
\tag{10}
$$

where 1) applies Assumption 2 (smoothness), 2) plugging in a, b, c& d, 3) collect the equation.

For a)

$$
\begin{aligned}
&-\eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right)\right\rangle \\
&= -\eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right) - \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right) + \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right)\right\rangle \\
&= -\eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right) - \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right)\right\rangle \\
&- \eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right)\right\rangle \\
&\overset{1)}{=} -\eta_0 \mathbb{E}_i \left\langle \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right), \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right) - \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right)\right\rangle - \eta_0 \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \\
&\overset{2)}{=} -\frac{1}{2}\eta_0 \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + \frac{1}{2}\eta_0 \mathbb{E}_i \left\| \nabla_{w_0} f_i\left(w_0^t, \tilde{\mathbf{w}}^t\right) - \nabla_{w_0} f_i\left(w_0^t, \mathbf{w}^t\right)\right\|^2 \\
&\overset{3)}{=} -\frac{1}{2}\eta_0 \mathbb{E}_i \left\| \nabla_{w_0} f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + \frac{1}{2}\eta_0 L_0^2 \mathbb{E}_i \left\| \mathbf{w}^t - \tilde{\mathbf{w}}^t\right\|^2
\end{aligned}
\tag{11}
$$

where 1) applies Assumption 3 (unbiased gradient), 2) applies $\langle a, b\rangle \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$, 3) applies Assumption 2 (smoothness).

For b):

$$
\begin{aligned}
&\frac{1}{2}L\eta_0^2\mathbb{E}_i\left\|\nabla_{w_0}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2 \\
&= \frac{1}{2}L\eta_0^2\mathbb{E}_i\left\|\nabla_{w_0}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_0}f_i\left(w_0^t,\mathbf{w}^t\right) + \nabla_{w_0}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{1)}{\leq} L\eta_0^2\mathbb{E}_i\left\|\nabla_{w_0}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_0}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + L\eta_0^2\mathbb{E}_i\left\|\nabla_{w_0}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{2)}{\leq} L\eta_0^2L_0^2\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2 + L\eta_0^2\mathbb{E}_i\left\|\nabla_{w_0}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{3)}{\leq} L\eta_0^2L_0^2\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2 + L\eta_0^2\left(\left\|\nabla_{w_0}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + \sigma_0^2\right) \\
&\leq L\eta_0^2L_0^2\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2 + L\eta_0^2\left\|\nabla_{w_0}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + L\eta_0^2\sigma_0^2
\end{aligned}
\tag{12}
$$

where 1): $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, 2) applies Assumption 2 (smoothness), 3) applies $\mathbb{E}(X^2) = \mathbb{E}(X)^2 + \mathrm{Var}(X)$ and Assumption 4 (bounded variance).

For c):

$$
\begin{aligned}
&-\eta_{m_t}\mathbb{E}_{i,u}\left\langle\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right),\hat{\nabla}_{m_t}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\rangle \\
&\overset{1)}{=} -\eta_{m_t}\mathbb{E}_{i,u}\left\langle\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right),\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\rangle \\
&= -\eta_{m_t}\mathbb{E}_{i,u}\left\langle\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right),\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right) + \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\rangle \\
&\overset{2)}{=} -\eta_{m_t}\mathbb{E}_{i,u}\left\langle\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right),\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\rangle \\
&\quad -\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{3)}{=} -\frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + \frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&= -\frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\quad + \frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) + \nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{4)}{=} -\frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + \eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_{\mu_{m_t},i}\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2 \\
&\quad + \eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{5)}{=} -\frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + \frac{1}{4}\eta_{m_t}\mu_{m_t}^2L_{m_t}^2d_{m_t}^2 \\
&\quad + \eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right) - \nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{6)}{=} -\frac{1}{2}\eta_{m_t}\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 + \frac{1}{4}\eta_{m_t}\mu_{m_t}^2L_{m_t}^2d_{m_t}^2 + \eta_{m_t}L_{m_t}^2\mathbb{E}_{i,u}\left\|\mathbf{w}^t-\tilde{\mathbf{w}}^t\right\|^2
\end{aligned}
\tag{13}
$$

where 1) applies Eq. 5 in Lemma 1, 2) applies Assumption 3 (unbiased gradient), 3) applies $\langle a, b\rangle \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$, 4) applies $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, 5) applies Eq. 7 in Lemma 1, 6) applies Assumption 2 (smoothness).

For d):

$$
\begin{aligned}
&\frac{1}{2}L\eta_{m_t}^2\,\mathbb{E}_{i,u}\left\|\hat{\nabla}_{m_t}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2\\
&\overset{1)}{\leq}\ \frac{1}{2}L\eta_{m_t}^2\,\mathbb{E}_{i,u}\left(2d_{m_t}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2+\frac{1}{2}\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\right)\\
&=\ L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&=\ L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)-\nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)+\nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&\overset{2)}{\leq}\ 2L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)-\nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2\\
&+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&\overset{3)}{\leq}\ 2L\eta_{m_t}^2\,d_{m_t}L_{m_t}^2\,\mathbb{E}_{i,u}\left\|\mathbf{w}^t-\tilde{\mathbf{w}}^t\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\mathbf{w}^t\right)\right\|^2+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&\overset{4)}{\leq}\ 2L\eta_{m_t}^2\,d_{m_t}L_{m_t}^2\,\mathbb{E}_{i,u}\left\|\mathbf{w}^t-\tilde{\mathbf{w}}^t\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\left(\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2+\sigma_{m_t}^2\right)\\
&+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&=\ 2L\eta_{m_t}^2\,d_{m_t}L_{m_t}^2\,\mathbb{E}_{i,u}\left\|\mathbf{w}^t-\tilde{\mathbf{w}}^t\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\sigma_{m_t}^2\\
&+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\\
&=\ 2L\eta_{m_t}^2\,d_{m_t}\,\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2+2L\eta_{m_t}^2\,d_{m_t}L_{m_t}^2\,\mathbb{E}_{i,u}\left\|\mathbf{w}^t-\tilde{\mathbf{w}}^t\right\|^2+2L\eta_{m_t}^2\,d_{m_t}\sigma_{m_t}^2\\
&+\frac{1}{4}L\eta_{m_t}^2\mu_{m_t}^2L_{m_t}^2d_{m_t}^2
\end{aligned}
\tag{14}
$$

