Label-Free Backdoor Attacks in Vertical Federated Learning

Wei Shen*, **Wenke Huang***, **Guancheng Wan**, **Mang Ye**†

National Engineering Research Center for Multimedia Software
School of Computer Science, Wuhan University, China
{weishen, wenkehuang, guanchengwan, yemang}@whu.edu.cn

Abstract

Vertical Federated Learning (VFL) involves multiple clients collaborating to train a global model, with distributed features of shared samples. While it becomes a critical privacy-preserving learning paradigm, its security can be significantly compromised by backdoor attacks, where a malicious client injects a target backdoor by manipulating local data. Existing attack methods in VFL rely on the assumption that the malicious client can obtain additional knowledge about task labels, which is not applicable in VFL. In this work, we investigate a new backdoor attack paradigm in VFL, **L**abel-**F**ree **B**ackdoor **A**ttacks **(LFBA)**, which does not require any additional task label information and is feasible in VFL settings. Specifically, while existing methods assume access to task labels or target-class samples, we demonstrate that the gradients of local embeddings reflect the semantic information of labels. It can be utilized to construct the target poison sample set. Besides, we uncover that backdoor triggers tend to be ignored and under-fitted due to the learning of original features, which hinders backdoor task optimization. To address this, we propose selectively switching poison samples to disrupt feature learning, promoting backdoor task learning while maintaining accuracy on clean data. Extensive experiments demonstrate the effectiveness of our method in various settings.

Code — https://github.com/shentt67/LFBA/

1 Introduction

Vertical Federated Learning (VFL) (Hardy et al. 2017; Yang et al. 2019, 2023) has become a significant privacy-preserving collaboration paradigm. It involves training a global model with distributed features but shared samples across different clients, with only one client owning the task labels. Compared with Horizontal Federated Learning (HFL), where clients possess the same feature space (McMahan et al. 2017; Yang et al. 2019; Hu et al. 2023; Ye et al. 2023, 2024a,b; Huang et al. 2024; Wang et al. 2024b; Tan et al. 2024), VFL has shown promising results and applications particularly in cross-domain applications (Song et al. 2021; Huang, Wang, and Han 2023; Yan et al. 2024). However, despite adhering to privacy protocols, VFL remains vulnerable to security concerns, especially with malicious attacks when the trustworthiness of participants is uncertain. A crucial concern lies in the backdoor attacks (Liu et al. 2021b; Chen et al. 2023, 2024), which involve maliciously adding triggers to the data and inducing the target output. In VFL, the malicious client can introduce a backdoor by altering the local raw features to control the model behavior. For example, as shown in Figure 1, in a collaboration between a short video platform and an e-commerce platform, the e-commerce platform can inject backdoors into the VFL model and induce target recommendations in the short video platform by adding triggers.

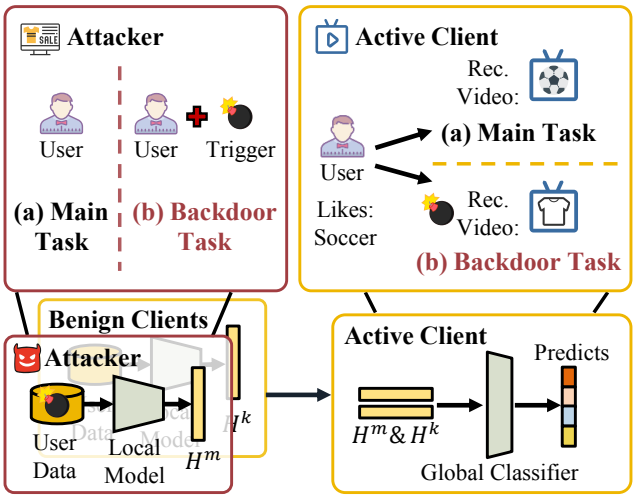


Figure 1: *An Example of Backdoor Attacks in VFL.* Consider the short video platform collaborates with the e-commerce company for recommending videos. The e-commerce company can act as the attacker to induce the target output of specific advertisement recommendations.

While backdoor attacks introduce security vulnerabilities in VFL, thoroughly investigating backdoor attacks in VFL is crucial for developing effective security measures. Unlike HFL scenarios where attackers have full access to samples and corresponding labels (Gu, Dolan-Gavitt, and Garg 2017; Li et al. 2022), executing backdoor attacks in VFL presents unique challenges: attackers lack access to task labels, complicating the execution of backdoor attacks. Existing research

*These authors contributed equally.
†Corresponding author.

has explored several backdoor attacks in VFL (Liu et al. 2021b; Gu and Bai 2023; Chen et al. 2023; He et al. 2023; Chen et al. 2024). For instance, Gu *et al.* propose LR-BA (Gu and Bai 2023), which trains a classifier with a few labeled samples and then minimizes the feature distance between the poison data and the target-class data. TECB (Chen et al. 2023) uses a universal trigger that is optimized with a few target-class samples in preset, and directly poisons the target-class sample to learn the target-trigger correspondence. However, these methods rely on strong preset knowledge of task labels, such as direct access to task labels or a predefined target-class sample set (one or more samples), which is often not feasible for attackers. Chen *et al.* (Chen et al. 2024) propose an approach that assumes the attacker knows the task labels involve an imbalanced binary classification problem. They suggest inferring labels by identifying head-class samples with the largest gradients and optimizing a universal trigger. However, it requires prior knowledge and makes strong assumptions about the prediction tasks, limiting its applicability.

