CE-VFAL: A Novel Framework for Communication-Efficient Vertical Federated Adversarial Learning

Tianxing Man[1*], **Jinjie Fang**[2*], **Ganyu Wang**[3], **Yu Bai**[2], **Zhaogeng Liu**[2], **Bin Gu**[2†] and **Yi Chang**[2,4,5†]

[1]Inner Mongolia University, China
[2]Jilin University, China
[3]Western University, Canada
[4]International Center of Future Science, Jilin University, China
[5]Engineering Research Center of Knowledge-Driven Human-Memory Intelligence, MOE, China
{mantx626, userelaina, jsgubin}@gmail.com, gwang382@uwo.ca, {fangjj24, zgliu20}@mails.jlu.edu.cn, yichang@jlu.edu.cn

Abstract

Vertical Federated Learning (VFL) involves multiple participants collaborating to train machine learning models on distinct feature sets from the same data samples. This training paradigm with distributed updating focuses on secure and efficient communication. Nevertheless, the trained models exhibit heightened vulnerability to adversarial attacks during inference, which can provoke misclassification. Adversarial Training (AT), which involves exposing models to intentionally crafted misleading examples during training, is widely regarded as the most effective method for enhancing model robustness. However, the significant communication costs entailing such example generation within the VFL context pose an open challenge to developing a Vertical Federated Adversarial Learning (VFAL) framework. To this end, we introduce a Communication-Efficient Vertical Federated Adversarial Learning framework, named CE-VFAL. CE-VFAL framework incorporates the lazy propagation principle, confining most propagations to client models during adversarial updates, thereby minimizing frequent client-server interactions. Moreover, CE-VFAL seamlessly integrates Zeroth Order Optimization (ZOO) into communication, effectively reducing communication load by transmitting the loss difference derived from the raw and perturbed embeddings for multiple point estimation. Furthermore, our theoretical analysis demonstrates the sublinear convergence rate by containing the errors caused by multi-source approximate gradients. Extensive experiments corroborate the robust performance while significantly reducing communication costs.

1 Introduction

Federated Learning (FL) [Kairouz *et al.*, 2021; Yang *et al.*, 2019; McMahan *et al.*, 2016] represents a distributed learning paradigm that empowers multiple clients to collectively train and utilize a machine learning (ML) model without sharing their data directly. Conventionally, most FL research considers the Horizontal FL (HFL) setting, where clients hold different data samples with the same feature space. In contrast, **vertical FL (VFL)** tackles the scenarios where clients have identical samples but disjoint feature spaces [Liu *et al.*, 2024; Wei *et al.*, 2022]. A typical VFL model comprises a top model maintained by a server and multiple bottom models, each residing at participating clients. Throughout the training, individual clients compute local embeddings of data features using their respective bottom models and transmit them to the server through a communication channel. The server aggregates these embeddings to update the top model and then sends gradients back to clients to update bottom models accordingly. Therefore, recent researches focuses on studying communication efficiency, involving reducing the cost of coordination and compressing the data transmitted between parties, such as multiple client updates [Zhang *et al.*, 2022], asynchronous coordination [Li *et al.*, 2020], and data compression [Castiglia *et al.*, 2022; Li *et al.*, 2020].

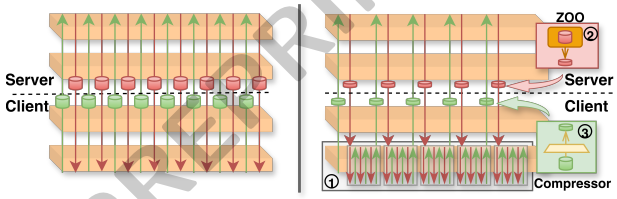

Figure 1: Communication cost for one-time adversarial sample generation (VFL+PGD (left) vs. CE-VFAL (right)).

Threat model. ML models have demonstrated vulnerability to **adversarial attacks**, carefully crafted inputs designed to induce misclassification *during inference*. Distributed deployment inherently exposes VFL model to additional security threats [Huang *et al.*, 2024; Duanyi *et al.*, 2023]. Adversarial attacks in VFL can manifest in multiple forms during inference: via third-party adversary intercepting and al-

tering embeddings during client-server communication [Duanyi *et al.*, 2023], or through malicious or colluding clients perturbing local features of raw data [Pang *et al.*, 2022; Qiu *et al.*, 2022]. The details of the threat model are presented in Appendix A.3. These diverse attacks underscore the unique security challenges in VFL systems, motivating the urgent need to address the VFL robustness problem.

Currently, much effort has been made to improve models' resistance to such perturbations in centralized training scenarios. Among these studies, the Adversarial Training (AT) scheme has demonstrated significant potential in bolstering the robustness of ML models[Tramèr *et al.*, 2018]. AT is a min-max robust training method that minimizes the worst-case training loss at adversarially perturbed examples [Madry *et al.*, 2017]. The deployment of AT in the FL paradigm termed Federated Adversarial Learning (FAL), has garnered attention, with a particular focus on HFL scenarios, where each participant maintains a complete copy of the model [Li *et al.*, 2023]. However, in VFL scenarios, a single global model is partitioned and distributed among the server and clients, resulting in a different architecture for **Vertical Federated Adversarial Learning (VFAL)**.

Observation. Due to layer-wise distributed deployment, VFAL presents unique challenges in communication efficiency. Adversarial sample generation during AT is computationally intensive, requiring sequential forward and backward propagation to calculate gradients concerning the input for iterative refinement [Madry *et al.*, 2017]. As figure 1-left shows, VFAL using PGD-10 (Projected Gradient Descent) requires about 10 times more communication cost than regular VFL due to 10 iterations needed to generate adversarial example. Consequently, significant communication costs pose an unacceptable barrier between robustness and communication efficiency, hindering VFAL framework development.