where 1) applies Eq. 9 in Lemma 1, 2) applies $\|a+b\|^2\leq 2\|a\|^2+2\|b\|^2$, 3) applies Assumption 2 (smoothness), 4) applies $\mathbb{E}(X^2)=\mathbb{E}(X)^2+\mathrm{Var}(X)$ and Assumption 4 (bounded variance).

Combine the gradient

Start with the Eq. 10, additionally taking expectation w.r.t. activated client m_t, and applying the Assumption 6 (independent client).

$$
\begin{aligned}
&\mathbb{E}_{m_t,i,u}\left[f\left(w_0^{t+1},w_1^t,\cdots,w_{m_t}^{t+1},\cdots,w_M^t\right)-f\left(w_0^t,w_1^t,\cdots,w_{m_t}^t,\cdots,w_M^t\right)\right]\\
&\quad\le -\left(\frac{1}{2}\eta_0-L\eta_0^2\right)\mathbb{E}_i\left\|\nabla_{w_0}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2-\sum_{m=1}^{M}p_m\left(\frac{1}{2}\eta_m-2L\eta_m^2d_m\right)\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2\\
&\quad+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2\\
&\quad+L\eta_0^2\sigma_0^2+\sum_{m=1}^{M}p_m2L\eta_m^2d_m\sigma_m^2+\sum_{m=1}^{M}p_m\frac{1}{4}\left(L\eta_m^2+\eta_m\right)\mu_m^2L_m^2d_m^2\\
&\quad\overset{1)}{\le} -\left(\frac{1}{2}\eta_0-L\eta_0^2\right)\mathbb{E}_i\left\|\nabla_{w_0}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2-\sum_{m=1}^{M}p_m\left(\frac{1}{2}\eta_m-2L\eta_m^2d_m\right)\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2\\
&\quad+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2+Q_1\\
&\quad\overset{2)}{\le} -\frac{1}{4}\eta_0\mathbb{E}_i\left\|\nabla_{w_0}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2-\sum_{m=1}^{M}p_m\frac{1}{4}\eta_m\mathbb{E}_{i,u}\left\|\nabla_{w_{m_t}}f\left(w_0^t,\mathbf{w}^t\right)\right\|^2\\
&\quad+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2+Q_1\\
&\quad\overset{3)}{\le} -\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}\mathbb{E}_i\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2\\
&\quad+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\mathbb{E}_i\|\mathbf{w}-\tilde{\mathbf{w}}\|^2+Q_1
\end{aligned}
\tag{15}
$$

where 1) to simplify the notation, define Q_1 to substitute the last row, 2) let $\eta_0 \le \frac{1}{4L}$ then $-\frac{1}{2}\eta_0+L\eta_0^2<-\frac{1}{4}\eta_0$, and let $\eta_m \le \frac{1}{4Ld_m}$, then $\frac{1}{2}\eta_0-L\eta_0^2\le\frac{1}{4}\eta_0$ and $\frac{1}{2}\eta_m-2L\eta_m^2d_m\le\frac{1}{4}\eta_m$, 3) uses the orthogonality of ∇f, i.e. $\left\|\nabla f\left(w_0,\mathbf{w}\right)\right\|^2=\left\|\nabla_{w_0}f\left(w_0,\mathbf{w}\right)\right\|^2+\sum_{m=1}^{M}\left\|\nabla_{w_m}f\left(w_0,\mathbf{w}\right)\right\|^2$.

Define the Lyapunov function to eliminate the client's delay.

Define a Lyapunov function.

$$
M^t=f\left(w_0^t,\mathbf{w}^t\right)+\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2 \tag{16}
$$

Taking expectation w.r.t. the activated client m_t, sample index i, and the random direction u.