Existing works rely on knowledge or strong assumptions of task labels, which are typically unavailable to attackers in VFL. Beyond these limitations, the key challenge in executing backdoor attacks in VFL is: *How can triggers be associated with the backdoor target without accessing task labels?* We identify clean-label attacks (Turner, Tsipras, and Madry 2018; Zhao et al. 2020; Huynh et al. 2024) as a suitable solution, as they directly poison target-class samples without altering the labels. To construct a target poison set, we leverage the fact that embedding gradients, which are related to task label information, are returned to all clients, including the attacker. Inspired by this, we introduce **G**radient-Guided **P**oison-Set **C**onstruction **(GPC)**, which builds the target poison set by calculating consistency of embedding gradients. The attacker defines a local anchor sample, and the attack goal is to classify trigger samples into the anchor class. Samples with the most consistent embedding gradients to the anchor are included in the poison set. By using embedding gradients as guidance, the attacker can construct a poison sample set consistently drawn from the target class, enabling backdoor attacks without task labels.

Besides, to effectively inject backdoors in VFL, we propose a novel poison method **S**electively S**A**mple S**W**itching **(SAW)**. When optimizing the backdoor tasks, the trigger and the target should be associated. However, since the chosen samples consistently come from the backdoor target, we argue that the model will focus on learning the reflection between the origin features and the target, leading to the trigger being ignored and under-fitted. Motivated by it, we propose an intuitive solution to disrupt feature learning, which encourages the model to focus on learning the trigger. Specifically, we switch the poison samples with other local samples, to disturb the feature learning and enhance backdoor optimization. Additionally, to maintain the clean data accuracy, we purposefully select only a subset of samples with the maximum gradients in the poison set to add the trigger and perform the switching. These are considered 'hard samples', and manipulating them has minimal impacts on the benign performance, achieving a better trade-off between the main task and the backdoor task.

In summary, our contributions can be outlined as follows:

- We propose a label-free backdoor attack method that is applicable in the VFL setting. Specifically, we construct the target-class poison set guided by embedding gradients, choosing samples consistently from the target class without requiring additional knowledge of task labels.
- We introduce a novel poison method that selectively switches the poison samples. It disrupts the original feature learning, enhances trigger learning, and achieves a better trade-off between the main and backdoor tasks.
- We conduct extensive experiments to demonstrate that our proposed method **L**abel-**F**ree **B**ackdoor **A**ttacks **(LFBA)**, is effective to perform backdoor attacks in various settings, without additional knowledge for task labels.

2 Related Work

Vertical Federated Learning. Vertical Federated Learning (VFL) (Hardy et al. 2017; Yang et al. 2023; Liu et al. 2024c; Ye et al. 2024b) is a privacy-preserving learning paradigm, where participants share overlapping sample spaces but have distinct data feature spaces. It has been widely explored in recent research (Zhang et al. 2021; Wu, Li, and He 2022; Wu, Hou, and He 2024; Gao et al. 2024; Qiu et al. 2024; Wang et al. 2024a), showing promising results and potentials in cross-domain collaborations, such as finance (Zheng et al. 2020; Long et al. 2020), healthcare (Huang, Wang, and Han 2023; Song et al. 2021; Yan et al. 2024), and recommendation systems (Zhang and Jiang 2021; Yuan et al. 2022; Wei et al. 2023), among others (Jin et al. 2021; Liu et al. 2021a; Zhang and Jiang 2021; Fan et al. 2024; Shen, Ye, and Huang 2024; Ye et al. 2024c; Liang et al. 2024a,b). In this paper, we investigate backdoor attacks in VFL without task labels, providing insights for exploring security threats in VFL.

Backdoor Attacks. Backdoor attacks (Gu, Dolan-Gavitt, and Garg 2017; Li et al. 2022; Fang et al. 2024; Liu et al. 2024b,d; An et al. 2024) were first proposed to inject a fixed activation pattern into images, to target a specific class. The original method in (Gu, Dolan-Gavitt, and Garg 2017) involved changing the labels of random samples to the target class, and then adding a fixed trigger to the samples, thereby training the model to learn the correspondence between the trigger and the target class. Besides, some research focuses on backdoor attacks that do not require label manipulation, making them more stealthy, known as *clean-label attacks* (Turner, Tsipras, and Madry 2018; Zhao et al. 2020; Ning et al. 2021; Hu et al. 2022; Gao et al. 2023; Huynh et al. 2024). Clean-label attacks assume the attacker adds triggers to the samples of the target class, enabling the model to learn the trigger-target correspondence without altering the labels. In this work, we explore the paradigm of clean-label attacks, where the malicious client in VFL cannot alter the labels.

Backdoor Attacks in VFL. Backdoor attacks have been extensively studied in Horizontal Federated Learning (HFL) settings (Bagdasaryan et al. 2020; Lyu et al. 2023; Qin et al. 2024; Liu et al. 2024a), where malicious clients can directly add triggers and alter the labels of local samples. However, additional challenges make these attacks difficult to execute in VFL settings. The primary challenge is that task label information is inaccessible to the malicious client in VFL,

complicating the attack. Several works have explored backdoor attacks in VFL settings (Liu et al. 2021b; Gu and Bai 2023; Chen et al. 2023; He et al. 2023; Chen et al. 2024). Nevertheless, they assume that the attacker has preset knowledge or assumptions about the task labels, which is not feasible in VFL. In this work, we propose a label-free approach to perform backdoor attacks, which is feasible in VFL.