Recognizing these challenges, we propose a novel framework named ***Communication-Efficient Vertical Federated Adversarial Learning (CE-VFAL)*** via introducing the lazy propagation principle and Zeroth Order Optimization (ZOO)[Wang *et al.*, 2024], aimed at enhancing VFL model robustness through communication-efficiency AT: 1) By integrating lazy propagation, CE-VFAL reduces the frequency of interactions between clients and the server by confining the majority of propagation in client parties during adversarial updates (Figure 1-①); 2) CE-VFAL incorporates ZOO into the communication phase to reduce the communication load of the backward propagation by transmitting only the loss difference derived from the raw and perturbed embeddings (Figure 1-②); 3) We demonstrate the compatibility of CE-VFAL with communication compression techniques, which can reduce the size of forwarded-transmitted messages (Figure 1-③). CE-VFAL is the first theoretically verified VFL framework that applies AT. Inspired by the optimal control theorem [Li *et al.*, 2018; Li and Hao, 2018], we constrain the multi-source approximate gradients arising from frozen slack variables, gradient estimation, and communication compression. Subsequently, we demonstrate the sublinear convergence rate of model updates through the training process.

Contributions. Our contributions are as follows:

1. We propose the communication-efficient CE-VFAL framework fundamentally reduces client-server interactions through lazy propagation, and integrates advanced compression and ZOO to minimize message sizes, achieving substantial communication efficiency.
2. We provide the first theoretical convergence analysis for VFAL, rigorously characterizing the interaction between multi-source approximation gradients and establishe an $\mathcal{O}(1/\sqrt{\mathcal{K}})$ convergence rate over $\mathcal{K}$ iterations, matching standard VFL guarantees.
3. Experiments across datasets and architectures verify that CE-VFAL achieves state-of-the-art robustness with substantially lower communication cost than prior VFL adversarial training methods.

2 Preliminaries

We consider a classifier $F : X \to Y$ parameterized by weights Θ and predicting a class $\hat{y} := F(x) := \arg\max_{y \in Y} f_y(x)$ for every input $x \in X$ with label $y \in Y$ in D, where f is a neural network. Notations are summarized in Appendix 2.1. We sort the indexes of sample, client, and network layer, for example, $x_{i,c,t}$ represents i-th sample, c-th client and t-th layer. Sample's index has higher priority than client's, and client's has higher priority than layer's.

Vertical Federated Learning. CE-VFAL applies to various VFL architectures. This paper focuses on the $splitVFL_c$ scenario, which typically decomposes neural network f into client models $\{f_1, f_2, \ldots, f_C\}$ parameterized by $\theta = \{\theta_1, \theta_2, \ldots, \theta_C\}$, which operate only on local data, and a sever model f_s parameterized by ψ. The server party holds Y. Let $\Theta = (\theta, \psi)$, the global loss can be as:

$$\begin{aligned} &\mathcal{L}(X, Y; \Theta) \\ &:= \frac{1}{n} \sum_{i=1}^{n} \ell \left(f_s \left(f_1(x_{i,1}; \theta_1), \ldots, f_C(x_{i,C}; \theta_C); \psi \right), y_i \right), \end{aligned} \tag{1}$$

where ℓ is the loss function, C is the number of client parties, and n is the sample size.

Adversarial Training. AT optimizes the parameters Θ of a neural network f on adversarially perturbed inputs, thereby increasing robustness against test-time attacks [Madry *et al.*, 2017]. This process involves solving a minimax problem on the cross-entropy loss:

$$\min_{\Theta} \mathbb{E}_{(x,y)\sim D} \left[\max_{\|\eta\| \leq \epsilon} \mathcal{L}(x + \eta, y; \Theta) \right], \tag{2}$$

where η is the adversarial perturbation, and its magnitude is constrained by $\|\eta\| \leq \epsilon$, where ϵ is a non-negative scalar that bounds the size of the perturbation. For the inner maximization objective in Equation (2), PGD takes the following step (with step size α) in each iteration:$\eta_{t+1} = \Pi_{\|\eta\| \leq \epsilon}(\eta_t + \alpha \frac{g(\eta_t)}{\|g(\eta_t)\|})$, where $g(\eta_t) = \nabla_\eta \mathcal{L}(x + \eta, y; \Theta)$ is the gradient respect to η, $\Pi_{\|\eta\| \leq \epsilon}$ is a projection to the ϵ-ball.

Definition 1. ***Vertical Federated Adversarial Learning.*** *Inspired by the idea of viewing AT as a dynamic system [Weinan, 2017], we can describe a T-layer deep neural network by the recursion $x_{t+1} = f_t(x_t, \theta_t)$, for $t = 0, \ldots, T-1$, where $x_t \in \mathbb{R}^{n_t}$ represents the states (i.e., the output of the t-th layer), $f_t : \mathbb{R}^{n_t} \times \mathbb{R}^{m_t} \to \mathbb{R}^{n_{t+1}}$ is the state transition map, $\theta_t \in \mathbb{R}^{m_t}$ are the trainable control parameters, and the initial conditions are provided by the inputs to the network, $x_{i,0}$. Rewrite (2) as the subsequent optimal control problem:*

$$\begin{aligned} \min_{\theta_1,\ldots,\theta_T} \max_{\eta_1,\ldots,\eta_S} \quad & \sum_{i=1}^{n} \ell(x_{i,T}, y_i) + \sum_{i=1}^{n} \sum_{t=0}^{T-1} R_t(x_{i,t}, \theta_t) \\ \textit{subject to} \quad & x_{i,t+1} = f_t(x_{i,t}, \theta_t), \quad t = 1, \ldots, T-1 \\ & x_{i,1} = f_0(x_{i,0} + \eta_i, \theta_0), \end{aligned} \tag{3}$$

where R_t is a potential regularizer for the t-th layer.

3 Methodology

In this section, we present CE-VFAL that substantially reduces both communication rounds and message sizes while maintaining model robustness. CE-VFAL introduces three key innovations through a carefully designed hybrid optimization strategy: (1) integration of ZOO to reduce backward communication load, (2) implementation of uniform scale compressor to decrease forward message size, and (3) adaptation of lazy propagation to minimize communication rounds during adversarial sample generation. For the convenience of subsequent analysis, we rewrite loss function as $\mathcal{A}_i(\eta, \psi, \theta)$.

