FedPass: Privacy-Preserving Vertical Federated Deep Learning with Adaptive Obfuscation

Hanlin Gu[1] , **Jiahuan Luo**[1] , **Yan Kang**[1] , **Lixin Fan**[1] and **Qiang Yang**[1,2]
[1]Webank, China
[2]Hong Kong University of Science and Technology, Hong Kong

Abstract

Vertical federated learning (VFL) allows an active party with labeled feature to leverage auxiliary features from the passive parties to improve model performance. Concerns about the private feature and label leakage in both the training and inference phases of VFL have drawn wide research attention. In this paper, we propose a general privacy-preserving vertical federated deep learning framework called FedPass, which leverages adaptive obfuscation to protect the feature and label simultaneously. Strong privacy-preserving capabilities about private features and labels are theoretically proved (in Theorems 1 and 2). Extensive experimental result s with different datasets and network architectures also justify the superiority of FedPass against existing methods in light of its near-optimal trade-off between privacy and model performance.

1 Introduction

Vertical federated learning (VFL) [Yang *et al.*, 2019] allows multiple organizations to exploit in a privacy-preserving manner their private datasets, which may have some sample IDs in common but are significantly different from each other in terms of *features*. VFL found a great deal of successful applications, especially in collaborations between banks, healthcare institutes, e-commerce platforms, etc. [Yang *et al.*, 2019; Li *et al.*, 2020].

Despite of these successful VFL applications, privacy risks ascribed to certain corner cases were reported, e.g., in [Jin *et al.*, 2021; Fu *et al.*, 2022a; He *et al.*, 2019]. On one hand, *active parties* in VFL are concerned by the risk of leaking *labels* to passive parties. On the other hand, *passive parties* are keen to protect their *private features* from being reconstructed by the active party. To this end, a variety of privacy defense mechanisms include adding noise [Fu *et al.*, 2022a; Liu *et al.*, 2021], gradient discretization [Dryden *et al.*, 2016], gradient sparsification [Aji and Heafield, 2017], gradient compression [Lin *et al.*, 2018], and mixup [Huang *et al.*, 2020; Zhang *et al.*, 2018] has been proposed to boost the *privacy-preserving capability* of VFL. Nevertheless, as shown by detailed analysis and empirical study in this paper (see Sect. 4.2), all the aforementioned defense mechanisms suffer from deteriorated model performance to a certain extent. In this paper, we analyze the root cause of compromised model performance and propose an effective privacy-preserving method, which not only provides a strong privacy guarantee under various privacy attacks but also maintains unbending model performance for a variety of experimental settings.

Existing privacy defense mechanisms [Fu *et al.*, 2022a; Liu *et al.*, 2020; Dryden *et al.*, 2016; Aji and Heafield, 2017; Lin *et al.*, 2018; Huang *et al.*, 2020] can be essentially viewed as an **obfuscation mechanism** that provides privacy guarantees by adopting an obfuscation function $g(\cdot)$ acting on private features x or labels y, illustrated as follows:

$$x \xrightarrow{g(\cdot)} g(x) \xrightarrow{G_\theta} H \xrightarrow{F_\omega} \ell \longleftarrow g(y) \xleftarrow{g(\cdot)} y, \qquad (1)$$

in which H is the forward embedding that the passive party transfers to the active party; ℓ is loss; G and F represent the passive and active model parameterized by θ and ω, respectively. Note that in Eq. (1), the strength of privacy-preserving capability is prescribed by the *extent of obfuscation* (a.k.a. *distortion*) via a fixed hyper-parameter. As investigated in [Zhang *et al.*, 2022; Kang *et al.*, 2022c], significant obfuscations inevitably bring about the loss of information in $g(x)$ and $g(y)$ and thus lead to the loss of model performance (empirical study in Sect. 4.2). We argue that a fixed obfuscation strategy does not consider the dynamic learning process and constitutes the root cause of model performance degradation.

As a remedy to shortcomings of the fixed obfuscation, we propose to **adapt obfuscation function** g_θ and g_ω during the learning of model F_θ and G_ω, such that the obfuscation itself is also optimized during the learning stage. That is to say, the learning of the obfuscation function also aims to preserve model performance by tweaking model parameters θ and ω:

$$x \xrightarrow{g_\theta(\cdot)} g_\theta(x) \xrightarrow{G_\theta} H \xrightarrow{g_\omega(\cdot)} g_\omega(H) \xrightarrow{F_\omega} \ell \longleftarrow y, \qquad (2)$$

We regard this adaptive obfuscation as the gist of the proposed method. In this work, we implement the adaptive obfuscation based on the passport technique originally designed for protecting the intellectual property of deep neural networks (DNN) [Fan *et al.*, 2021; Li *et al.*, 2022a]. More specifically, we propose to embed *private passports* in both active and passive party models of VFL to effectively defend against privacy attacks. We therefore name the proposed method **FedPass**, which has three advantages: i) Private pass-

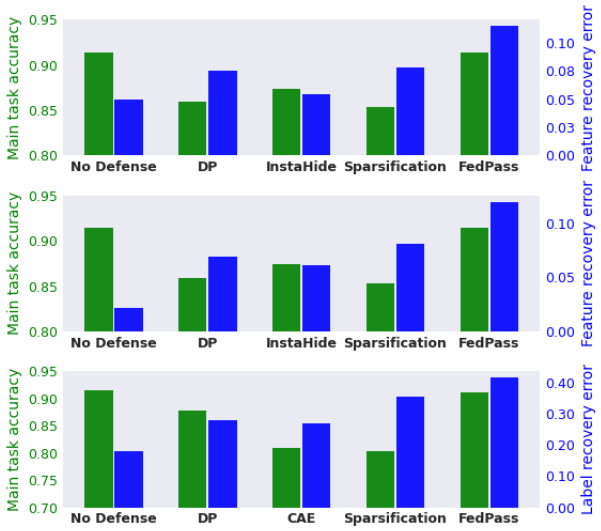

Figure 1: Comparison of FedPass with baseline defense methods against **Model Inversion** attack [He *et al.*, 2019] (the first line), **CAFE** attack [Jin *et al.*, 2021] (the second line), and **Model Completion** attack [Fu *et al.*, 2022a] (the third line) in terms of their main task accuracy (the higher the better, green) and data (feature or label) recovery error (the higher the better, blue) on ResNet-CIFAR10.