$$
\begin{aligned}
&\mathbb{E}\left(M^{t+1}-M^{t}\right)\\
&=\mathbb{E}\left[f\left(w_0^{t+1},\mathbf{w}^{t+1}\right)+\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{t+1+1-i}-\mathbf{w}^{t+1-i}\right\|^2\right]\\
&-\mathbb{E}\left[f\left(w_0^{t},\mathbf{w}^{t}\right)+\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2\right]\\
&=\mathbb{E}\left[f\left(w_0^{t+1},\mathbf{w}^{t+1}\right)-f\left(w_0^{t},\mathbf{w}^{t}\right)\right]+\sum_{i=1}^{\tau}\theta_i\mathbb{E}\left\|\mathbf{w}^{t+1+1-i}-\mathbf{w}^{t+1-i}\right\|^2\\
&-\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2\\
&\overset{1)}{\leq}-\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^{t},\mathbf{w}^{t}\right)\right\|^2+Q_1\\
&+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\underbrace{\mathbb{E}\left\|\tilde{\mathbf{w}}^{t}-\mathbf{w}^{t}\right\|^2}_{a)}\\
&+\underbrace{\sum_{i=1}^{\tau}\theta_i\mathbb{E}\left\|\mathbf{w}^{t+1+1-i}-\mathbf{w}^{t+1-i}\right\|^2-\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2}_{b)}\\
&\overset{2)}{\leq}-\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^{t},\mathbf{w}^{t}\right)\right\|^2+Q_1\\
&+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\tau\sum_{i=1}^{\tau}\mathbb{E}\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2\\
&+\theta_1\mathbb{E}\left\|\mathbf{w}^{t+1}-\mathbf{w}^{t}\right\|^2+\sum_{i=1}^{\tau-1}(\theta_{i+1}-\theta_i)\mathbb{E}\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2-\theta_\tau\mathbb{E}\left\|\mathbf{w}^{t+1-\tau}-\mathbf{w}^{t-\tau}\right\|^2\\
&\leq-\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^{t},\mathbf{w}^{t}\right)\right\|^2+Q_1\\
&+\theta_1\mathbb{E}\left\|\mathbf{w}^{t+1}-\mathbf{w}^{t}\right\|^2\\
&+\sum_{i=1}^{\tau-1}\left(\theta_{i+1}-\theta_i+\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\tau\right)\mathbb{E}\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2\\
&-\left\{\theta_\tau-\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\tau\right\}\mathbb{E}\left\|\mathbf{w}^{t+1-\tau}-\mathbf{w}^{t-\tau}\right\|^2
\end{aligned}
\tag{17}
$$

where 1) plugging in Eq. 15, 2) plugging in a) and b).

For a) in Eq. 17:

$$
\mathbb{E}\left\|\tilde{\mathbf{w}}^{t}-\mathbf{w}^{t}\right\|^2\overset{1)}{\leq}\mathbb{E}\left\|\sum_{i=1}^{\tau}\left(\mathbf{w}^{i+1}-\mathbf{w}^{i}\right)\right\|^2\overset{2)}{\leq}\tau\sum_{i=1}^{\tau}\mathbb{E}\left\|\mathbf{w}^{t+1-i}-\mathbf{w}^{t-i}\right\|^2 \tag{18}
$$

where 1) applies Assumption 7 (uniformly bounded delay), 2) applies Cauchy-Schwarz inequality, i.e. $\left(\sum_{i=0}^{n-1} x_i\right)^2 = \left(\sum_{i=0}^{n-1} 1 \cdot x_i\right)^2 \leq n \sum_{i=0}^{n-1} x_i^2$.

For b) in Eq. 17:

$$\begin{aligned}
&\sum_{i=1}^{\tau} \theta_i \mathbb{E}\left\|\mathbf{w}^{t+1+1-i} - \mathbf{w}^{t+1-i}\right\|^2 - \sum_{i=1}^{\tau} \theta_i \mathbb{E}\left\|\mathbf{w}^{t+1-i} - \mathbf{w}^{t-i}\right\|^2 \\
&= \theta_1 \mathbb{E}\left\|\mathbf{w}^{t+1} - \mathbf{w}^{t}\right\|^2 + \sum_{i=1}^{\tau-1} (\theta_{i+1} - \theta_i) \mathbb{E}\left\|\mathbf{w}^{t+1-i} - \mathbf{w}^{t-i}\right\|^2 - \theta_\tau \mathbb{E}\left\|\mathbf{w}^{t+1-\tau} - \mathbf{w}^{t-\tau}\right\|^2
\end{aligned} \tag{19}$$

Let $\theta_1 = \tau^2\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M} p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right]$ and define the recursive formula for θ_i:

$$\theta_{i+1} = \theta_i - \tau Q_1 \tag{20}$$

if follows that:

$$\begin{aligned}
&\theta_\tau - \tau\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M} p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right] \\
&= \theta_1 - \tau^2\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M} p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right] = 0
\end{aligned} \tag{21}$$

Then Eq. 17 becomes

$$\begin{aligned}
&\mathbb{E}\left(M^{t+1} - M^t\right) \\
&\leq -\frac{1}{4}\min\left\{\eta_0, p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + Q_1 \\
&+ \tau^2\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M} p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right]\underbrace{\mathbb{E}\left\|\mathbf{w}^{t+1} - \mathbf{w}^t\right\|^2}_{c)} \\
&\overset{1)}{\leq} -\frac{1}{4}\min\left\{\eta_0, p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + Q_1 \\
&+ \tau^2\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M} p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right]\sum_{m=1}^{M} p_m\eta_m^2\left(2d_m\mathbf{G}_m^2 + \frac{1}{2}\mu_m^2 L_m^2 d_m^2\right) \\
&\overset{2)}{\leq} -\frac{1}{4}\min\left\{\eta_0, p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^t, \mathbf{w}^t\right)\right\|^2 + Q_1 \\
&+ Q_2
\end{aligned} \tag{22}$$

where 1) plugs in c), 2) simplify the notation by denoting the second line as Q_2.

For c):

$$
\begin{aligned}
&\mathbb{E}_{m_t,i,u}\left\|\mathbf{w}^{t+1}-\mathbf{w}^{t}\right\|^2 \\
&\overset{1)}{=} \mathbb{E}_{m_t,i,u}\eta_{m_t}^2\left\|\hat{\nabla}_{m_t}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2 \\
&\overset{2)}{\leq} \mathbb{E}_{m_t,i,u}\eta_{m_t}^2\left(2d_{m_t}\left\|\nabla_{w_{m_t}}f_i\left(w_0^t,\tilde{\mathbf{w}}^t\right)\right\|^2+\frac{1}{2}\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\right) \\
&\overset{3)}{\leq} \mathbb{E}_{m_t,i,u}\eta_{m_t}^2\left(2d_{m_t}\mathbf{G}_{m_t}^2+\frac{1}{2}\mu_{m_t}^2L_{m_t}^2d_{m_t}^2\right) \\
&\overset{4)}{\leq} \sum_{m=1}^{M}p_m\eta_m^2\left(2d_m\mathbf{G}_m^2+\frac{1}{2}\mu_m^2L_m^2d_m^2\right)
\end{aligned}
\tag{23}
$$

where 1) the update rule for the communication round, 2) applies Eq. 9 in Lemma 1, 3) applies Assumption 5 (bounded block-coordinated gradient), 4) applies Assumption 6 (independent client).