Assumptions. The formal definition and detailed discussion of the assumptions are in Appendix 2.2. We make several crucial assumptions: the functions f_t, f_c, $\mathcal{L}$, and R_t are K-Lipschitz continuous in x, uniformly with respect to θ and ψ, the gradient of the adversarial loss function, $\nabla_\eta \mathcal{A}_i(\eta, \psi, \theta)$,$\nabla_\psi \mathcal{A}_i(\theta, \psi, \theta)$ and $\nabla_\eta \mathcal{A}_i(\eta, \psi, \theta)$ satisfy Lipschitz conditions (Assumption 1); the adversarial loss function $\mathcal{A}_i(\eta, \psi, \theta)$ possesses an unbiased gradient (Assumption 2) and is characterized by bounded Hessian matrices H_ψ and H_θ (Assumption 3), as well as bounded block-coordinate gradients Q_ψ and Q_θ (Assumption 4); $\mathcal{A}_i(\eta, \psi, \theta)$ exhibits z-strong concavity with respect to η (Assumption 5).

3.1 ZOO for Backpropagation

To reducing communication rounds, CE-VFAL reduces the backward message size by employing ZOO (Details in Appendix B.1) within a cascading hybrid optimization strategy. ZOO is applied only during the communication phase, while the rest of the model uses First-Order (FO) gradients. We estimate gradients by sampling random perturbations and evaluating their impact with average random gradient estimation (Avg-RandGradEst).

$$\hat{\nabla}_{x_{t_c}} f_s(\psi, x_{t_c}) = \frac{1}{q} \sum_{j=1}^{q} \underbrace{[f_s(x_{t_c} + \mu u_j) - f_s(x_{t_c})]}_{\delta^j} \frac{u_j}{\mu}, \tag{4}$$

where t_c is the layer number of c-th client, x_{t_c} represents the output embeddings of c-th client and μ is the smoothing parameter. The distribution $\mathcal{P}$ of u follows the multivariate normal distribution $N(0, I)$, and q is the number of sampling times. Each δ_j is the loss difference derived from the original and perturbed embeddings. Both server and clients synchronize their random sequences by sharing a common random seed at the start of training. According to (4), the server only needs to communicate a set $\delta = \{\delta^1, \delta^2, \ldots, \delta^q\}$ back to the clients for the gradient estimization. In contrast, FO gradients require transmitting partial derivatives for the entire embedding. The size $\|\delta\| = q$ is significantly smaller than the size of the partial derivatives, equal to $\|x_{t_c}\|$ (that is $\|x_{t_c}\| * B$ for one batch).

Estimated gradient. ZOO is used to estimate gradients rather than exact calculations, leading to an "Estimated Gradient." The estimation error due to the finite difference approximation is bounded with Lemma 1.

Lemma 1. ***Bound Zeroth-Order Error.*** *For arbitrary $f \in C_L^1(R^d)$, we have:*
1) $f_\mu(x)$ is continuously differentiable, its gradient is L_μ-Lipschitz continuous with $L_\mu \leq L$:

$$\nabla f_\mu(x) = \mathbb{E}_u[\hat{\nabla} f(x)],$$

where u is drawn from the uniform distribution over the unit Euclidean sphere, $f_\mu(x) = \mathbb{E}(f(x + \mu u))$ is the smooth approximation of f.
2) For any $x \in R_d$, the following inequalities satisfy:

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

Proof of this lemma is provided in [Liu *et al.*, 2018; Gao *et al.*, 2018].

3.2 Compression for Forward Propagation

To further alleviate the communication load, we showcase the compatibility of CE-VFAL with communication compression techniques (Details in Appendix 1.6). Utilizing the ZOO method, the size of the backward message is reduced to just q. Thus, CE-VFAL substantially reduces the size of the forward-transmitted messages by employing a uniform scale compressor [Bennett, 1948] $C(\cdot)_8$. In each communication round, the clients transmit compressed embeddings, reducing the byte size from 32 bits to 8 bits, to the server instead of the raw messages.

Definition 2. ***Compression Error (forward message).*** *Considering sample i, we can define the compression error of $\mathcal{C}_f$: $\epsilon_{i,c}$, $c = 1, 2, ..., C$, i.e. $\epsilon_{i,c} = \mathcal{C}_\theta(i, x_{t_c}) - x_{i,t_c}$. We denote the expected norm of the error from the client c at global iteration k as $\mathcal{E}_{i,c}^k = \mathbb{E}||\epsilon_{i,c}^k||^2$, and $\mathcal{E}^k = \max_c \mathcal{E}_{i,c}^k$. Since all client operations are synchronized, the error from all clients is $\epsilon_i^k = (\epsilon_{i,1}^k, \epsilon_{i,2}^k, ..., \epsilon_{i,C}^k)$. Then, the expected norm of the error from all clients:*

$$\mathbb{E}||\epsilon_i^k||^2 = \mathbb{E}||(\epsilon_{i,1}^k, \epsilon_{i,2}^k, ..., \epsilon_{i,C}^k)||^2 \leq \sum_{c=1}^{C} \mathbb{E}||\epsilon_{i,c}^k||^2 \leq C\mathcal{E}^k.$$

Compression gradient. In the M-loop, a compression gradient occurs when the output embeddings are compressed for transmission, resulting in compression error (Assumption 2

and 3, Definition 1). By ensuring the compression preserves the essential gradient information, we show that these errors are bounded and do not impede convergence in Lemma 1.

Lemma 2. ***Bound Compression Error.*** *Under Assumption 3, 4, and Definition 2, the norm of the difference between the loss function value with and without compression error is bounded:*

$$\mathbb{E}||\nabla_{\psi}\hat{\mathcal{A}}_i(\eta,\psi,\theta)-\nabla_{\psi}\mathcal{A}_i(\eta,\psi,\theta)||\leq H_{\psi}^2C\mathcal{E}^k,$$

$$\mathbb{E}||\nabla_{\theta}\hat{\mathcal{A}}_i(\eta,\psi,\theta)-\nabla_{\theta}\mathcal{A}_i(\eta,\psi,\theta)||\leq Q_{\theta}^2H_{\theta}^2C\mathcal{E}^k,$$

$$\mathbb{E}||\nabla_{x_{t_c}}\hat{\mathcal{A}}_i(\eta,\psi,\theta)-\nabla_{x_{t_c}}\mathcal{A}_i(\eta,\psi,\theta)||\leq H_{\theta}^2C\mathcal{E}^k.$$