3 Preliminary

3.1 Formal Problem Definition

Objective of Vertical Federated Learning. In VFL, clients share overlapping sample spaces but possess private feature spaces. Each client holds a local model to extract embeddings of origin data. The goal is to collaboratively train a prediction model where only one client, i.e., *the active client,* holds the task labels. The other clients, i.e., *the passive clients*, send embeddings of the shared samples to the active client and participate in training the global prediction model. The gradients are then sent back to each client for local model updates. Define K as the number of clients, and N is the shared samples discovered by alignment protocols (Hardy et al. 2017) across clients, defined as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ is the raw data with d dimensions, and y_i is the corresponding label. The features of each sample x_i are distributed across clients as $x_i = \{x_i^k\}_{k=1}^{K}$, with $x_i^k \in \mathbb{R}^{d^k}$. The sample features in each client P^k can be defined as $D^k = \{x_i^k\}_{k=1}^{K}$. The objective of VFL can be formulated:

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(G(H_i^1, ..., H_i^K; \theta^g), y_i), \tag{1}$$

where each client P^k holds a local model $f_k(\cdot; \theta^k)$ that computes the embeddings $H_i^k = f_k(x_i^k; \theta^k)$ from the raw features. The final prediction is made at the active client $P^a, a \in \{1, ..., K\}$ with a global model $G(\cdot; \theta^g)$. The parameters of the overall VFL model are defined as $\Theta = \{\theta^1, ..., \theta^K, \theta^g\}$. Define $\mathcal{L}$ as the loss function, where a cross-entropy loss can be employed for classification tasks.

Objective of Backdoor Attacks in VFL. The goal of backdoor attacks is to establish the reflection between the designed trigger and the target class, with poisoned data from the attacker $P^m, m \in \{1, ..., K\}$ (the other benign clients are defined as $\{P^k\}_{k=1}^{K} - \{P^m\}$). Besides, the performance on clean data is maintained. Define the whole VFL model as $F(\cdot; \Theta)$, the objective of backdoor attacks can be formulated:

$$\min_{\Theta} \underbrace{\frac{1}{N_c} \sum_{i=1}^{N_c} \mathcal{L}(F(x_i; \Theta), y_i)}_{\text{Main Task}} + \underbrace{\frac{1}{N_p} \sum_{i=1}^{N_p} \mathcal{L}(F(x_i + \delta; \Theta), \tau)}_{\text{Backdoor Task}}, \tag{2}$$

where N_c and N_p represent the numbers of samples in the clean dataset $D_c \subseteq D$ and poisoned sample set $D_p \subseteq D$ with $D_c \cup D_p = D$. The trigger for the backdoor attacks is denoted by δ and the backdoor target is denoted by τ. In previous backdoor attacks, data poisoning often relies on knowledge of the target label information, where sample labels were altered to the target, or the poison set was selected based on the sample labels belonging to the target class. However, in VFL, the sample labels of collaboration tasks are private information, posing challenges for the attacker to alter the labels or construct the poison set. To overcome these limitations, in this work, we explore a backdoor attack approach applicable to VFL in a label-free manner.

3.2 Threat Model

We explore the scenario where one of the passive clients in VFL assumes the attacker. It is a plausible assumption that malicious clients intend to inject their intended backdoor into the VFL model while obeying the VFL protocol.

Attacker Capability. As a passive client, the attacker has access to its local data, model, and the gradients of intermediate embeddings during the training process. The attacker does not possess any information about the task labels. It can manipulate all of its local data and model parameters, but cannot alter any labels in the active client.

Attacker Objective. The goal of the attacker is to inject a target backdoor into the VFL model, which will be activated by a predefined trigger pattern. Once the VFL model is deployed, the attacker can add the trigger to the local data of samples, and the final predictions in the active client will be intentionally misclassified to the backdoor target. However, the classification accuracy on clean data should be maintained.

4 Methodology

To bypass label limitations in VFL, we follow a clean-label setting (Turner, Tsipras, and Madry 2018; Zhao et al. 2020), which involves: (1) creating a poison sample set from the target class; (2) adding triggers on these samples for model training to inject the backdoor. It avoids modifying sample labels, adhering to the fundamental assumptions in VFL. For step (1), we define an anchor sample and choose poison samples that are most consistent with the anchor based on their embedding gradients. For step (2), we selectively switch the local data of poison samples with other samples, enhancing attacks while preserving main task accuracy.

4.1 Gradient-Guided Poison-Set Construction

Motivation. In VFL, as a passive client, the task label information is not accessible to the attacker, presenting challenges for backdoor attacks. However, the attacker receives updated gradients of local embeddings, which closely relate to the sample labels. In this case, we leverage the embedding gradients as guidances to construct the target poison set, ensuring it consistently originates from the backdoor target.

Gradient-Guided Poison-Set Construction. To effectively inject the backdoor into the VFL model, the attacker first chooses an anchor sample as guidance. Denote the anchor sample as x_r, with features x_r^m distributed to the attacker client. The backdoor target τ is set to the class of the anchor sample y_r, and the label is only possessed in the active client. During training, each client sends local embeddings H^k and receives corresponding gradients calculated by the active client. Similarly, the attacker client sends the embeddings $\{H_i^m\}_{i=1}^{N}$ computed from the local data $D^m = \{x_i^m\}_{i=1}^{N}$.