Bound the gradient $\nabla f\left(\boldsymbol{w}_0^t,\mathbf{w}^t\right)$

Start with Eq. 22:

$$
\begin{aligned}
&\mathbb{E}\left(M^{t+1}-M^t\right) \\
&\quad\leq -\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}\mathbb{E}\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2+Q_1+Q_2
\end{aligned}
\tag{24}
$$

Summing over the global iteration $t=0,1,...T-1$, arrange the equation and divide it by T from both sides.

$$
\begin{aligned}
&\frac{1}{4T}\min\left\{\eta_0,p_m\eta_m\right\}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\quad\leq \frac{\mathbb{E}\left(M^0-M^T\right)}{T}+Q_1+Q_2 \\
&\quad\overset{1)}{\leq} \frac{\mathbb{E}\left(f^0-f^*\right)}{T}+Q_1+Q_2
\end{aligned}
\tag{25}
$$

where 1) applies $\mathbb{E}\left(M^0-M^T\right)=f\left(w_0^0,\mathbf{w}^0\right)-f\left(w_0^T,\mathbf{w}^T\right)-\sum_{i=1}^{\tau}\theta_i\left\|\mathbf{w}^{T-i}-\mathbf{w}^{T-i}\right\|^2\leq f\left(w_0^0,\mathbf{w}^0\right)-f\left(w_0^T,\mathbf{w}^T\right)\leq f^0-f^*$, we use f^0 to denote $f\left(w_0^0,\mathbf{w}^0\right)$ and applying Assumption 1.

Dividing $\zeta=\frac{1}{4}\min\left\{\eta_0,p_m\eta_m\right\}$ from both sides:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\quad\leq \frac{\mathbb{E}\left(f^0-f^*\right)}{T\zeta}+\frac{Q_1}{\zeta}+\frac{Q_2}{\zeta} \\
&\quad\overset{1)}{\leq} \frac{\mathbb{E}\left(f^0-f^*\right)}{T\zeta} \\
&\quad+\frac{1}{\zeta}L\eta_0^2\sigma_0^2+\frac{1}{\zeta}\sum_{m=1}^{M}p_m2L\eta_m^2d_m\sigma_m^2+\frac{1}{\zeta}\sum_{m=1}^{M}p_m\frac{1}{4}\left(L\eta_m^2+\eta_m\right)\mu_m^2L_m^2d_m^2 \\
&\quad+\frac{1}{\zeta}\tau^2\left[\left(\frac{1}{2}\eta_0+L\eta_0^2\right)L_0^2+\sum_{m=1}^{M}p_m\left(\eta_m+2L\eta_m^2d_m\right)L_m^2\right]\sum_{m=1}^{M}p_m\eta_m^2\left(2d_m\mathbf{G}_m^2+\frac{1}{2}\mu_m^2L_m^2d_m^2\right)
\end{aligned}
\tag{26}
$$

where 1) plugs in a) and b).

To simplify the result, let $L_* = \max_m \{L, L_0, L_m\}$, $d_* = \max_m \{d_m\}$, $\eta_0 = \eta_m = \eta \leq \frac{1}{4L_* d_*}$, $\frac{1}{p_*} = \min_m p_m$, $\mu_* = \max_m \{\mu_m\}$, $\mathbf{G}_* = \max_m \{\mathbf{G}_m\}$, then $\zeta = \frac{1}{4}\min\{\eta_0, p_m\eta_m\} = \frac{\eta}{4p_*}$. Equation 26 can be further simplified:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(w_0^t, \mathbf{w}^t)\right\|^2 \\
&\overset{1)}{\leq} \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} \\
&+ \frac{4p_*}{\eta}L\eta_0^2\sigma_0^2 + \frac{4p_*}{\eta}\sum_{m=1}^{M}p_m 2L\eta_m^2 d_m\sigma_m^2 + \frac{4p_*}{\eta}\sum_{m=1}^{M}p_m\frac{1}{4}\left(L\eta_m^2 + \eta_m\right)\mu_m^2 L_m^2 d_m^2 \\
&+ \frac{4p_*}{\eta}\tau^2\left[\left(\frac{1}{2}\eta_0 + L\eta_0^2\right)L_0^2 + \sum_{m=1}^{M}p_m\left(\eta_m + 2L\eta_m^2 d_m\right)L_m^2\right]\sum_{m=1}^{M}p_m\eta_m^2\left(2d_m\mathbf{G}_m^2 + \frac{1}{2}\mu_m^2 L_m^2 d_m^2\right) \\
&\overset{2)}{\leq} \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} \\
&+ \frac{4p_*}{\eta}L\eta_0^2\sigma_0^2 + \frac{4p_*}{\eta}\sum_{m=1}^{M}p_m 2L\eta_m^2 d_m\sigma_m^2 + \frac{4p_*}{\eta}\sum_{m=1}^{M}p_m\frac{1}{4}\left(L\eta_m^2 + \eta_m\right)\mu_m^2 L_m^2 d_m^2 \\
&+ \frac{4p_*}{\eta}\tau^2\left[\left(\frac{1}{2}\eta_0 + \frac{1}{4}\eta_0\right)L_0^2 + \sum_{m=1}^{M}p_m\left(\eta_m + \frac{1}{2}\eta_m\right)L_m^2\right]\sum_{m=1}^{M}p_m\eta_m^2\left(2d_m\mathbf{G}_m^2 + \frac{1}{2}\mu_m^2 L_m^2 d_m^2\right) \\
&\overset{3)}{\leq} \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} \\
&+ 4p_* L_*\eta\sigma_*^2 + 8p_* L_*\eta d_*\sigma_*^2 + p_* L_*\eta\mu_*^2 L_*^2 d_*^2 + p_* L_*\mu_*^2 L_*^2 d_*^2 \\
&+ p_*\tau^2\left(\frac{9}{2}\eta\right)L_*^2\eta\left(4d_*\mathbf{G}_*^2 + \mu_*^2 L_*^2 d_*^2\right) \\
&\overset{4)}{\leq} \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} \\
&+ \eta\left(4p_* L_*\sigma_*^2 + 8p_* L_* d_*\sigma_*^2 + p_* L_*\mu_*^2 L_*^2 d_*^2\right) \\
&+ \eta^2\left(\frac{9}{2}p_*\tau^2 L_*^2\left(4d_*\mathbf{G}_*^2 + \mu_*^2 L_*^2 d_*^2\right)\right) \\
&+ \mu_*^2\left(p_* L_* L_*^2 d_*^2\right) \\
&\overset{5)}{\leq} \frac{4p_*\mathbb{E}\left(f^0 - f^*\right)}{T\eta} \\
&+ \eta\left(4p_* L_*\sigma_*^2 + 8p_* L_* d_*\sigma_*^2 + p_* L_*\mu_*^2 L_*^2 d_*^2\right) \\
&+ \eta^2\left(18p_*\tau^2 L_*^2 d_*\mathbf{G}_*^2 + 5p_*\tau^2 L_*^2\mu_*^2 L_*^2 d_*^2\right) \\
&+ \mu_*^2\left(p_* L_*^3 d_*^2\right)
\end{aligned}
\tag{27}
$$