3.3 Lazy Propagation

VFAL enhances robustness but incurs significant communication costs due to the numerous rounds required for adversarial sample generation. CE-VFAL employs the lazy propagation principle, confining most forward and backward propagation to the first network layer of client models during adversarial updates. We ingeniously leverage lazy propagation, originally designed to reduce computational costs, to minimize the communication rounds, keeping most adversary updates localized. Considering the VFL model as a dynamic system according to (3), the key mathematical insight lies in the computations can be constrained solely to the first layer while updating adversaries, implemented using a slack variable p (i.e. the delayed gradient at the first layer):

$$p_i^m=\nabla_{g_{\tilde{\Theta}}}\left(\ell(g_{\tilde{\Theta}}(f_0(x_i+\eta_i^{m,0})),y_i)\right)\cdot\nabla_{x_{i,1}}\left(g_{\tilde{\Theta}}(f_0(x_i+\eta_i^{m,0}))\right), \tag{5}$$

where $g_{\tilde{\Theta}}$ represents the server model and the client models excluding the first layer, and $f_0=\{f_{1,0},f_{2,0},\ldots,f_{C,0}\}$ represents the first layer of all C client models with output $x_{i,1}=\{x_{i,1,1},x_{i,2,1},\ldots,x_{i,C,1}\}$.

For the objective function (3), the **inner maximization loop** for adversary updating involves an M-loop, with each iteration of the M-loop containing an N-loop. In each M-loop, the slack variable p is calculated by querying the entire model. During the N-loop, the adversarial perturbation η is updated using p and a gradient ascent step, while keeping the network parameters Θ fixed. After M iterations, a single update to Θ is performed in the **outer minimization loop** using a gradient descent step.

Delayed gradient. This frozen slack variant introduces an oracle error in adversary updating, resulting in a delayed gradient. Given the strong concavity of η in the inner loop, we define the gradient of the loss function concerning η with Assumption 5 and bound it with Lemma 2.

Lemma 2. **Bound the gradient for** η. Under Assumption 1, Lemma 1, 3, at the i-th sample and k-th iteration, let x_{i,t_c} be the output of c-th client and $G'=K^{t_c+1}(K^{t_c}+t_c(t_c-1)K^{2t_c-2}+t_cK^{2t_c})$, $\alpha_x<\frac{1}{L_{\eta\eta}}$, we define $G=KG'$, $\alpha_x<\frac{1}{L_{\eta\eta}}$, the pseudo-partial derivative for η satisfies the following inequality $\hat{\eta}_i=\underset{\substack{m=1,\ldots,M\\n=1,\ldots,N}}{\operatorname{argmin}}\left\|\hat{\nabla}_{\eta}\hat{\mathcal{A}}_i\left(\eta_i^{m,n},\psi_i,\theta_i\right)\right\|$ then:

$$\begin{aligned}\mathbb{E}\left\|\nabla_{\eta}\mathcal{A}_i(\hat{\eta}_i,\psi_i,\theta_i)\right\|^2\leq&\left[D(\mathcal{X})L_{\eta\eta}^2\left(1-\frac{z}{L_{\eta\eta}}\right)^{MN+1}\right.\\&\left.+\frac{2G^2}{L_{\eta\eta}}(N-1)^2\left(\frac{2}{z}+\frac{1}{2L_{\eta\eta}}\right)\right]\\&\times3\left(H_{\theta}^2C\mathcal{E}^k+\frac{L^2\mu^2d^2}{4}+K^2\right).\end{aligned} \tag{6}$$

When calculating the gradient of η, we also introduce ZOO error and compression error. Therefore, we divide the gradient of η into two parts. The first part is the delayed gradient calculated within the client. The second part is the gradient which includes ZOO error and compression error, which is generated during the transfer from the server to the client. Finally, we obtain the gradient of η using the chain rule.

3.4 Algorithm

The training scheme of CE-VFAL is outlined in Algorithm 1 and detailed further in Appendix 3. Each client computes the embeddings of its local features, denoted as $x_{i,c,0}^{m,n}$, where i denotes the selected sample. These embeddings are transmitted to the server after compressing with $C(\cdot)_8$. The server receives embeddings from all clients and introduces random disturbances to calculate perturbed embeddings and the loss difference $\{\delta_{i,c}^j\}_{j=1}^q$. Clients estimate gradients with δ and then calculate the slack variable $p_{i,c}^m$. With $p_{i,c}^m$, client parties updates the adversarial sample N times (Line 10 in Algorithm 1) using gradient ascent. After M times such updating, the server party and client parties update the parameters, denoted as ψ and θ respectively, using gradient descent. This process iterates until convergence criteria are met.

3.5 Communication Efficiency of CE-VFAL

In analyzing the communication efficiency, we focus on communication rounds and load which are shown in Table 1.

- **Backward Communication Load:** CE-VFAL leverages the ZOO to drastically reduce the communication load during the backward pass. Traditional FO-based VFAL methods require transmitting $|x_{t_c}|*B$ elements from the server to the client for each backward propagation. In contrast, CE-VFAL reduces this load to q elements, where q represents the sampling times for multiple point estimations of ZOO method and is typically much smaller than $|x_{t_c}|*B$.
- **Forward Communication Load:** During the forward pass, CE-VFAL employs a compression technique denoted as $C(|x_{t_c}|*B)_8$, which reduces the byte size of forward-transmitted messages from 32 bits to 8 bits.
- **Number of Communication Iterations:** By integrating N communication-free iterations within each M communication iteration, CE-VFAL significantly reduces the total number of communication iterations required.