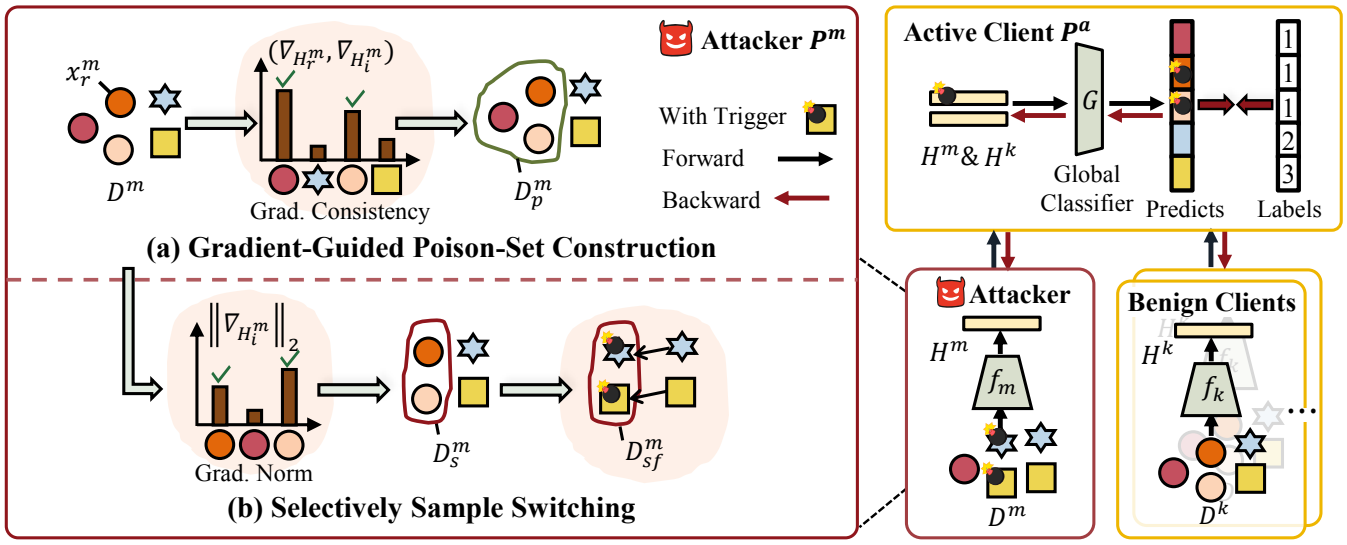


Figure 2: *The framework of Label-Free Backdoor Attacks (LFBA).* To construct the poison set that consistently from the backdoor target, *(a) Gradient-Guided Poison-Set Construction (GPC in Section 4.1):* the attacker chooses an anchor sample locally, and the samples with maximum consistency are chosen for the poison set. They are consistently from the same class of the anchor, i.e., the backdoor target; To effectively poison samples for injecting target backdoor, *(b) Selectively Sample Switching (SAW in Section 4.2):* the attacker selectively switches the poison samples that are with maximum embedding gradients, which promotes the backdoor optimization by disturbing the origin feature learning while maintaining the main task accuracy.

Subsequently, the attacker can obtain the embedding gradients $\{\nabla_{H_i^m}\}_{i=1}^N$ from the active client. The gradients of embeddings can be calculated by:

$$\nabla_{H_i^k} = \frac{\partial \mathcal{L}(G(H_i^1, ..., H_i^K; \theta^g), y_i)}{\partial H_i^k}. \tag{3}$$

Given that the embedding gradients are calculated with sample labels and retain the label information, we utilize the gradients as guidance to construct a sample set consistent with the backdoor target. Specifically, we calculate the consistency of embedding gradients between the anchor and other samples and construct the target set comprising samples with the highest consistency. The construction process can be defined as follows:

$$D_p = \arg\max_{D_p} \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{\nabla_{H_r^m} \cdot \nabla_{H_i^m}}{\|\nabla_{H_r^m}\|_2 \|\nabla_{H_i^m}\|_2}, \tag{4}$$

where $\nabla_{H_r^m}$ is the gradient of the anchor sample embedding H_r^m. The target set of chosen samples is denoted as D_p, containing N_p samples with the ratio $p = \frac{N_p}{N}$. Concretely, we construct the poison set during the initial training epoch and maintain the same set for subsequent training rounds. Guided by the embedding gradients, the selected samples are intended to consistently with the same label as the anchor sample, which is the backdoor target. Utilizing the poison set consistently from the backdoor target and adding designed triggers, the VFL model can learn the association between the backdoor target and the triggers without the need to alter any labels. During the construction process, the attacker has no access to label information about the backdoor target, which aligns with the constraints in the VFL setting.

4.2 Selectively Sample Switching

Motivation. After obtaining the target sample set, the attacker can add the designed triggers to the chosen samples and inject the target backdoor directly. Consider the samples from the backdoor target τ in D_p, the backdoor task can be formulated:

$$\min_{\Theta} \frac{1}{N_p} \sum_{i=1}^{N_p} \mathcal{L}(F(\underbrace{x_i}_{\text{Feature Learning}} + \underbrace{\delta}_{\text{Trigger Learning}}; \Theta), \tau). \tag{5}$$

To inject the target backdoor, the reflection between the trigger δ and the target τ is expected to be established, i.e., *Trigger Learning*. However, with the poison samples consistently chosen from the backdoor target τ, the model will focus on *Feature Learning*: the correspondence between the original features and the target. In this case, the trigger will be ignored and under-fitted, hindering the optimization of trigger learning. To address this limitation, an intuitive solution is to disturb the feature learning, thereby encouraging the VFL model to focus on learning triggers.

Selectively Sample Switching. Within the attacker capability, we propose to switch the poison samples with other samples to disturb feature learning, promoting backdoor task optimization. We first switch each poison sample features x_i^m with the local data of other samples $x_j^m \in \{D - D_p\}$ not in the poison set. Then a designed trigger is added to the local data of the poison samples. The poisoning process with switching can be formulated:

$$x_i^m \rightarrow x_j^m + \delta. \tag{6}$$

In this way, the feature learning of poison samples is disrupted, promoting the VFL model to focus on trigger learning

for enhancing attack performance. Besides, to maintain the accuracy on clean data, we selectively switch and add triggers to only a portion of the poison set samples. It minimizes the impact on original feature learning, achieving a better trade-off between main task accuracy and backdoor performance. Concretely, we select the hard samples with the maximum embedding gradients, which are considered difficult to learn and promote main task performance. The construction of the switching sample set can be defined as:

$$D_s = \arg\max_{D_s} \frac{1}{N_s} \sum_{i=1}^{N_s} \|\nabla_{H_i^m}\|_2, \tag{7}$$

where $D_s \subseteq D_p$ is the selected sample set to perform poisoning by switching and adding the trigger. The size of D_s is N_s, with a switch ratio of $s = \frac{N_s}{N}$. Define $[\cdot]$ as data concatenation, the final poisoned data can be formulated:

$$D_{sf} = \{([x_i^1, ..., x_j^m + \delta, ..., x_i^K], y_i)\}_{i=1}^{N_s}. \tag{8}$$

The modified training data D_f can be defined as follows:

$$D_f = D_c + (D_p - D_s) + D_{sf}. \tag{9}$$

With the final training data D_f, the target backdoor can be injected into the VFL model successfully without performance degradation on clean data, under the applicable label-free assumptions. For a clearer illustration, we provide an overview of the proposed attacks in Algorithm 1.