where 1) plugs in the above variables for ζ, 2) simplify by $\eta_0 \leq \frac{1}{4L}$ and $\eta_m \leq \frac{1}{4Ld_m}$, 3) plugs in the variables η, μ_*, L_*, 4) collect the η and μ, 5) simply $\frac{9}{2} < 5$.

The proof of Theorem 1 is complete. □

Suppose we set $\eta = \frac{1}{\sqrt{T}}$, and $\mu = \frac{1}{\sqrt{T}}$, the above equation becomes:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2 \\
&\overset{5)}{\leq} \frac{1}{\sqrt{T}}\left[4p_*\mathbb{E}\left(f^0-f^*\right)+4p_*L_*\sigma_*^2+8p_*L_*d_*\sigma_*^2\right] \\
&\quad+\frac{1}{T}\left(18p_*\tau^2L_*^2d_*\mathbf{G}_*^2+5p_*\tau^2\mu_*^2L_*^4d_*^2+p_*L_*^3d_*^2\right) \\
&\quad+\frac{1}{T^{\frac{3}{2}}}\left(p_*L_*^3d_*^2\right)
\end{aligned} \tag{28}$$

Therefore,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(w_0^t,\mathbf{w}^t\right)\right\|^2=\mathcal{O}\left(\frac{d}{\sqrt{T}}\right) \tag{29}$$

where $d=d_*=\max_m\left\{d_m\right\}$ (for clear notation), T is the number of iterations.

The proof of Corollary 1 is complete. □

Appendix 2: Discussion on threat model where the attacker deviates from the protocol

The "honest" threat model corresponds to a scenario in which all participants strictly adhere to the prescribed protocol. In contrast, we introduced the "malicious" threat model, allowing the attacker to deviate from the specified learning protocol. There are various targets for the attacker to deviate from the protocol, such as impeding the learning process (Fang et al., 2020), injecting a backdoor into the model (Liu et al., 2020) or influencing the prediction outcomes (Fu et al., 2022).

Deviation from the protocol by participants is considered less realistic in practical applications within the context of VFL. As VFL participants are typically accountable large institutions, the detection of malicious conduct from these entities could lead to significant reputational and financial losses. Consequently, the substantial risks generally outweigh the potential gains from engaging in malicious behavior.

Our framework can defend against some attacks in scenarios where the attacker deviates from the protocol. Specifically, if the attack needs access to accurate gradient information, our framework remains resilient. For example, it can defend against backdoor attacks utilizing gradient replacement (Liu et al., 2020), since the attacker cannot acquire the accurate gradient. However, our framework is unable to thwart attacks unrelated to gradient information, such as the active manipulation of the optimization process during training to influence model predictions (Fu et al., 2022).

Appendix 3: Supplementary experiment details

The algorithm for Syn-ZOO-VFL:

Table 6 Computational cost for propagation of the models

Asynchronous frameworks	Client's propagation	Sun of the Clients' propagation time per epoch (s)	Server propagation	Server propagation time per epoch (s)
VAFL	F + B[a]	1.61 ± 0.02	F + B	4.32 ± 0.08
ZOO-VFL	F + F	0.69 ± 0.39	F + F + F	2.69 ± 0.39
VFL-Cascaded	F + F	0.76 ± 0.02	F + F + B	5.18 ± 0.14

[a] F, forward propagation; B, backward propagation

Algorithm 2 The Synchronous Modification of ZOO-VFL (Zhang et al., 2021)