ports embedded in passive and active models prevent attackers from inferring features and labels. It is *exponentially hard* to infer features by launching various attacks, while attackers are defeated by a *non-zero recovery error* when attempting to infer private labels (see Sect. 3.4). ii) Passport-based obfuscation is learned in tandem with the optimization of model parameters, thus, preserving model performance (see Sect. 3.3). iii) The learnable obfuscation is efficient, with only minor computational costs incurred since no computationally extensive encryption operations are needed (see Sect. 4.2).

As Figure 1 illustrated, FedPass achieves almost lossless main task accuracy while obtaining the largest data recovery error in defending against three privacy attacks investigated in our experiments (see Sect. 4 for more results).

2 Related Work

We review related work from three aspects, namely, *vertical federated learning* (VFL) and *privacy attacks in VFL*, and *protection mechanism*

Vertical Federated Learning A variety of VFL methods has been proposed. [Hardy *et al.*, 2017] proposed vertical logistic regression (VLR) using homomorphic encryption (HE) to protect feature privacy. [Fu *et al.*, 2022b] proposed vertical neural network that employs a hybrid privacy-preserving strategy combining HE and secret sharing (SS) to protect feature. [Cheng *et al.*, 2021] proposed the SecureBoost, a VFL version of XGBoost, that leverages HE to protect the information exchanged among parties. To tackle the data deficiency issue of VFL, [Kang *et al.*, 2022b] combined semi-supervised learning and cross-view training to estimate missing features and labels for further training, [Liu *et al.*, 2020; Kang *et al.*, 2022a] integrated transfer learning into VFL to help the target party predict labels, while [He *et al.*, 2022] proposed a federated hybrid self-supervised learning framework to boost the VFL model performance through self-supervised learning based on unlabeled data.

Privacy Attacks in VFL There are two categories of privacy attacks in VFL: feature inference (FI) attacks and label inference (LI) attacks. FI attacks are typically launched by the active party to infer the features of a passive party. They can be tailored to shallow models such as logistic regression [Hu *et al.*, 2022] and decision trees [Luo *et al.*, 2021]. For deep neural networks, model inversion [He *et al.*, 2019] and CAFE [Jin *et al.*, 2021] are two representative FI attacks that infer private features through inverting passive parties' local models. LI attacks are mounted by a passive party to infer labels owned by the active party. The literature has proposed two kinds of LI attacks: the gradient-based [Li *et al.*, 2022b] and the model-based [Fu *et al.*, 2022a]. The former infers labels though analyzing the norm or direction of gradients back-propagated to the passive party, while the latter infers labels based on a pre-trained attacking model.

Defense mechanism We divide defense mechanisms applied to VFL into three categories: Protect features, labels and model parameters (or gradients). Specifically, most existing defense mechanisms typically apply to either model parameters or gradients to protect private data (features or labels). General defense mechanisms such as differential privacy [Abadi *et al.*, 2016] and sparsification [Fu *et al.*, 2022a; Lin *et al.*, 2018] can be used to defend against both the feature and label inference attacks by distorting (i.e., adding noise or compressing) model parameters or gradients. Specialized defense mechanisms such as MARVELL [Li *et al.*, 2022b] and Max-Norm [Li *et al.*, 2022b] are tailored to thwart label inference attacks by adding noise to gradients. InstaHide [Huang *et al.*, 2020] and Confusional AutoEncoder [Zou *et al.*, 2022] are two representative defense mechanisms that encode private data directly to protect data privacy. Cryptography-based defense mechanisms such as HE [Hardy *et al.*, 2017] and MPC [Gascón *et al.*, 2016] can protect both features and labels, but they impose a huge burden on computation and communication, especially for deep neural networks.

3 The Proposed Method

We introduce the VFL setting and threat models in Sect. 3.1 and Sect. 3.2, followed by elaboration on the proposed adaptive obfuscation framework, FedPass, in Sect. 3.3. We analyze the privacy preserving capability of FedPass in Sect. 3.4.

3.1 Vertical Federated Learning Setting

We assume that a vertical federated learning setting consists of one active party P_0 and K passive parties $\{P_1, \cdots, P_K\}$ who collaboratively train a VFL model $\Theta = (\theta, \omega)$ to optimize the following objective:

$$\min_{\omega,\theta_1,\cdots,\theta_K} \frac{1}{n}\sum_{i=1}^{n} \ell(F_\omega \circ (G_{\theta_1}(x_{1,i}), G_{\theta_2}(x_{2,i}), \cdots, G_{\theta_K}(x_{K,i})), y_i), \tag{3}$$

Threat model	Adversary	Attacking Target	Attacking Method	Adversary's Knowledge
Semi-honest	A passive party	*Labels* owned by the active party	Model Completion	A few labeled samples
	The active party	*Features* owned by a passive party	Model Inversion	Some labeled samples
		Features owned by a passive party	CAFE	Passive models

Table 1: Threat model we consider in this work.