Algorithm 1: The framework of LFBA in VFL

Input: Initial training data D and VFL model $F(;\Theta_0)$; The active client P^a and malicious client P^m; The trigger δ.
Output: Trained VFL model $F(;\Theta_T)$, activated with trigger δ for target τ.

for epoch $t \leftarrow 1, ..., T$ **do**
  for all clients $P^k \leftarrow P^1, ..., P^K$ in parallel **do**
    Compute $H^k = f_k(x^k; \theta_t^k)$;
    Send H^k to P^a.
  end for
  for active client P^a **do**
    $L_t = \mathcal{L}(G(H_i^1, ..., H_i^K; \theta_t^g), y_i)$;
    Return ∇_{H^k} calculated by Equation (3);
    Update global model via $\nabla_{\theta_t^g} = \frac{\partial L_t}{\partial \theta_t^g}$.
  end for
  for attacker P^m **do**
    if t==1 **then**
      Choose an anchor in local data x_a^m;
      Construct D_p through Equation (4);
    end if
    Poison D into D_f by Equation (6)-Equation (9).
  end for
  for all clients $P^k \leftarrow P^1, ..., P^K$ in parallel **do**
    Update local model with $\nabla_{\theta_t^k} = \frac{\partial \nabla_{H^k}}{\partial \theta_t^k}$.
  end for
end for

4.3 Discussion and Limitation

Existing Attacks with Gradients in FL. Existing methods in FL also use gradient information to aid backdoor attacks. Several works (Sun et al. 2019; Wang et al. 2020) propose performing projected gradient updates in the attacker client, where the attack model stays close to the global model. Yoo *et al.* (Yoo and Kwak 2022) suggest using gradient ensembling from multiple poison rounds to improve attack generalization. Nguyen *et al.* (Nguyen et al. 2024) leverage historical gradient variations to pick infrequently updated neurons for poisoning, reducing the dilution effect from benign clients. In this work, we propose constructing the poison set by embedding gradients in VFL, providing insights for future works.

Existing Attacks in VFL. Existing methods rely on prior knowledge or assumptions about task labels. In contrast, we design several baselines without additional knowledge for comparison. This highlights the effectiveness of our method with each key component. Please refer to Section 5 for details.

Limitations. However, our method LFBA may face challenges in certain situations: (1) If the number of clients increases and the attacker features remain extremely limited, the attack performance may decrease. (2) The sample switching may result in sub-optimal performance due to discrepancies between the switching features and the backdoor target.

5 Experiment

Datasets. We evaluate our method on four real-world datasets, with data distributed to multiple clients, and only the active client holds the task labels: (1) *NUS-WIDE* (Chua et al. 2009): A multi-modal dataset contains 1000 text features and 634 image features, labeled with multiple classes. We use a five-class subset including 'buildings', 'grass', 'animal', 'water', and 'person', with 69966 training samples and 46693 testing samples. (2) *UCIHAR* (Anguita et al. 2013): A human activity recognition dataset with six classes: 'walking', 'walking upstairs', 'walking downstairs', 'sitting', 'standing', and 'laying', with 7352 training samples and 2947 testing samples. (3) *Phishing* (Asuncion, Newman et al. 2007): It provides 30 features indicating whether a website is a phishing website, with 8844 training samples and 2211 test samples. (4) *CIFAR-10* (Krizhevsky, Hinton et al. 2009): It is an image dataset for 10 classification tasks with 50000 training samples and 10000 testing samples. We conduct evaluations with two-client settings ($K = 2$) and four-client settings ($K = 4$). On the NUS-WIDE dataset, image features and text features are distributed separately to different clients when $K = 2$. In other cases, the features are equally partitioned to all clients.

Baselines. We compare several applicable baselines without requiring task labels in VFL: (1) *Vanilla*: The VFL baseline without attacks. (2) *DGPC*: Construct the poison set D_p by GPC and directly add triggers. (3) *RGPC*: Randomly select N_s samples in D_p to add triggers. (4) *RS-GPC*: Randomly select N_s samples in D_p to switch and add triggers.

Evaluation Metrics. We evaluate the attack performance with three metrics: the accuracy ($\mathcal{M}$) on clean data and the attack success rate ($\mathcal{A}$) of poison data (Gu, Dolan-Gavitt, and Garg 2017). Additionally, we use the mean values of ($\mathcal{M}$) and ($\mathcal{A}$), referred to as $\mathcal{V}$, to assess the trade-off performance.