Initialize variables for workers $m \in [M]$
for $t = 0, ..., T-1$ **do**
  Random sample a sample i (or batch B).
  for client m in $[M]$ in parallel **do**
    Client m compute and send $h_{m,i} = h_m(w_m; x_{m,i})$ and $\hat{h}_{m,i} = h_m(w_m + \mu \mathbf{u}_{m,i}; x_{m,i})$ to the server.
    The server calculates $\delta_m = f_i(w_0, ...\hat{h}_{m,i}...) - f_i(w_0, h_{1,i}, ...h_{M,i})$ and send back to the client.
    Client m calculate the stochastic gradient w.r.t. its local parameter w_m with the δ_m received from the server: $\hat{\nabla}_{w_m} f_i(\cdot) = \frac{\phi(d_m)}{\mu} \delta_m \mathbf{u}_{m,i}$
    Client m update its parameter with gradient descent $w_m \leftarrow w_m - \eta_m \hat{\nabla}_{w_m} f_i(\cdot)$
  end for
  The server calculates its local stochastic gradient estimation via $\hat{\nabla}_{w_0} f_i(\cdot) = \frac{\phi(d_0)}{\mu} \left[f_i(w_0 + \mu \mathbf{u}_{0,i}, ...\hat{h}_{m,i}...) - f_i(w_0, h_{1,i}, ...h_{M,i}; y_i) \right] \mathbf{u}_{0,i}$
  The server update its local parameter with gradient descent $w_0 \leftarrow w_0 - \eta_0 \hat{\nabla}_{w_0} f_i(\cdot)$
end for

Appendix 4: Experiments on different aspects of VFL-cascaded

Computation cost

To facilitate the operation of ZOO on the client side and FOO on the server side, the server within the VFL-Cascaded architecture is required to undertake additional computational tasks. The primary distinction between our framework and others lies in the number of forward and backward propagations executed by the participants.

We assess propagation counts and propagation time consumption among the asynchronous VFL frameworks in Table 6. The VAFL (Chen et al., 2020) is optimized with FOO, with both the client and server executing a singular forward and backward propagation. The ZOO-VFL (Zhang et al., 2021) undergoes an update through ZOO. The client involves an additional forward propagation on the perturbed parameter, while the server incorporates two extra forward propagations-one on the client's perturbed inputs and another on its

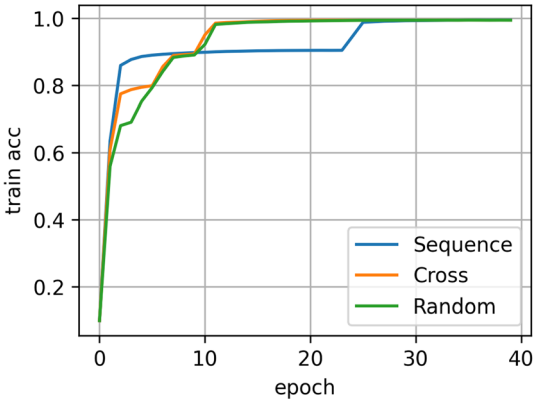


Fig. 8 Convergence of VFL-cascaded with different feature separation

locally perturbed parameter. No backward propagation is required. In our framework, VFL-Cascaded, the server facilitates its local optimization through a backward propagation and aids the client's preparation with an additional propagation.

We recorded the propagation times for both clients and the server in the base experiment on MNIST using the MLP model with four clients, as outlined in Sect. 6. All experiments are run through five independent runs.

As indicated in the Table 6, our framework has a slightly increased propagation time for the server compared to other frameworks. However, the observed difference is small.

Experiment on different feature separation

We conducted experiments involving various feature separations, adhering to the experimental setup employed in the base experiment on MNIST, utilizing the MLP model with four clients. The original feature separation is contingent on the first dimension of the image, whereby the first client receives the upper quarter of the image, and the second client obtains the second quarter, etc. Two additional separations, namely "cross" and "random," are introduced in this experiment. In the cross separation, the image is divided by a cross in the middle, assigning one corner to each client. For the random separation, each client randomly selects non-overlapping $\frac{1}{4}$ features from the entire set of features.

The convergence experiment results for different feature separation methods are presented in Fig. 8.

Notably, no significant differences are observed among the various feature separation methods.

Experiment on different model split

We performed additional experiments involving different model splits, specifically conducting an ablation study on model splitting in the CIFAR10 experiment.

The majority of the experimental details align with those outlined in Sect. 6 for the CIFAR10 experiment. In this particular experiment, we add a different model split where

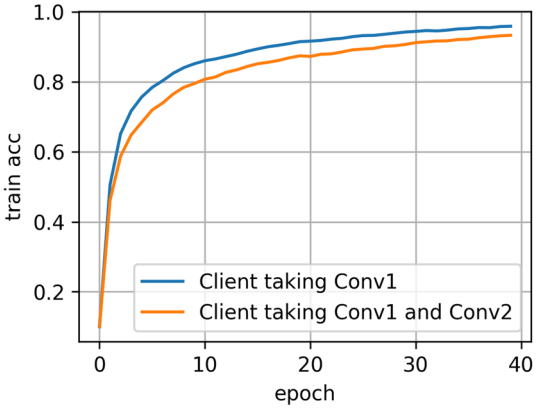


Fig. 9 Convergence of different model splitting

Table 7 Test accuracy (%) for different model splits

	Test accuracy
Client taking Conv1	87.2±0.6
Client taking Conv1 and Conv2	84.8±0.4

the client is responsible for the first two layers of the ResNet18 (conv1, conv2), while the server manages the remaining components.

The convergence results are presented in Fig. 9 and the corresponding test accuracy is presented in Table 7.

Notably, the client handling two layers bears a heavier parameter load optimized with ZOO, leading to a slower convergence rate, which is aligned with theoretical expectations.