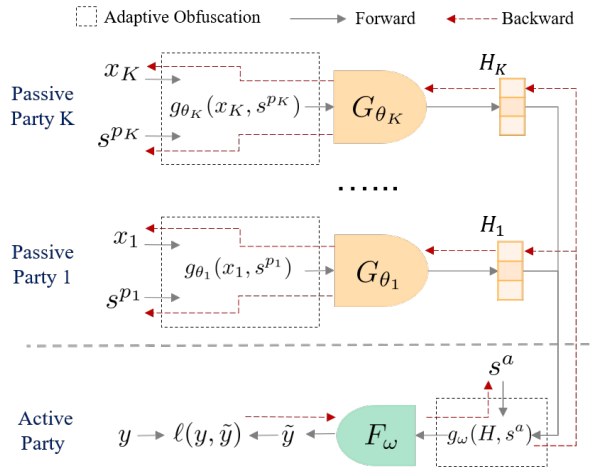

Figure 2: Overview of a VFL setting, in which multiple passive parties and one active party collaboratively train a VFL model, where passive parties only have the private features x, whereas the active party has private labels y. Both the active party and the passive party adopt the adaptive obfuscation by inserting passport into their models to protect features and labels.

in which Party P_k owns features $\mathcal{D}_k = (x_{k,1}, \cdots, x_{k,n}) \in \mathcal{X}_k$ and the passive model G_{θ_k}, the active party owns the labels $y \in \mathcal{Y}$ and active model F_ω, $\mathcal{X}_k$ and $\mathcal{Y}$ are the feature space of party P_k and the label space respectively. Each passive party k transfers its forward embedding H_k to the active party to compute the loss. The active model F_ω and passive models $G_{\theta_k}, k \in \{1, \ldots, K\}$ are trained based on backward gradients (See Figure 2 for illustration). Note that, before training, all parties leverage Private Set Intersection (PSI) protocols to align data records with the same IDs.

3.2 Threat Model

We assume all participating parties are *semi-honest* and do not collude with each other. An adversary (i.e., the attacker) $P_k, k = 0, \cdots, K$ faithfully executes the training protocol but may launch privacy attacks to infer the private data (features or labels) for other parties.

We consider two types of threat models: i) The active party wants to reconstruct the private features of a passive party through the model inversion attack [He *et al.*, 2019] or CAFE attack [Jin *et al.*, 2021]. ii) A passive party wants to infer the private labels of the active party through the model completion attack [Fu *et al.*, 2022a]. Table 1 summarizes threat models considered in this work.

3.3 FedPass

This section illustrates two critical steps of the proposed FedPass, i) embedding private passports to adapt obfuscation; ii) generating passports randomly to improve privacy preserving capability.

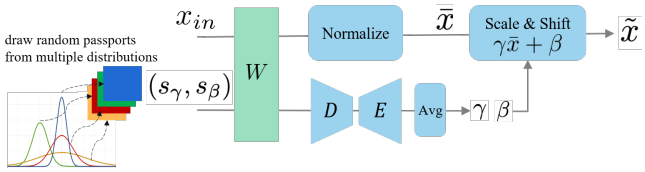

Figure 3: Adaptive obfuscation ($g(\cdot)$). We implemented $g(\cdot)$ by inserting a passport layer into a normal neural network layer.

Embedding Private Passports

In this work, we adopt the DNN passport technique proposed by [Fan *et al.*, 2019; Fan *et al.*, 2021] as an implementation for the adaptive obfuscation framework of FedPass. Specifically, the adaptive obfuscation is determined as follows:

$$\begin{aligned} g_W(x_{in}, s) =& \gamma(Wx_{in}) + \beta, \\ \gamma =& \text{Avg}\Big(D\big(E(Ws_\gamma)\big)\Big) \\ \beta =& \text{Avg}\Big(D\big(E(Ws_\beta)\big)\Big) \end{aligned} \tag{4}$$

where W denotes the model parameters of the neural network layer for inserting passports, x_{in} is the input fed to W, γ and β are the scale factor and the bias term. Note that the determination of the crucial parameters γ and β involves the model parameter W with private passports s_γ and s_β, followed by a autoencoder (Encoder E and Decoder D with parameters W') and a average pooling operation Avg$(\cdot)$. Learning adaptive obfuscation formulated in Eq.(4) brings about two desired properties:

- Passport-based parameters γ and β provide strong privacy guarantee (refer to Sect. 3.4): without knowing passports it is exponentially hard for attacker to infer layer input x_{in} from layer output $g_W(x_{in}, s)$, because attacker have no access to γ and β (see Theorem 1).
- Learning adaptive obfuscation formulated in Eq. (4) optimizes the model parameter W through three backpropagation paths via β, γ, W, respectively, which helps preserve model performance. This is essentially equivalent to adapting the obfuscation (parameterized by γ and β) to the model parameter W (more explanations in Appendix C); This adaptive obfuscation scheme offers superior model performance compared to fixed obfuscation schemes (see Sect. 4.2).

The training procedure of FedPass in Vertical Federated Learning is illustrated as follows (described in Algorithm 1):

1. Each passive party k applies the adaptive obfuscation to its private features with its private passports s^{p_k} and then sends the forward embedding H_k to the active party (line 3-9 of Algo. 1);
2. The active party sums over all $H_k, k \in \{1, \ldots, K\}$ as H, and applies the adaptive obfuscation to H with its private passports s^a, generating $\tilde{H}$. Then, the active party computes the loss $\tilde{\ell}$ and updates its model through back-propagation. Next, the active party computes gradients $\nabla_{H_k}\tilde{\ell}$ for each passive party k and sends $\nabla_{H_k}\tilde{\ell}$ to passive party k (line 10-19 of Algo. 1);
3. Each passive party k updates its model θ_k according to $\nabla_{H_k}\tilde{\ell}$ (line 20-22 of Algo. 1).