Algorithm 1: CE-VFAL-M-N

Input: Learning rates a_x, a_ψ, a_θ; Perturbation μ; Random vector u from distribution $\mathcal{P}$;
Output: Model parameters $\Theta = \{\theta_1, \theta_2, \ldots, \theta_C, \psi\}$;
Initialization: Initialize parameters $\theta_1, \theta_2, \ldots, \theta_C, \psi$;
while *not convergent* **do**
  Randomly select a sample x_i;
  for $m = 1$ *to* M **do**
    for *each client* c **do**
      Compute embeddings x_{i,t_c}, and send $C(x_{i,t_c})_8$ to server;
      while *receiving loss differences* $\{\delta_{i,c}^j\}_{j=1}^q$ **do**
        Compute ZO gradient $\hat{\nabla}_{x_{i,t_c}} l_i^m$ (Eq. 4);
        Compute slack variable p_i^m (Eq. 5);
        for $n = 1$ *to* N **do**
          Update adversarial sample $x_{i,c,0}^{m,n}$;
    Server receives x_{i,t_c} and compute perturbations;
    Server computes and sends loss differences $\{\delta_i^j\}_{j=1}^q$;
  for *each client* c **do**
    Update local model parameters θ_c;
  Update server model parameters ψ;

Methods	Adv. Upd. Iter.		Comm. Load per Iter.	
	Comm.	Comm-Free.	Forward	Backward
Clean_VFL	–	–	$(\lvert x_{t_c}\rvert * B)_{32}$	$\lvert x_{t_c}\rvert * B$
VFL_PGD-R	R	–	$(\lvert x_{t_c}\rvert * B)_{32}$	$\lvert x_{t_c}\rvert * B$
VFL_FreeAT-ϵ	ϵ	–	$(\lvert x_{t_c}\rvert * B)_{32}$	$\lvert x_{t_c}\rvert * B$
VFL_FreeLB-m	m	–	$(\lvert x_{t_c}\rvert * B)_{32}$	$\lvert x_{t_c}\rvert * B$
CE-VFAL-M-N	M	N	$C(\lvert x_{t_c}\rvert * B)_8$	q

Table 1: Comparison on Communication rounds and load in **one batch** with size B (red means the iterations with communication, and blue means communication-free iterations)

4 Convergence Analysis

The convergence analysis of CE-VFAL presents unique theoretical challenges arising from the intricate interactions among multiple sources of gradient approximations: (1) *delayed gradients* induced by lazy propagation mechanism, (2) *compression gradients* stemming from embedding quantization, and (3) *estimated gradients* resulting from zeroth-order optimization. These approximations introduces two analytical challenges: precisely characterizing the complex interplay among these various gradient approximations, and developing rigorous bounds that account for their cumulative effects throughout the training process.

Theorem 1. Under Assumptions 1∼4, let $L_\star = \max\{L, L_\psi, L_\theta, L_{\psi\eta}, L_{\theta\eta}\}$, $\alpha_m = \min\{\alpha_\psi, \alpha_\theta\}$, $\alpha_M = \max\{\alpha_\psi, \alpha_\theta\}$, $\xi_\theta = 1 + L_\theta \alpha_M$ and $\xi_\psi = 1 + L_\psi \alpha_M$. Also, we denote $\pi(M,N) = \frac{2G^2}{L_{\eta\eta}}(N-1)^2\left(\frac{2}{z} + \frac{1}{2L_{\eta\eta}}\right) + D(\mathcal{X})L_{\eta\eta}^2\left(1 - \frac{z}{L_{\eta\eta}}\right)^{MN+1}$, $\eta_i^* = \text{argmax}_\eta \mathcal{A}_i(\eta, \psi, \theta)$ and $\Lambda = \mathcal{R}\left(\eta^{*,0}, \psi^0, \theta^0\right) - \inf_k(\mathcal{R}\left(\eta^{*,k}, \psi^k, \theta^k\right))$. If the conditions satisfied: $\alpha_x < 1/L_{\eta\eta}$ and $\frac{\alpha_M}{\alpha_m} < \infty$, the global true gradient of CE-VFAL-M-N satisfies the following:

$$\frac{1}{\mathcal{K}}\sum_{k=0}^{\mathcal{K}-1}\mathbb{E}\left[||\nabla\mathcal{R}\left(\eta^{*,k}, \psi^k, \theta^k\right)||^2\right] \leq \mathcal{T}_1 + \mathcal{T}_2 + E_p + E_c + E_z. \tag{7}$$

First-order optimization convergence term:

$$\mathcal{I}_1 = \frac{2\Lambda}{\alpha_m \mathcal{K}}. \tag{8}$$

Stochastic gradient descent term:

$$\mathcal{I}_2 = \frac{2L_\star \alpha_M^2(\sigma_\psi^2 + \sigma_\theta^2)}{\alpha_m}. \tag{9}$$

Zeroth-order error term:

$$E_z = \mu^2\left(\frac{3\alpha_M \xi_\theta L_\star^2 d^2}{4\alpha_m} + \frac{3\pi(M,N)L_\star^2 d^2 a_M \xi_\psi}{2a_m z^2} + \frac{9\pi(M,N)L_\star^2 d^2 a_M \xi_\theta}{4a_m z^2}\right). \tag{10}$$

Compression error term:

$$E_c = \mathcal{E}\frac{\alpha_M}{\alpha_m}\left(2\xi_\psi H_\psi^2 C + 3\xi_\theta Q_\theta^2 H_\theta^2 C + \frac{3H_\theta^2 C\pi(M,N)}{z^2}(2\xi_\psi L_\star^2 + 3\xi_\theta L_\star^2)\right). \tag{11}$$

Lazy propagation error term:

$$E_p = \frac{3\alpha_M K^2 \pi(M,N)}{\alpha_m z^2}\left(2\xi_\psi L_\star^2 + 3\xi_\theta L_\star^2\right). \tag{12}$$

The term $\mathcal{T}_1$ is typical for convergence of first-order optimization algorithms on smooth non-convex functions; The term $\mathcal{T}_2$ is typical for stochastic gradient descent at fixed step size; Terms E_c are the errors during forward communication due to compression; Term E_z is the errors due to ZOO; Term E_p is errors due to lazy propagation.