Supplementary experiment on real-world dataset

We conducted this experiment using the Give Me Some Credit (GMSC) dataset (Credit Fusion, 2011), a real-world dataset containing information from 250,000 anonymous borrowers. The aim of this dataset is to predict instances where individuals fail to repay an installment, extending beyond 90 days from the due date within a 2-year timeframe.

The dataset consists of 10 features and 1 label for each sample. We assumed that there were two clients in the VFL. The first 5 features belong to the first client, while the remaining features belong to the second client. To address the substantial imbalance between positive and negative classes, downsampling was applied to the negative class, equalizing their sizes with the positive class. Subsequently, the dataset was partitioned into a training set, comprising 75% of the data, and a testing set, comprising the remaining 25%.

We utilized a Logistic Regression (LR) model on the clients, with the server aggregating predictions by summing the logits from both clients. The batch size was fixed at 64. Learning rates for all frameworks were chosen through a grid search within the range [0.1, 0.01, 0.001, 0.0001]. The optimal value for the hyperparameter μ was determined as 0.001 from the set [0.1, 0.001, 0.0001, 0.0001] using grid search. The model was trained for a total of 50 epochs. The convergence curve is depicted in Fig. 10, and the test accuracy results for those frameworks are detailed in Table 8. The results suggest that all optimization methods

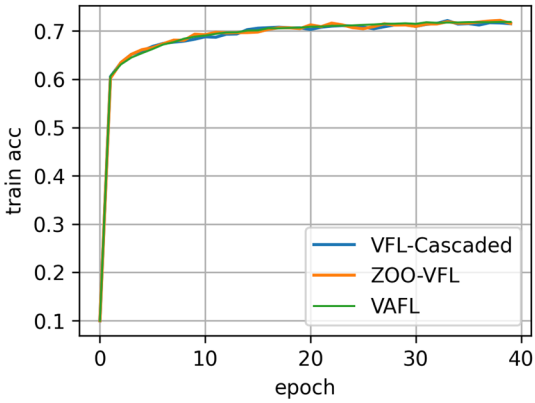


Fig. 10 Convergence of VFL on GMSC dataset

Table 8 Test accuracy (%) for the experiment on real-world dataset

	Test accuracy
VAFL	71.6 ± 0.03
ZOO-VFL	72.0 ± 0.5
VFL-Cascaded	72.3 ± 0.6

perform effectively for this task, likely due to the model's simplicity, making it easy to optimize.

Author Contributions Ganyu Wang developed the theory and conducted the experiment. Ganyu Wang, Qingsong Zhang, and Xiang Li wrote the paper. Xiang Li, Boyu Wang, and Bin Gu verified the theory and the experiment. Boyu Wang, Bin Gu, and Charles X. Ling supervised the project.

Funding Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grants program.

Data availability All datasets used in this research are public datasets.

Code availability Code will be made public if accepted.

Declarations

Conflict of interest Western University, Xidian University, Mohamed bin Zayed University of Artificial Intelligence.

Ethics approval Waive. No ethics approval is needed for this research.

References

Ahmad, A., Luo, W., & Robles-Kelly, A. (2023). Robust federated learning under statistical heterogeneity via hessian-weighted aggregation. *Machine Learning, 112*(2), 633–654.

Badar, M., Nejdl, W., & Fisichella, M. (2023). FAC-fed: Federated adaptation for fairness and concept drift aware stream classification. *Machine Learning* 1–26.
Casado, F. E., Lema, D., Iglesias, R., Regueiro, C. V., & Barro, S. (2023). Ensemble and continual federated learning for classification tasks. *Machine Learning* 1–41.
Castiglia, T.J., Das, A., Wang, S., & Patterson, S. (2022). Compressed-VFL: Communication-efficient learning with vertically partitioned data. In *International conference on machine learning* (pp. 2738–2766). PMLR
Castiglia, T., Wang, S., & Patterson, S. (2022). Flexible vertical federated learning with heterogeneous parties. arXiv preprint arXiv:2208.12672.
Chen, T., Jin, X., Sun, Y., & Yin, W. (2020). VAFL: A method of vertical asynchronous federated learning. arXiv preprint arXiv:2007.06081.
Commission, E. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (General Data Protection Regulation). OJ, 2016-04-27.
Credit Fusion, W.C. (2011). Give Me Some Credit. Kaggle. https://kaggle.com/competitions/GiveMeSomeCredit.
Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
Fang, M., Cao, X., Jia, J., & Gong, N. (2020). Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)* (pp. 1605–1622).
Fang, W., Zhao, D., Tan, J., Chen, C., Yu, C., Wang, L., Wang, L., Zhou, J., & Zhang, B. (2021). Large-scale secure XGB for vertical federated learning. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 443–452).
Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1322–1333).
Fu, C., Zhang, X., Ji, S., Chen, J., Wu, J., Guo, S., Zhou, J., Liu, A. X., & Wang, T. (2022). Label inference attacks against vertical federated learning. In *31st USENIX security symposium (USENIX Security 22)*, Boston, MA.
Gao, X., Jiang, B., & Zhang, S. (2018). On the information-adaptive variants of the ADMM: An iteration complexity perspective. *Journal of Scientific Computing, 76*(1), 327–363.
Ghadimi, S., & Lan, G. (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization, 23*(4), 2341–2368.
Gu, B., Xu, A., Huo, Z., Deng, C., & Huang, H. (2021). Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. *IEEE Transactions on Neural Networks and Learning Systems*.
Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677.
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
Hu, Y., Niu, D., Yang, J., & Zhou, S. (2019). FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2232–2240).
Jin, X., Chen, P.-Y., Hsu, C.-Y., Yu, C.-M., & Chen, T. (2021). CAFE: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems, 34*, 994–1006.
Kairouz, P., McMahan, H., Avent, B., Bellet, A., Bennis, M., Bhagoji, A., Bonawitz, K., Charles, Z., Cormode, G., & Cummings, R., et al. (2019). Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977.
Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning* (pp. 5132–5143). PMLR.
Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report.
LeCun, Y., Cortes, C., & Burges, C. (2010). Mnist handwritten digit database. ATT Labs [Online]. http://yann.lecun.com/exdb/mnist**2**.
Li, L., Zhan, D.-c., & Li, X.-c. (2022). Aligning model outputs for class imbalanced non-IID federated learning. *Machine Learning* 1–24.
Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37*(3), 50–60.