The three steps iterate until the performance of the joint model does not improve.

Random Passport Generation

How passports are generated is crucial in protecting data privacy. Specifically, when the passports are embedded in a convolution layer or linear layer with c channels[1], for each channel $j \in [c]$, the passport $s(j)$ (the j_{th} element of vector s) is randomly generated as follows:

$$s(j) \sim \mathcal{N}(\mu_j, \sigma^2), \quad \mu_j \in \mathcal{U}(-N, 0), \tag{5}$$

where all $\mu_j, j = 1, \cdots, c$ are different from each other, σ^2 is the variance of Gaussian distribution and N is the *passport range*, which are two crucial parameters of FedPass. The strong privacy-preserving capabilities rooted in such a random passport generation strategy are justified by theoretical analysis in Theorems 1 and 2 as well as experiment results in Sect. 4.2.

3.4 Privacy-Preserving Capability of FedPass

We investigate the privacy-preserving capability of FedPass against feature reconstruction attack and label inference attack. Note that we conduct the privacy analysis with linear regression models, for the sake of brevity. Proofs are deferred to Appendix D.

Definition 1. *Define the forward function of the passive model G and the active model F:*

- *For passive layer: $H = G(x) = W_p s^p_\gamma \cdot W_p x + W_p s^p_\beta$.*
- *For active layer: $y = F(H) = W_a s^a_\gamma \cdot W_a H + W_a s^a_\beta$.*

where W_p, W_a are 2D matrices of the passive and active models; $\cdot$ denotes the inner product, s^p_γ, s^p_β are passports of the passive party, s^a_γ, s^a_β are passports of the active party.

Hardness of Feature Restoration with FedPass

Consider the white-box Model Inversion (MI) attack (i.e., model inversion step in CAFE [Jin *et al.*, 2021; He *et al.*, 2019]) that aims to inverse the model W_p to recover features $\hat{x}$ approximating original features x. In this case, the attacker (i.e., the active party) knows the passive model parameters W_p, forward embedding H and the way of embedding passport, but does not know the passport.

Theorem 1. *Suppose the passive party protects features x by inserting the s^p_β. The probability of recovering features by the attacker via white-box MI attack is at most $\frac{\pi^{m/2}\epsilon^m}{\Gamma(1+m/2)N^m}$ such that the recovering error is less than ϵ, i.e., $\|x - \hat{x}\|_2 \leq \epsilon$,*

[1]For the convolution layer, the passport $s \in \mathbb{R}^{c \times h_1 \times h_2}$, where c is channel number, h_1 and h_2 are height and width; for the linear layer, $s \in \mathbb{R}^{c \times h_1}$, where c is channel number, h_1 is height.

Algorithm 1 FedPass

Input: Communication rounds T, Passive parties number K, learning rate η, batch size b, the passport range and variance $\{N^a, \sigma^a\}$ and $\{N^{p_k}, \sigma^{p_k}\}$ for the active party and passive party k respectively, the feature dataset $\mathcal{D}_k = (x_{k,1}, \cdots, x_{k,n_k})$ owned by passive party k, the aligned label $y = (y_1, \cdots, y_{n_0})$ owned by the active party.
Output: Model parameters $\theta_1, \cdots, \theta_K, \omega$

1: Initialize model weights $\theta_1, \cdots, \theta_K, \omega$.
2: **for** t in communication round T **do**
3: ▷ *Passive parties perform:*
4: **for** Passive Party k in $\{1, \ldots, K\}$ **do**:
5: Sample a batch $B_k = (x_{k,1}, \cdots, x_{k,b})$ from the dataset $\mathcal{D}_k$
6: Sample the passport tuple $s^{p_k} = (s^{p_k}_\gamma, s^{p_k}_\beta)$ according to Eq. (5) and N^{p_k}, σ^{p_k}
7: Compute $\tilde{B}_k = g_{\theta_k}(B_k, s^{p_k})$
8: Compute $H_k \leftarrow G_{\theta_k}(\tilde{B}_k)$
9: Send H_k to the active party
10: ▷ *The active party performs:*
11: Obtain the label y matched with $\{B_k\}_{k=1}^K$.
12: $H = \sum_{k=1}^K H_k$
13: Sample the passport tuple $s^a = (s^a_\gamma, s^a_\beta)$ via Eq. (5) and N^a, σ^a
14: Compute $\tilde{H} = g_\omega(H, s^a)$
15: Sample a batch of the label $Y = (y_1, \cdots, y_b)$
16: Compute cross-entropy loss: $\tilde{\ell} = \ell(F_\omega(\tilde{H}), Y)$
17: Update the active model as: $\omega = \omega - \eta\nabla_\omega\tilde{\ell}$
18: **for** k in $\{1, \ldots, K\}$ **do**:
19: Compute and send $\nabla_{H_k}\tilde{\ell}$ to each passive party k
20: ▷ *Passive parties perform:*
21: **for** Passive Party $k \in \{1, \ldots, K\}$ **do**:
22: Update θ_k by $\theta_k = \theta_k - \eta[\nabla_{H_k}\tilde{\ell}][\nabla_{\theta_k}H_k]$
return $\theta_1, \cdots, \theta_K, \omega$

Algorithm 2 Adaptive Obfuscation ($g(\cdot)$)

Input: Model parameters W of the neural network layer for inserting passports; the input x_{in} to which the adaptive obfuscation applies to; passport keys $s = (s_\gamma, s_\beta)$.
Output: The obfuscated version of the input.