Corollary 1. *If we choose* α_θ *and* α_ψ *as* $\mathcal{O}(\frac{1}{\sqrt{\mathcal{K}}})$, $\mu = \mathcal{O}(\frac{1}{\mathcal{K}^{\frac{1}{4}}})$, $\mathcal{E} = \mathcal{O}(\frac{1}{\sqrt{\mathcal{K}}})$, $\Gamma = \mathcal{O}(\frac{1}{\sqrt{\mathcal{K}}})$, *we can derive the sublinear convergence rate:*

$$\frac{1}{\mathcal{K}}\sum_{k=0}^{\mathcal{K}-1}\mathbb{E}\left[||\nabla\mathcal{R}\left(\eta^{*,k}, \psi^k, \theta^k\right)||^2\right] \leq \mathcal{O}(\frac{1}{\sqrt{\mathcal{K}}}) + \mathcal{O}(\frac{N}{M}). \tag{13}$$

By constraining multi-source approximate gradients, we demonstrate the sublinear convergence rate $\mathcal{O}(\frac{1}{\sqrt{\mathcal{K}}})$.

Remark 1. *The term* $\mathcal{O}(\frac{N}{M})$ *reveals the optimal parameter selection strategy:* M *should be set as large as the communication budget allows since* $\pi(M,N)$ *decreases with respect to* M. *Due to the convexity of* $\pi(M,N)$ *in* N, *the optimal* N *exists at the point where the partial derivative with respect to* N *becomes positive—beyond this point, excessive* N *values deteriorate robust accuracy. This dependence is empirically validated through ablation experiments (Figure 3).*

Proof Sketch. We begin by transforming the original min-max optimization problem into a Hamiltonian system (Appendix A.4). The convergence analysis leverages three types of approximate gradients: delayed gradient (Lemma 2), compression gradient (Lemma 3), and estimated gradient (Lemma 1). We establish the global convergence by proving that the loss function is L-smooth (Assumption 1). By combining the results from M, N, and outer loop analyses, we demonstrate that the model parameters converge asymptotically (Theorem 1). We provided detailed proof in Appendix 2.

5 Experiments

We conducted a systematic suite of experiments to assess the efficacy of our proposed framework. Due to space limitations, experimental details are detailed in Appendix 4. The source code for this project is available at the following URL: https://workelaina.github.io/CE-VFAL/.

5.1 Experiment Setups

Datasets. We evaluate on public datasets including image datasets MNIST [LeCun *et al.*, 1998], FER [Barsoum *et al.*, 2016], CIFAR10 and CIFAR100 [Krizhevsky, 2009], and tabular dataset Criteo [Tien *et al.*, 2014]. To simulate the VFL scenario, we allocated distinct features to each party based on the described methodology in prior works [Luo *et al.*, 2021; Qiu *et al.*, 2022; Fu *et al.*, 2022].

Training procedures. The number of parties is set to three, comprising one server and two clients. The server's model consists of two linear layers, while the client models use ResNet18 for MNIST and FER, ResNet34 for CIFAR10, and MLP for Criteo. Throughout the communication process between the server and clients, the size of the message (i.e., the embedding) for each sample is set to 128.

Baselines. We implemented various AT algorithms within a standard VFL context as baselines, including **VFL_PGD-**R [Madry *et al.*, 2017], **VFL_FreeAT-**ϵ [Shafahi *et al.*, 2019], **VFL_FreeLB-**m [Zhu *et al.*, 2019], and **Clean_VFL** [Vepakomma *et al.*, 2018], which represents a standard VFL model without AT. To ensure a fair comparison with **CE-VFAL-**M**-**N, we constrained the number of communications conducted in a single adversarial sample generation not to exceed PGD's setting, denoted as $\max\{M, R, \epsilon, m\} = R$.

Adversarial attack. To substantiate robustness, we employed a suite of adversarial attack methods: FGSM is a fast and non-iterative attack [Kurakin *et al.*, 2016]; PGD R operates by iteratively perturbing input data using gradients to maximize the model's loss [Madry *et al.*, 2017]; CW incorporates a custom loss function to ensure minimal perturbations while maintaining misclassification [Carlini and Wagner, 2017]; AutoAttack (AA) combines various attack strategies, enhancing the likelihood of evading detection and bypassing defenses [Croce and Hein, 2020].

Methods	Robust Accuracy (%)					Comm. Cost (MB)
	Clean	FGSM	PGD_40	CW_500	AA	
Clean_VFL	**99.12**	81.64	38.02	40.16	33.68	117.24
VFL_PGD-40	95.31	87.58	**80.01**	92.02	89.80	4804.72
VFL_FreeAT-8	95.04	85.79	74.54	88.34	88.76	937.53
VFL_FreeLB-40	97.68	82.78	53.63	88.16	85.57	4687.59
CE-VFAL-5-10	**97.87**	**90.01**	79.68	**91.23**	**93.02**	**73.62**

Table 2: Results of MNIST Robust Training.

5.2 Main Results on Various Datasets

Our proposed CE-VFAL consistently delivers strong robustness with substantially lower communication cost across datasets. On MNIST (Table 2), it achieves leading clean/robust accuracy with only $\sim$1/65 of VFL_PGD's communication. On FER (Table 3), CE-VFAL maintains robust performance under multiple attacks while incurring minimal overhead. On CIFAR10 (Table 4), it offers the best overall trade-off, reducing communication to 5.7% of VFL_PGD. On Criteo (Table 5), CE-VFAL-5-10 matches clean accuracy while significantly improving robustness, whereas VFL_PGD-40 requires $\sim$65$\times$ higher communication for comparable robustness.

Methods	Robust Accuracy (%)					Comm. Cost (MB)
	Clean	FGSM	PGD_40	CW_100	AA	
Clean_VFL	**74.66**	48.08	10.31	28.12	36.20	56.14
VFL_PGD-40	65.77	62.83	**64.42**	65.43	63.38	2299.40
VFL_FreeAT-8	68.96	64.93	55.32	60.56	67.11	448.61
VFL_FreeLB-40	69.27	47.09	12.28	26.91	42.45	2242.94
CE-VFAL-5-10	**73.65**	**68.75**	57.84	**66.23**	**70.49**	**35.55**

Table 3: Results of FER Robust Training.

Methods	Robust Accuracy (%)					Comm. Cost (MB)
	Clean	FGSM	PGD_20	CW_100	AA	
Clean_VFL	**84.03**	53.32	37.13	10.95	76.45	97.74
VFL_PGD-10	83.42	59.08	39.52	57.25	77.98	1074.22
VFL_FreeAT-8	79.58	63.63	**44.71**	57.13	78.29	781.34
VFL_FreeLB-10	78.92	52.09	33.65	53.67	73.09	976.66
CE-VFAL-5-3	**83.56**	**63.69**	40.27	**64.93**	**78.51**	**61.46**

Table 4: Results of CIFAR10 Robust Training.