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

Li, X., Jiang, M., Zhang, X., Kamp, M., & Dou, Q. (2021). FedBN: Federated learning on non-IID features via local batch normalization. In *International conference on learning representations*. https://openreview.net/pdf?id=6YEQUn0QICG.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero, A. O., III., & Varshney, P. K. (2020). A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine, 37*(5), 43–54.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., & Amini, L. (2018). Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in neural information processing systems* (vol. 31).

Liu, Y., Kang, Y., Li, L., Zhang, X., Cheng, Y., Chen, T., Hong, M., & Yang, Q. (2019). A communication efficient vertical federated learning framework. Scanning Electron Microsc Meet at.

Liu, Y., Ma, Z., Liu, X., Ma, S., Nepal, S., Deng, R. H., & Ren, K. (2020). Boosting privately: Federated extreme gradient boosting for mobile crowdsensing. In *2020 IEEE 40th international conference on distributed computing systems (ICDCS)* (pp. 1–11). IEEE.

Liu, Y., Yi, Z., & Chen, T. (2020). Backdoor attacks and defenses in feature-partitioned collaborative learning. arXiv preprint arXiv:2007.03608.

Luo, X., Wu, Y., Xiao, X., & Ooi, B. C. (2021). Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th international conference on data engineering (ICDE)* (pp. 181–192). IEEE.

Makhija, D., Han, X., Ho, N., & Ghosh, J. (2022). Architecture agnostic federated learning for neural networks. arXiv preprint arXiv:2202.07757.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172).

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

Mishchenko, K., Malinovsky, G., Stich, S., & Richtárik, P. (2022). Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International conference on machine learning* (pp. 15750–15769). PMLR.

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

Ranbaduge, T., & Ding, M. (2022). Differentially private vertical federated learning. arXiv preprint arXiv:2211.06782.

Sabater, C., Bellet, A., & Ramon, J. (2022). An accurate, scalable and verifiable protocol for federated differentially private averaging. *Machine Learning, 111*(11), 4249–4293.

Shi, J., Bian, J., Richter, J., Chen, K.-H., Rahnenführer, J., Xiong, H., & Chen, J.-J. (2021). Modes: Model-based optimization on distributed embedded systems. *Machine Learning, 110*(6), 1527–1547.

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1310–1321).

Sun, J., Yang, X., Yao, Y., & Wang, C. (2022). Label leakage and protection from forward embedding in vertical federated learning. arXiv preprint arXiv:2203.01451.

Vepakomma, P., Gupta, O., Swedish, T., & Raskar, R. (2018). Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564.

Wang, Y., Lin, L., & Chen, J. (2022). Communication-efficient adaptive federated learning. arXiv preprint arXiv:2205.02719.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q., & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15*, 3454–3469.

Wei, K., Li, J., Ma, C., Ding, M., Wei, S., Wu, F., Chen, G., & Ranbaduge, T. (2022). Vertical federated learning: Challenges, methodologies and experiments. arXiv preprint arXiv:2202.04309.

Weng, H., Zhang, J., Xue, F., Wei, T., Ji, S., & Zong, Z. (2020). Privacy leakage of real-world vertical federated learning. arXiv preprint arXiv:2011.09290.

Yang, K., Fan, T., Chen, T., Shi, Y., & Yang, Q. (2019). A quasi-newton method based vertical federated learning framework for logistic regression. arXiv preprint arXiv:1912.00513.

Zhang, Q., Gu, B., Dang, Z., Deng, C., & Huang, H. (2021). Desirable companion for vertical federated learning: New zeroth-order gradient based algorithm. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2598–2607).

Zhang, Q., Gu, B., Deng, C., & Huang, H. (2021). Secure bilevel asynchronous vertical federated learning with backward updating. In *Proceedings of the AAAI conference on artificial intelligence* (vol. 35, pp. 10896–10904).

Zhao, B., Mopuri, K.R., & Bilen, H. (2020). iDLG: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610.

Zhou, J., Chen, C., Zheng, L., Wu, H., Wu, J., Zheng, X., Wu, B., Liu, Z., & Wang, L. (2020). Vertically federated graph neural network for privacy-preserving node classification. arXiv preprint arXiv:2005.11903.

Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. In *Advances in neural information processing systems* (vol. 32).

Authors and Affiliations

Ganyu Wang[1] · Qingsong Zhang[2] · Xiang Li[1] · Boyu Wang[1] · Bin Gu[3] · Charles X. Ling[1]

✉ Charles X. Ling
charles.ling@uwo.ca

Ganyu Wang
gwang382@uwo.ca

Qingsong Zhang
qszhang1995@gmail.com

Xiang Li
lxiang2@uwo.ca

Boyu Wang
bwang@csd.uwo.ca

Bin Gu
jsgubin@gmail.com

[1] Western University, London, ON N6A 3K7, Canada

[2] Xidian University, Xi'an 710126, Shaanxi, China

[3] Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, United Arab Emirates