Compute $\gamma = \text{Avg}\left(D\big(E(W * s_\gamma)\big)\right)$
Compute $\beta = \text{Avg}\left(D\big(E(W * s_\beta)\big)\right)$
return $\gamma(W * x_{in}) + \beta$

where m denotes the dimension of the passport via flattening, N denotes the passport range formulated in Eq. (5) and $\Gamma(\cdot)$ denotes the Gamma distribution. Theorem 1 demonstrates the attacker's probability of recovering features within error ϵ is exponentially small in the dimension of passport size m. The successful recovering probability is inversely proportional to the passport range N to the power of m.

Hardness of Label Recovery with FedPass

Consider the passive model competition attack [Fu *et al.*, 2022a] that aims to recover labels owned by the active party. The attacker (i.e., the passive party) leverages a small auxiliary labeled dataset $\{x_i, y_i\}_{i=1}^{n_a}$ belonging to the original training data to train the attack model W_{att}, and then infer labels for the test data. Note that the attacker knows the trained passive model G and forward embedding $H_i = G(x_i)$. Therefore, they optimizes the attack model W_{att} by minimizing $\ell = \sum_{i=1}^{n_a} \|W_{att}H_i - y_i\|_2$.

Assumption 1. *Suppose the original main algorithm of VFL is convergent. For the attack model, we assume the error of the optimized attack model W^*_{att} on test data $\tilde{\ell}_t$ is larger than that of the auxiliary labeled dataset $\tilde{\ell}_a$.*

Theorem 2. *Suppose the active party protect y by embedding s^a_γ, and adversaries aim to recover labels on the test data with the error $\tilde{\ell}_t$ satisfying:*

$$\tilde{\ell}_t \geq \min_{W_{att}} \sum_{i=1}^{n_a} \|(W_{att} - T_i)H_i\|_2, \tag{6}$$

where $T_i = diag(W_a s^a_{\gamma,i})W_a$ and $s^a_{\gamma,i}$ is the passport for the label y_i embedded in the active model. Moreover, if $H_{i_1} = H_{i_2} = H$ for any $1 \leq i_1, i_2 \leq n_a$, then

$$\tilde{\ell}_t \geq \frac{1}{(n_a - 1)} \sum_{1 \leq i_1 < i_2 \leq n_a} \|(T_{i_1} - T_{i_2})H\|_2 \tag{7}$$

Proposition 1. *Since passports are randomly generated and W_a and H are fixed, if the $W_a = I, H = \vec{1}$, then it follows that:*

$$\tilde{\ell}_t \geq \frac{1}{(n_a - 1)} \sum_{1 \leq i_1 < i_2 \leq n_a} \|s^a_{\gamma,i_1} - s^a_{\gamma,i_2}\|_2) \tag{8}$$

Theorem 2 and Proposition 1 show that the label recovery error $\tilde{\ell}_t$ has a lower bound, which deserves further explanations. First, when passports are randomly generated for all data, i.e., $s^a_{\gamma,i_1} \neq s^a_{\gamma,i_2}$, then a non-zero label recovery error is guaranteed no matter how adversaries attempt to minimize it. The recovery error thus acts as a protective random noise imposed on true labels. Second, the magnitude of the recovery error monotonically increases with the variance σ^2 of the Gaussian distribution passports sample from (in Eq. (5)), which is a crucial parameter to control privacy-preserving capability (see Experiment results in Appendix B) are in accordance with Theorem 2. Third, it is worth noting that the lower bound is based on the training error of the auxiliary data used by adversaries to launch PMC attacks. Given possible discrepancies between the auxiliary data and private labels, e.g., in terms of distributions and the number of dataset samples, the actual recovery error of private labels can be much larger than the lower bound. Again, this strong protection is observed in experiments (see Sect. 4.2).

4 Experiment

We present empirical studies of FedPass in defending against feature reconstruction attack and label inference attack.

4.1 Experiment Setting

Models & Datasets & VFL Setting

We conduct experiments on three datasets: *MNIST* [LeCun *et al.*, 2010], *CIFAR10* [Krizhevsky *et al.*, 2014] and ModelNet [Wu *et al.*, 2015]. We adopt LeNet [LeCun *et al.*, 1998] for conducting experiments on MNIST and adopt *AlexNet* [Krizhevsky *et al.*, 2012] and *ResNet18* [He *et al.*, 2016] on CIFAR10. For each dataset, passive party only provides private data while active party only provides labels.

We simulate a VFL scenario by splitting a neural network into a bottom model and a top model and assigning the bottom model to each passive party and the top model to the active party (see details of the experimental setting in Appendix A).

Privacy Attack Methods

We investigate the effectiveness of FedPass through three privacy attacks designed for VFL, namely, Passive Model Completion (PMC) attack [Fu *et al.*, 2022a], CAFE attack [Jin *et al.*, 2021] and Model Inversion (MI) attack [He *et al.*, 2019]. The first attack is a label inference attack, whereas the last two are feature reconstruction attacks (see details in Appendix A).