Models. In all experiments, each client employs a local model

Methods	*NUS-WIDE*						*UCI-HAR*					
	$K=2$			$K=4$			$K=2$			$K=4$		
	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$
Vanilla	83.98	12.36	48.17	82.56	2.27	42.25	92.98	4.48	48.73	90.70	2.73	46.72
DGPC	83.56	98.30	90.93	80.23	85.85	83.04	87.75	77.28	82.52	88.90	86.12	87.51
RGPC	83.86	96.86	90.36	82.21	81.60	80.29	86.33	66.33	76.33	90.50	84.40	87.45
RS-GPC	83.46	98.55	91.01	82.26	86.45	84.36	91.13	98.72	94.93	89.93	88.80	89.37
LFBA	83.93	99.85	**91.89**	82.36	95.13	**88.75**	91.99	99.96	**95.98**	90.69	90.63	**90.66**

Methods	*Phishing*						*CIFAR-10*					
	$K=2$			$K=4$			$K=2$			$K=4$		
	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$
Vanilla	95.12	2.22	48.67	92.76	2.55	47.66	78.22	3.00	40.61	73.66	4.26	38.96
DGPC	93.26	78.16	85.71	90.90	76.59	83.75	70.39	96.48	83.43	69.77	82.40	76.09
RGPC	93.71	77.67	85.69	91.68	70.23	80.96	71.17	94.87	83.02	71.13	80.02	75.58
RS-GPC	94.30	81.36	87.83	91.63	77.98	84.81	77.12	97.18	87.15	70.20	91.94	81.07
LFBA	93.76	83.09	**88.42**	91.95	80.74	**86.35**	77.68	98.22	**87.95**	72.33	93.47	**82.90**

Table 1: *Comparisons with Baselines.* Bold represents the highest accuracy.

$p=\frac{N_p}{N}$	*NUS-WIDE*		
	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$
Vanilla	83.98	12.36	48.17
0.02	83.86	99.49	91.67
0.06	83.73	99.86	91.79
0.1	83.93	99.85	**91.89**
0.14	83.76	99.97	91.87
0.18	83.03	99.81	91.42
0.2	82.79	95.91	89.35

(a) Different Poison Ratios.

$s=\frac{N_s}{N}$	*NUS-WIDE*		
	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{V}$
Vanilla	83.98	12.36	48.17
0.1	83.62	99.70	91.66
0.3	83.93	99.85	**91.89**
0.5	83.73	99.94	91.84
0.7	83.63	99.97	91.80
0.9	83.57	99.96	91.77
1	83.54	99.94	91.74

(b) Different Switch Ratios.

Table 2: *Ablation with Different N_p and N_s.* Our attack method is effective with different numbers of poison samples.

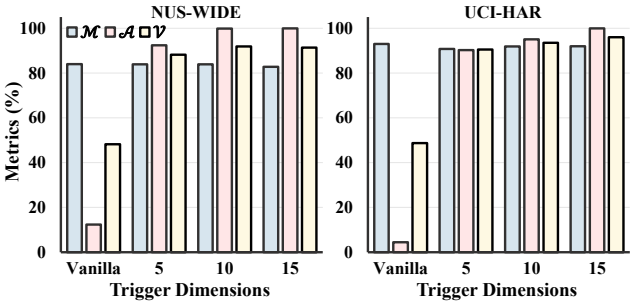


Figure 3: *Ablation with Different Trigger Sizes.* Our attack method is effective with different trigger sizes.

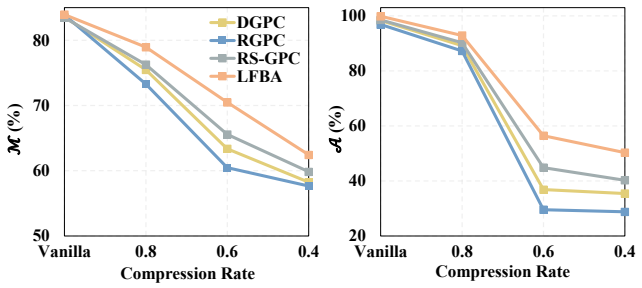
to extract embeddings, while the active client employs an additional global model for final predictions. For the NUS-WIDE and UCI-HAR datasets, we utilize a 4-layer linear model as the local model and a 3-layer model for the global model. For the Phishing dataset, we use a 2-layer model for both the local model and the global prediction model. For CIFAR-10, we utilize the ResNet18 (He et al. 2016) as the local model and a 3-layer linear model for the global model.

Implement Details. We randomly set several dimensions to a fixed value as the triggers on the NUS-WIDE, UCI-HAR, and Phishing datasets. For CIFAR-10, we utilize the same trigger pattern as BadNets (Gu, Dolan-Gavitt, and Garg 2017). The backdoor target is consistent with the anchor sample class: on the NUS-WIDE and UCI-HAR datasets, it is the fourth class, e.g., 'water' in the NUS-WIDE dataset; on the Phishing dataset, it is 'not the phishing website'. For CIFAR-10, the target is the seventh class, i.e., 'frog'. All models are trained until convergence using the Adam optimizer (Kingma and Ba 2015) with a batch size of 256. The learning rate of all models is set to 0.001 for the NUS-WIDE and CIFAR-10 datasets, and 0.003 for the UCI-HAR and Phishing datasets. The poison sample ratio $p=\frac{N_p}{N}$ is set between 0.1 and 0.3, and the switching sample ratio $s=\frac{N_s}{N_p}$ is set between 0 and 1 (e.g., $p=0.1$ and $s=0.3$ for the NUS-WIDE dataset).

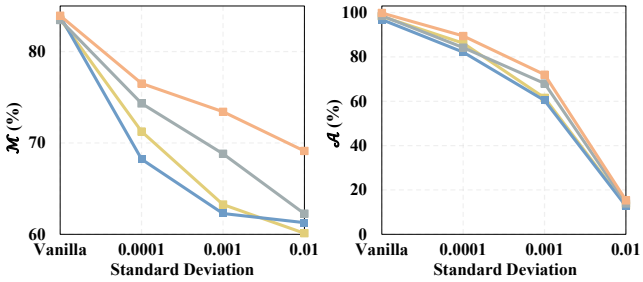
5.1 Ablation Study

Comparison with Baselines. We provide comparisons with baseline methods and present results in Table 1, where LFBA achieves high attack success rates, up to 92.64% on average, without significant degradation in main task accuracy. Comparing DGPC with Vanilla, the constructed poison set can be utilized to effectively inject backdoors, resulting in an average ASR of 85.15%. Comparing RGPC with RS-GPC shows an average gain of 8.63% in ASR, demonstrating that the sample switching method is effective for enhancing attacks. Comparing LFBA with RS-GPC, the trade-off performance increases by 1.55% on average, indicating that the selective switching strategy is effective for a better trade-off.