Methods	Robust Accuracy (%)				Comm. (MB)
	Clean	FGSM	PGD-40	AA	
Clean_VFL	**76.24**	3.09	39.13	33.72	277.71
VFL_PGD-40	76.03	23.54	73.91	71.32	11386.11
VFL_FreeAT-8	73.74	1.33	20.23	16.21	2221.68
VFL_FreeLB-40	76.23	3.20	39.73	34.27	11108.40
CE-VFAL-5-10	74.72	**70.69**	**74.29**	**73.36**	**174.14**

Table 5: Results of Criteo Robust Training.

Evaluation on Training Time. Figure 2 demonstrates that CE-VFAL-5-3 achieves 80% accuracy within 5,000 seconds, converging 6-8 times faster than baselines, while maintaining a superior balance between efficiency and robustness.

5.3 Ablation Study

The dependence on M and N. We conducted extensive experiments on the MNIST dataset to explore the dependence on parameters M and N. Figure 3 illustrate the change in accuracy with a fixed $M = 3, 5, 8, 10$, respectively, while varying N. It is evident that performance rapidly degrades with increasing N beyond a certain threshold, as predicted by Corollary 1. This observation highlights the sensitivity of the model's performance to N, emphasizing the importance of optimizing N to maintain high accuracy.

Effect of ZOO Parameters. Table 6 investigates how sampling times (q) in ZOO affect model robustness and communication overhead. While ZOO consistently achieves lower communication costs than first-order methods, increasing q

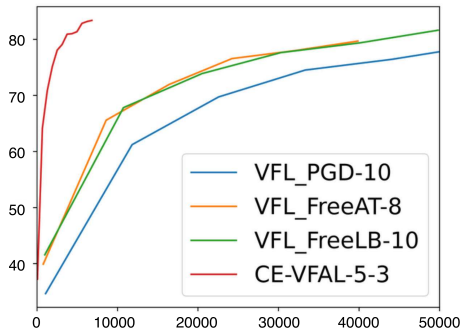

Figure 2: CIFAR10 Acc-Time.

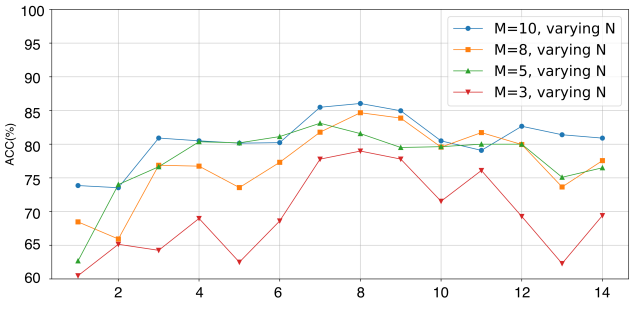

Figure 3: Results with fixed M and varying N.

substantially improves robustness across different attacks. Although ZOO with maximum sampling falls slightly short of first-order performance on some attacks, it achieves comparable clean accuracy and even superior performance on certain attacks with lower communication overhead.

q	Clean	PGD-40	CW-100	Comm. Cost (MB)
20	98.22	73.32	96.78	**73.63**
100	99.05	78.43	98.06	75.07
500	99.13	79.95	**98.72**	82.22
FO	**99.28**	**91.57**	98.40	366.25

Table 6: Ablation study on ZOO Parameters.

Effect of Compression Parameters. We investigate the impact of compression by evaluating different compression rates on forward messages. Table 7 reveals an interesting trade-off between compression rate, model robustness, and communication efficiency. Moderate compression achieves comparable clean accuracy to the uncompressed baseline while demonstrating superior robustness and significantly reducing communication costs. However, extreme compression severely degrades model performance across all metrics.

Experiments on different clients. To evaluate CE-VFAL's scalability across different federation scales, we conducted experiments on the Criteo dataset with varying client numbers. Table 8 demonstrates that CE-VFAL maintains superior performance and low communication overhead across different client configurations, exhibiting consistent robustness against various attacks while scaling effectively.

Para.	Clean	PGD-40	CW-100	Comm. Cost (MB)
1	11.43	11.35	8.63	**9.54**
2	97.43	**86.80**	95.68	18.70
4	97.39	81.79	96.11	37.01
8	97.45	76.48	96.26	73.63
16	97.45	78.32	96.09	146.87
32	**97.78**	78.63	**96.54**	293.32

Table 7: Ablation study on compression parameters.

Training Methods	Robust Accuracy (%)				Com. Cost (MB)
	Clean	FGSM	PGD-40	AA	
with 5 clients					
Clean_VFL	**76.23**	4.10	51.29	47.77	694.27
VFL_PGD-40	76.08	20.45	74.13	71.25	28465.27
VFL_FreeAT-8	74.60	2.72	19.86	17.47	5554.20
VFL_FreeLB-40	76.23	4.19	51.82	48.34	27771.00
CE-VFAL-5-10	74.81	**70.00**	**74.21**	**73.13**	**434.50**
with 8 clients					
Clean_VFL	**76.16**	12.84	61.96	62.32	1110.84
VFL_PGD-40	75.95	31.89	74.09	71.57	45544.43
VFL_FreeAT-8	75.12	6.06	33.96	32.45	8886.72
VFL_FreeLB-40	76.14	8.57	56.67	56.19	44433.59
CE-VFAL-5-10	74.82	**71.00**	**74.29**	**73.20**	**694.85**

Table 8: Comparison on Criteo with 5 and 8 clients.

Experiments on different perturbation budgets (ϵ). To evaluate robustness against varying attack intensities, we conducted ablation experiments with different perturbation budgets (ϵ). Table 9 shows that as perturbation budgets increase, all models experience declining defensive capabilities. CE-VFAL shows remarkable stability across all perturbation levels, maintaining consistently robustness even under strong attacks while others show substantial degradation.