Baseline Defense Methods

We adopt four defense mechanisms as baselines to evaluate the effectiveness of FedPass in defending against feature reconstruction and label inference attacks. Each defense mechanism is controlled by a defense strength parameter to trade-off privacy leakage and model performance. For **Differential Privacy (DP)** [Abadi *et al.*, 2016], we experiment with Gaussian noise levels ranging from 5e-5 to 1.0. We add noise to gradients for defending against MC attack while add noise to forward embeddings for thwarting CAFE and MI attacks. For **Sparsification** [Fu *et al.*, 2022a; Lin *et al.*, 2018], we implement gradient sparsification [Fu *et al.*, 2022a] and forward embedding sparsification [Lin *et al.*, 2018] for defending against label inference attack and feature reconstruction attack, respectively. Sparsification level are chosen from 0.1% to 50.0%. For **Confusional AutoEncoder** (CAE) [Zou *et al.*, 2022], we follow the implementation of the original paper. That is, both the encoder and decoder of CAE have the architecture of 2 FC layers. Values of the hyperparameter that controls the confusion level are chosen from 0.0 to 2.0. For **InstaHide** [Huang *et al.*, 2020], we mix up 1 to 4 of images to trade-off privacy and utility. For **FedPass**, the range of the mean of Gaussian distribution N is from 2 to 200, the variance is from 1 to 64. Passports are embedded in the last convolution layer of the passive party's model and first fully connected layer of the active party's model.

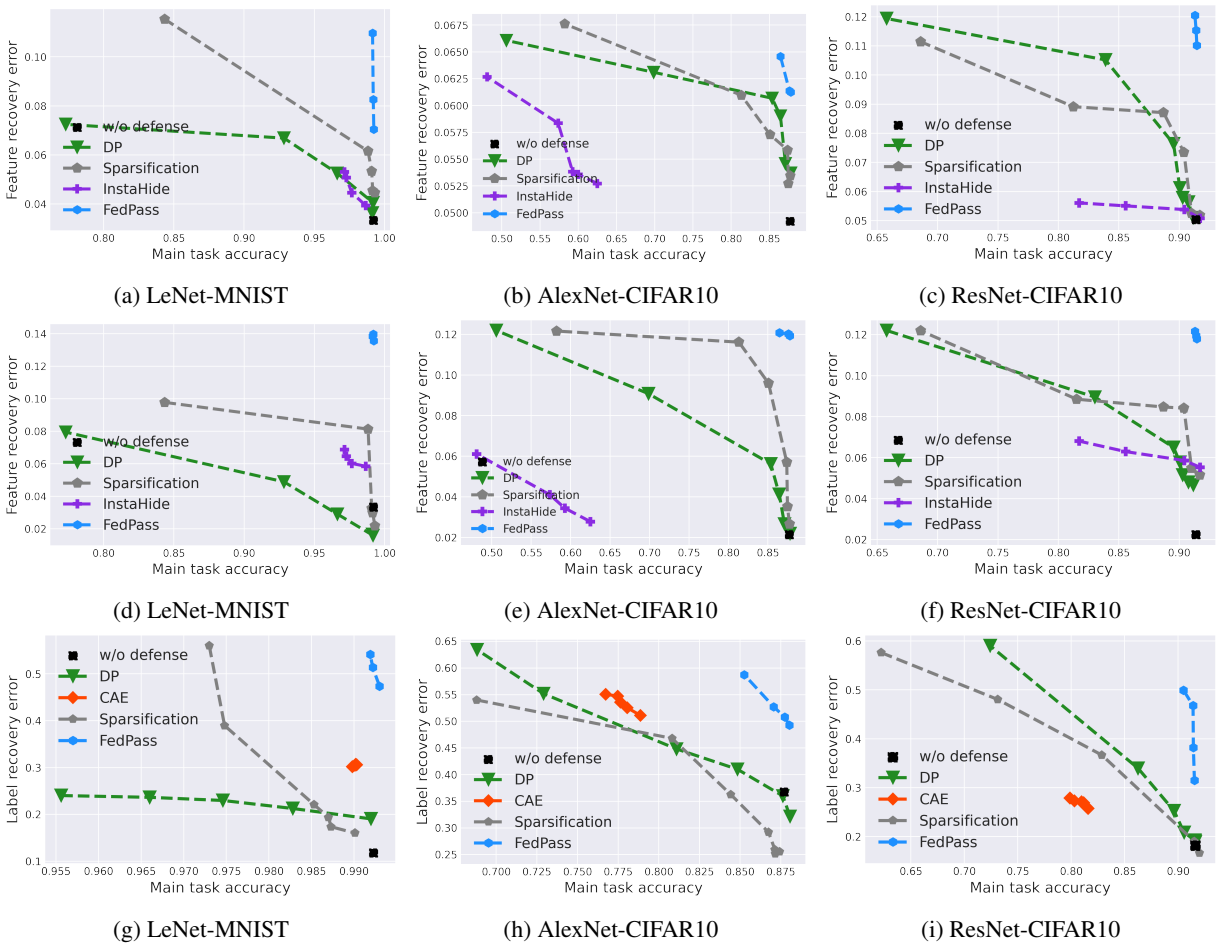

Figure 4: Comparison of different defense methods in terms of their trade-offs between main task accuracy and data (feature or label) recovery error against three attacks on LeNet-MNIST, AlexNet-CIFAR10 and ResNet-CIFAR10, respectively. **Model Inversion** (the first line) and **CAFE** (the second line) are feature reconstruction attacks, whereas **Passive Model Completion** (the third line) is a label inference attack. *A better trade-off curve should be more toward the top-right corner of each figure*.

Defense / Attack		w/o defense	CAE	Sparsification	DP	InstaHide	FedPass
CAFE	LeNet	0.033	—	0.049±0.026	0.033±0.018	0.061±0.004	**0.137±0.002**
	AlexNet	0.019	—	0.058±0.026	0.042±0.017	0.023±0.004	**0.105±0.001**
	ResNet	0.021	—	0.067±0.014	0.057±0.014	0.053±0.002	**0.109±0.001**
MI	LeNet	0.033	—	0.060±0.020	0.049±0.010	0.046±0.005	**0.087±0.001**
	AlexNet	0.043	—	0.047±0.003	0.046±0.006	0.032±0.001	**0.054±0.001**
	ResNet	0.046	—	0.065±0.012	0.063±0.015	0.047±0.001	**0.105±0.004**
PMC	LeNet	0.117	0.302±0.002	0.277±0.140	0.216±0.015	—	**0.506±0.028**
	AlexNet	0.322	0.415 ±0.008	0.283±0.065	0.358±0.051	—	**0.460±0.025**
	ResNet	0.166	0.217±0.004	0.268±0.088	0.237±0.087	—	**0.379±0.065**

Table 2: The Calibrated averaged performance (CAP) for different defense mechanisms against CAFE, MI and PMC attacks.

Evaluation Metrics

We use data (feature or label) recovery error and main task accuracy to evaluate defense mechanisms. We adopt the ratio of incorrectly labeled samples by a label inference attack to all labeled samples to measure the performance of that label inference attack. We adopt Mean Square Error (MSE) [Zhu *et al.*, 2019] between original images and images recovered by a feature reconstruction attack to measure the performance of that feature reconstruction attack. MSE is widely used to assess the quality of recovered images. A higher MSE value indicates a higher image recovery error. In addition, we leverage Calibrated Averaged Performance (CAP) [Fan *et al.*, 2020] to quantify the trade-off between main task accuracy and data recovery error. CAP is defined as follows:

Definition 2 (Calibrated Averaged Performance (CAP))**.** *For a given Privacy-Preserving Mechanism $g_s \in \mathcal{G}$ (s denotes the controlled parameter of g, e.g., the sparsification level, noise level and passport range) and attack mechanism $a \in \mathcal{A}$, the Calibrated Averaged Performance is defined as:*

$$CAP(g_s, a) = \frac{1}{m} \sum_{s=s_1}^{s_m} Acc(g_s, x) * Rerr(x, \hat{x}_s), \quad (9)$$

where $Acc(\cdot)$ denotes the main task accuracy and $Rerr(\cdot)$ denotes the recovery error between original data x and estimated data $\hat{x}_s$ via attack a.

4.2 Experiment Results

Defending against the Feature Reconstruction Attack

Figure 4 (a)-(f) compare the trade-offs between feature recovery error (y-axis) and main task accuracy (x-axis) of FedPass with those of baselines against MI and CAFE attacks on three models. We observe: i) DP and Sparsification can generally achieve either a high main task performance or a large feature recovery error (low privacy leakage), but not both. For example, DP and Sparsification can achieve a main task performance as high as ≥ 0.90 while obtaining a feature recovery error as low as ≤ 0.06 on ResNet-CIFAR10. At the other extreme, DP and Sparsification can achieve ≥ 0.11 feature recovery error but obtain ≤ 0.70 main task performance on ResNet-CIFAR10. ii) InstaHide generally can not thwart MI and CAFE attacks. Even mixing up with more data, InstaHide still leads to a relatively small feature recovery error while its main task performance degrades significantly. iii) The trade-off curves of FedPass reside near the top-right corner under both attacks on all models, indicating that FedPass achieves the best performance on preserving feature privacy while maintaining the model performance. For example, FedPass achieves ≥ 0.91 main task accuracy and ≥ 0.12 feature recovery error under MI and CAFE attacks on ResNet-CIFAR10. Table 2 also demonstrates that FedPass has the best trade-off between privacy and performance under MI and CAFE attacks.

Figure 5 showcases that, when protected by FedPass ($r8$), reconstructed images under the CAFE attack on all three datasets are essentially random noise, manifesting that FedPass can thwart the CAFE attack effectively. At the same time, the model performance of FedPass is almost lossless compared to that of the original model (w/o defense) on every dataset (see Figure 4). This superior trade-off of FedPass is in sharp contrast to existing methods, among which DP and Sparsification with high protection levels ($r5$ and $r7$) lead to significantly deteriorated model performance.

Defending against the Label Inference Attack

Figure 4 (g)-(i) compare the trade-offs between label recovery error (y-axis) and main task accuracy (x-axis) of FedPass with those of baselines in the face of the PMC attack on three models. It is observed DP, Sparsification, and CAE fail to achieve the goal of obtaining a low level of privacy leakage while maintaining model performance, whereas FedPass is more toward the top-right corner, indicating that FedPass has a better trade-off between privacy and performance. Table 2 reinforce the observation that FedPass achieves the best trade-off between privacy and performance under PMC attack.

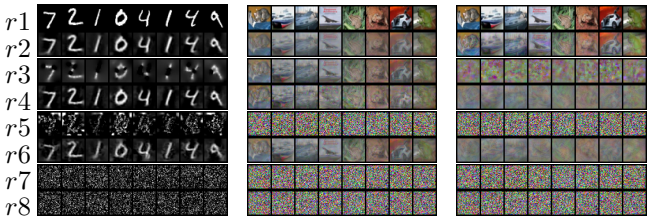

Figure 5: Original images and images reconstructed by CAFE attack for different defense mechanisms on LeNet-MNIST, AlexNet-CIFAR10 and ResNet-CIFAR10, respectively. From top to bottom, a row represents original image ($r1$), no defense ($r2$), InstaHide ($r3$), DP with noise level 0.2 ($r4$) and 2 ($r5$), Sparsification with sparsification level 0.5 ($r6$) and 0.05 ($r7$), and FedPass ($r8$).

Training and Inference Time

Table 3 investigates the training time (for one epoch) and inference time for FedPass and baseline defense methods. It shows that the FedPass is as efficient as the VFL w/o defense for both training and inference procedures (the training time on MNIST for each epoch is 7.03s and inference time is 1.48s) because embedding passport only introduces a few model parameters to train. It is worth noting that the training time of InstaHide is almost twice that of other methods because InstaHide involves mixing-up multiple feature vectors or labels, which is time-consuming.