Methods	$\epsilon = 0.2$			$\epsilon = 0.5$			$\epsilon = 0.8$		
	FGSM	PGD-40	AA	FGSM	PGD-40	AA	FGSM	PGD-40	AA
Clean_VFL	60.13	72.56	70.74	24.03	63.94	65.16	5.38	47.35	43.51
VFL_PGD-40	67.95	75.13	71.32	48.19	73.98	71.32	28.47	73.94	71.32
VFL_FreeAT-8	50.97	68.03	64.75	14.88	52.45	53.00	2.42	28.60	27.56
VFL_FreeLB-40	60.45	72.60	70.79	24.58	64.16	65.44	5.52	47.93	43.97
CE-VFAL-5-10	**74.03**	**74.58**	**73.34**	**72.91**	**74.41**	**73.34**	**71.31**	**74.32**	**73.34**

Table 9: Comparison with different perturbation budgets ϵ.

6 Conclusions

This paper presents CE-VFAL, a novel framework that significantly enhances the robustness of VFL models while maintaining high communication efficiency. By integrating the lazy propagation principle, ZOO, and compression techniques, CE-VFAL achieves a substantial reduction in the frequency and load of client-server interactions. Our theoretical analysis provides the first convergence proof for a VFAL framework, highlighting the sublinear convergence rate and robustness against adversarial attacks. Extensive experiments demonstrate CE-VFAL's superior performance in balancing communication costs with SOTA robustness.

Acknowledgements

Dr. Yi Chang was supported by the National Key R&D Program of China under Grant No. 2023YFF0905400, and the National Natural Science Foundation of China through grants No. U2341229 and No. 62076138. Dr. Zhaogeng Liu was supported by the Young Scientists Fund (C Class) of the National Natural Science Foundation of China under Grant No. 62506142.

Contribution Statement

Tianxing Man and Jinjie Fang (denoted by $*$ on the first page) contributed equally to the conceptualization, methodology, investigation, and writing of this work. Ganyu Wang supported the methodology and review. Yu Bai and Zhaogeng Liu contributed to the evaluation and visualization. Bin Gu and Yi Chang (the corresponding authors, denoted by † on the first page) contributed to the methodology, as well as the review and editing of the manuscript.

References

[Barsoum *et al.*, 2016] Emad Barsoum, Chi Zhang, Carlos Canton Ferrer, and Zhang Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. In *ACM International Conference on Multimodal Interaction*. ACM, 2016.

[Bennett, 1948] William Ralph Bennett. Spectra of quantized signals. *The Bell System Technical Journal*, 27(3):446–472, 1948.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[Castiglia *et al.*, 2022] Timothy J Castiglia, Anirban Das, Shiqiang Wang, and Stacy Patterson. Compressed-vfl: Communication-efficient learning with vertically partitioned data. In *International Conference on Machine Learning*, pages 2738–2766. PMLR, 2022.

[Croce and Hein, 2020] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[Duanyi *et al.*, 2023] YAO Duanyi, Songze Li, XUE Ye, and Jin Liu. Constructing adversarial examples for vertical federated learning: Optimal client corruption through multi-armed bandit. In *The Twelfth International Conference on Learning Representations*, 2023.

[Fu *et al.*, 2022] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX security symposium (USENIX Security 22)*, pages 1397–1414, 2022.

[Gao *et al.*, 2018] Xiang Gao, Bo Jiang, and Shuzhong Zhang. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76:327–363, 2018.

[Huang *et al.*, 2024] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324. IEEE, 1998.

[Li and Hao, 2018] Qianxiao Li and Shuji Hao. An optimal control approach to deep learning and applications to discrete-weight neural networks. In *International Conference on Machine Learning*, pages 2985–2994. PMLR, 2018.

[Li *et al.*, 2018] Qianxiao Li, Long Chen, Cheng Tai, and E Weinan. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(165):1–29, 2018.

[Li *et al.*, 2020] Ming Li, Yiwei Chen, Yiqin Wang, and Yu Pan. Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression. In *2020 16th international conference on control, automation, robotics and vision (ICARCV)*, pages 291–296. IEEE, 2020.

[Li *et al.*, 2023] Xiaoxiao Li, Zhao Song, and Jiaming Yang. Federated adversarial learning: A framework with convergence analysis. In *International Conference on Machine Learning*, pages 19932–19959. PMLR, 2023.

[Liu *et al.*, 2018] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[Liu *et al.*, 2024] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[Luo *et al.*, 2021] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *2021*

IEEE 37th International Conference on Data Engineering (ICDE), pages 181–192. IEEE, 2021.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2:2, 2016.

[Pang *et al.*, 2022] Qi Pang, Yuanyuan Yuan, Shuai Wang, and Wenting Zheng. Adi: Adversarial dominating inputs in vertical federated learning systems. *arXiv preprint arXiv:2201.02775*, 2022.

[Qiu *et al.*, 2022] Pengyu Qiu, Xuhong Zhang, Shouling Ji, Tianyu Du, Yuwen Pu, Jun Zhou, and Ting Wang. Your labels are selling you out: Relation leaks in vertical federated learning. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[Shafahi *et al.*, 2019] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

[Tien *et al.*, 2014] Jean-Baptiste Tien, joycenv, and Olivier Chapelle. Display advertising challenge. https://kaggle.com/competitions/criteo-display-ad-challenge, 2014. Kaggle.

[Tramèr *et al.*, 2018] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, 2018.

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

[Wang *et al.*, 2024] Ganyu Wang, Bin Gu, Qingsong Zhang, Xiang Li, Boyu Wang, and Charles X Ling. A unified solution for privacy and communication efficiency in vertical federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[Wei *et al.*, 2022] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. Vertical federated learning: Challenges, methodologies and experiments. *arXiv preprint arXiv:2202.04309*, 2022.

[Weinan, 2017] Ee Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11, 2017.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Zhang *et al.*, 2022] Jie Zhang, Song Guo, Zhihao Qu, Deze Zeng, Haozhao Wang, Qifeng Liu, and Albert Y Zomaya. Adaptive vertical federated learning on unbalanced features. *IEEE Transactions on Parallel and Distributed Systems*, 33(12):4006–4018, 2022.

[Zhu *et al.*, 2019] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. *arXiv preprint arXiv:1909.11764*, 2019.