Defense Method	LeNet-MNIST		AlexNet-Cifar10		ResNet-Cifar10	
	Train	Infer	Train	Infer	Train	Infer
w/o defense	7.03	1.48	22.37	2.20	22.64	2.18
CAE	7.30	1.48	22.71	2.27	23.02	2.21
Sparsification	6.93	1.45	22.39	2.12	22.61	2.21
DP	7.01	1.49	22.24	2.23	22.63	2.16
InstaHide	21.76	1.50	37.07	2.19	46.26	2.18
FedPass (ours)	7.05	1.46	22.58	2.13	22.61	2.16

Table 3: Comparison of training time (for one epoch) and inference time among different defense mechanisms.

5 Conclusion

This paper proposes a novel privacy-preserving vertical federated deep learning framework called FedPass, which leverages adaptive obfuscation to protect the label and data simultaneously. Specifically, the proposed adaptive obfuscation is implemented by embedding private passports in the passive and active models to adapt the deep learning model such that the model performance is preserved. The extensive experiments on multiple datasets and theoretical analysis demonstrate that the FedPass can achieve significant improvements over the other protected methods in terms of model performance and privacy-preserving ability.

Acknowledgements

This work is partly supported by National Key Research and Development Program of China (2020YFB1805501).

Contribution Statement

Hanlin Gu and Jiahuan Luo contributed equally to this work. Lixin Fan is the corresponding author.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[Aji and Heafield, 2017] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 440–445, 2017.

[Cheng *et al.*, 2021] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36(06):87–98, nov 2021.

[Dryden *et al.*, 2016] Nikoli Dryden, Tim Moon, Sam Ade Jacobs, and Brian Van Essen. Communication quantization for data-parallel training of deep neural networks. In *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*, pages 1–8. IEEE, 2016.

[Fan *et al.*, 2019] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in neural information processing systems*, 32, 2019.

[Fan *et al.*, 2020] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In *Federated Learning*, pages 32–50. Springer, 2020.

[Fan *et al.*, 2021] Lixin Fan, Kam Woh Ng, Chee Seng Chan, and Qiang Yang. Deepip: Deep neural network intellectual property protection with passports. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2021.

[Fu *et al.*, 2022a] 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), Boston, MA*, 2022.

[Fu *et al.*, 2022b] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. Blindfl: Vertical federated machine learning without peeking into your data. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 1316–1330, New York, NY, USA, 2022. Association for Computing Machinery.

[Gascón *et al.*, 2016] Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans. Secure linear regression on vertically partitioned datasets. *IACR Cryptol. ePrint Arch.*, 2016:892, 2016.

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

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.*, 2019] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019.

[He *et al.*, 2022] Yuanqin He, Yan Kang, Jiahuan Luo, Lixin Fan, and Qiang Yang. A hybrid self-supervised learning framework for vertical federated learning. *arXiv preprint arXiv:2208.08934*, 2022.

[Hu *et al.*, 2022] Yuzheng Hu, Tianle Cai, Jinyong Shan, Shange Tang, Chaochao Cai, Ethan Song, Bo Li, and Dawn Song. Is vertical logistic regression privacy-preserving? a comprehensive privacy analysis and beyond. *arXiv preprint arXiv:2207.09087*, 2022.

[Huang *et al.*, 2020] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International conference on machine learning*, pages 4507–4518. PMLR, 2020.

[Jin *et al.*, 2021] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. *NeurIPS*, 34:994–1006, 2021.

[Kang *et al.*, 2022a] Y. Kang, Y. He, J. Luo, T. Fan, Y. Liu, and Q. Yang. Privacy-preserving federated adversarial domain adaptation over feature groups for interpretability. *IEEE Transactions on Big Data*, (01):1–12, jul 2022.

[Kang *et al.*, 2022b] Yan Kang, Yang Liu, and Xinle Liang. FedCVT: Semi-supervised Vertical Federated Learning with Cross-view Training. *ACM Transactions on Intelligent Systems and Technology (TIST)*, May 2022.

[Kang *et al.*, 2022c] Yan Kang, Jiahuan Luo, Yuanqin He, Xiaojin Zhang, Lixin Fan, and Qiang Yang. A framework for evaluating privacy-utility trade-off in vertical federated learning. *arXiv preprint arXiv:2209.03885*, 2022.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[Krizhevsky *et al.*, 2014] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LeCun *et al.*, 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[Li *et al.*, 2020] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[Li *et al.*, 2022a] Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. Fedipr: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Li *et al.*, 2022b] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. In *International Conference on Learning Representations*, 2022.

[Lin *et al.*, 2018] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

[Liu *et al.*, 2020] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 35(4):70–82, 2020.

[Liu *et al.*, 2021] Yang Liu, Zhihao Yi, Yan Kang, Yuanqin He, Wenhan Liu, Tianyuan Zou, and Qiang Yang. Defending label inference and backdoor attacks in vertical federated learning. *arXiv preprint arXiv:2112.05409*, 2021.

[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)*. IEEE, apr 2021.

[Wu *et al.*, 2015] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[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.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[Zhang *et al.*, 2022] Xiaojin Zhang, Yan Kang, Kai Chen, Lixin Fan, and Qiang Yang. Trading off privacy, utility and efficiency in federated learning. *arXiv preprint arXiv:2209.00230*, 2022.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, , and Song Han. Deep leakage from gradients. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[Zou *et al.*, 2022] Tianyuan Zou, Yang Liu, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, Qiang Yang, and Ya-Qin Zhang. Defending batch-level label inference and replacement attacks in vertical federated learning. *IEEE Transactions on Big Data*, 2022.