Ablation with Different N_p and N_s. To investigate the impacts of poison sample ratio p and switch sample ratio s, we conduct two ablation experiments: (a) we change the poison ratios p while keeping the switch ratio fixed at 0.3, and (b) we change the switch ratio s while keeping p fixed at 0.1. As shown in Table 2a, the attack performance of LFBA remains



(a) Gradient Compression.



(b) Gaussian Noise.

Figure 4: *Experiments with Defenses.* The results show the effectiveness of LFBA under different defense strategies.

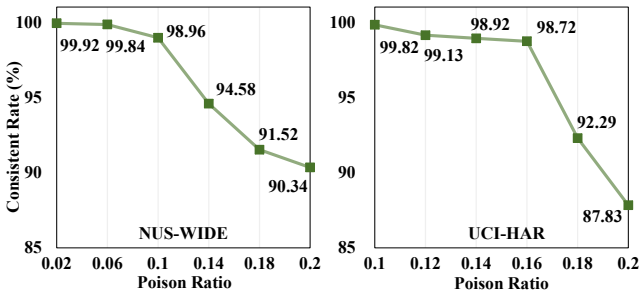
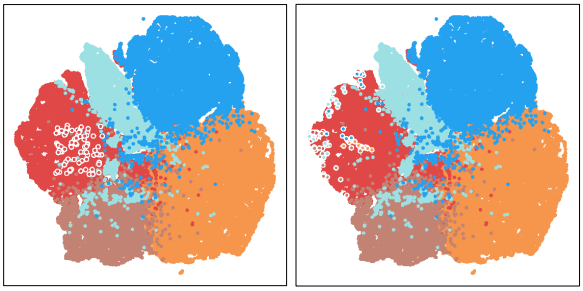


Figure 5: *Consistent Rate with Different Poison Ratios p.* The constructed poison set is consistently from the target class.



(a) Poison-Set Visualization. (b) Attack Visualization.

Figure 6: *Feature Visualizations.* (a) The samples in the poison set are consistently from the target class. (b) The samples with triggers are misclassified into the target class.

effective with changes in p, although decreases with relatively large values, e.g., $p = 0.2$. It is because the consistency of the constructed poison set decreases, hindering the trigger learning. However, a small sample set is usually preferred to ensure the attack is stealthy. As shown in Table 2b, the performance of LFBA remains stable and effective across different s values. Both experiments show the effectiveness of our method with various poison and switch ratios.

Ablation with Different Trigger Sizes. We conduct ablations on the NUS-WIDE and UCI-HAR datasets, to explore the impacts of trigger sizes. As depicted in Figure 3, backdoor attacks can be successful (over 90%) with different trigger sizes even when only 0.03% of the original features are changed to the fixed trigger value (5 trigger dimensions vs. 1634 original feature dimensions on the NUS-WIDE dataset).

5.2 Extended Analysis

Attack under Defenses. To evaluate the attack effectiveness under defenses, we apply two defense strategies and test LFBA on four datasets: gradient compression (Shokri and Shmatikov 2015; Lin et al. 2018; Fu et al. 2022), and adding Gaussian noise to the gradients (Fu et al. 2022). We use the compression rates of 0.8, 0.6, and 0.4, and the Gaussian noise standard deviations of 0.0001, 0.001, and 0.01 for evaluation. As depicted in Figure 4, although stronger defenses can reduce the attack success rates, they also degrade main task performance significantly, indicating the defenses fail.

Consistent Rate of Poison Set. We calculate the consistent rate of the constructed poison set D_p, which indicates the proportion of samples that belong to the class of the anchor sample (backdoor target). As illustrated in Figure 5, the constructed sample set consistently includes samples from the attack target across different poison ratios p, demonstrating that GPC effectively constructs the target poison set.

Visualization Results. We further visualize the features acquired from the VFL model with the injected backdoor. In Figure 6a, we visualize parts of samples in poison set D_p, which are consistently from the same class (backdoor target). Besides, we poison several samples with triggers and visualize them in Figure 6b, where the poisoned samples cluster into the backdoor target, indicating successful attacks.

6 Conclusion

In Vertical Federated Learning (VFL), the trained model may be vulnerable to backdoor attacks, where the final output is rendered to the backdoor target once the specific triggers are added. In this paper, we investigate an applicable backdoor attack in VFL that does not require knowledge of the prediction tasks or any label information. Specifically, we use the gradients of embeddings as guidance to construct a consistent poison set for the backdoor target. Additionally, we propose selectively switching the sample features of the poison set to enhance backdoor task optimization, and achieve a better trade-off between the main task and the backdoor task. This research provides valuable insights for executing backdoor attacks in VFL, and will induce potential implications for real-world security applications and studies.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 62176188, 62225113, 623B2080). The numerical calculations in this paper have been supported by the super-computing system in the Supercomputing Center of Wuhan University.

References

An, S.; Chou, S.-Y.; Zhang, K.; Xu, Q.; Tao, G.; Shen, G.; Cheng, S.; Ma, S.; Chen, P.-Y.; Ho, T.-Y.; et al. 2024. Elijah: Eliminating backdoors injected in diffusion models via distribution shift. In *AAAI*, 10847–10855.

Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J. L.; et al. 2013. A public domain dataset for human activity recognition using smartphones. In *Esann*, 3.

Asuncion, A.; Newman, D.; et al. 2007. UCI machine learning repository.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *AISTATS*, 2938–2948.

Chen, P.; Du, X.; Lu, Z.; and Chai, H. 2024. Universal adversarial backdoor attacks to fool vertical federated learning. *Computers & Security*, 103601.

Chen, P.; Yang, J.; Lin, J.; Lu, Z.; Duan, Q.; and Chai, H. 2023. A practical clean-label backdoor attack with limited information in vertical federated learning. In *ICDM*, 41–50.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 1–9.

Fan, Z.; Fang, H.; Zhou, Z.; Pei, J.; Friedlander, M. P.; and Zhang, Y. 2024. Fair and efficient contribution valuation for vertical federated learning. In *ICLR*.

Fang, J.; Zhang, G.; Cui, Q.; Tang, C.; Gu, L.; Li, L.; Gu, J.; and Zhou, J. 2024. Backdoor Adjustment via Group Adaptation for Debiased Coupon Recommendations. In *AAAI*, 11944–11952.

Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Zhou, J.; Liu, A. X.; and Wang, T. 2022. Label inference attacks against vertical federated learning. In *USENIX Security 22*, 1397–1414.

Gao, D.; Wan, S.; Fan, L.; Yao, X.; and Yang, Q. 2024. Complementary Knowledge Distillation for Robust and Privacy-Preserving Model Serving in Vertical Federated Learning. In *AAAI*, 19832–19839.

Gao, Y.; Li, Y.; Zhu, L.; Wu, D.; Jiang, Y.; and Xia, S.-T. 2023. Not all samples are born equal: Towards effective clean-label backdoor attacks. *Pattern Recognition*, 109512.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Gu, Y.; and Bai, Y. 2023. LR-BA: Backdoor attack against vertical federated learning using local latent representations. *Computers & Security*, 103193.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, Y.; Shen, Z.; Hua, J.; Dong, Q.; Niu, J.; Tong, W.; Huang, X.; Li, C.; and Zhong, S. 2023. Backdoor Attack Against Split Neural Network-Based Vertical Federated Learning. *TIFS*.

Hu, M.; Xia, Z.; Yan, D.; Yue, Z.; Xia, J.; Huang, Y.; Liu, Y.; and Chen, M. 2023. GitFL: Uncertainty-Aware Real-Time Asynchronous Federated Learning Using Version Control. In *RTSS*, 145–157.

Hu, S.; Zhou, Z.; Zhang, Y.; Zhang, L. Y.; Zheng, Y.; He, Y.; and Jin, H. 2022. Badhash: Invisible backdoor attacks against deep hashing with clean label. In *ACM MM*, 678–686.

Huang, C.-j.; Wang, L.; and Han, X. 2023. Vertical federated knowledge transfer via representation distillation for healthcare collaboration networks. In *WWW*, 4188–4199.

Huang, W.; Ye, M.; Shi, Z.; Wan, G.; Li, H.; Du, B.; and Yang, Q. 2024. Federated learning for generalization, robustness, fairness: A survey and benchmark. *TPAMI*.

Huynh, T.; Nguyen, D.; Pham, T.; and Tran, A. 2024. COMBAT: Alternated Training for Effective Clean-Label Backdoor Attacks. In *AAAI*, 2436–2444.

Jin, X.; Chen, P.-Y.; Hsu, C.-Y.; Yu, C.-M.; and Chen, T. 2021. Cafe: Catastrophic data leakage in vertical federated learning. In *NeurIPS*, 994–1006.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Y.; Jiang, Y.; Li, Z.; and Xia, S.-T. 2022. Backdoor learning: A survey. *TNNLS*, 5–22.

Liang, K.; Meng, L.; Liu, Y.; Liu, M.; Wei, W.; Liu, S.; Tu, W.; Wang, S.; Zhou, S.; and Liu, X. 2024a. Simple Yet Effective: Structure Guided Pre-trained Transformer for Multi-modal Knowledge Graph Reasoning. In *ACM MM*, 1554–1563.

Liang, K.; Meng, L.; Zhou, S.; Tu, W.; Wang, S.; Liu, Y.; Liu, M.; Zhao, L.; Dong, X.; and Liu, X. 2024b. MINES: Message Intercommunication for Inductive Relation Reasoning over Neighbor-Enhanced Subgraphs. In *AAAI*, 10645–10653.

Lin, Y.; Han, S.; Mao, H.; Wang, Y.; and Dally, W. J. 2018. Deep gradient compression: Reducing the communication bandwidth for distributed training.

Liu, T.; Zhang, Y.; Feng, Z.; Yang, Z.; Xu, C.; Man, D.; and Yang, W. 2024a. Beyond traditional threats: A persistent backdoor attack on federated learning. In *AAAI*, 21359–21367.

Liu, X.; Jia, X.; Gu, J.; Xun, Y.; Liang, S.; and Cao, X. 2024b. Does few-shot learning suffer from backdoor attacks? In *AAAI*, 19893–19901.

Liu, Y.; Fan, T.; Chen, T.; Xu, Q.; and Yang, Q. 2021a. Fate: An industrial grade platform for collaborative learning with data protection. *JMLR*, 1–6.

Liu, Y.; Kang, Y.; Zou, T.; Pu, Y.; He, Y.; Ye, X.; Ouyang, Y.; Zhang, Y.-Q.; and Yang, Q. 2024c. Vertical federated learning: Concepts, advances, and challenges. *TKDE*.

Liu, Y.; Zou, T.; Kang, Y.; Liu, W.; He, Y.; Yi, Z.; and Yang, Q. 2021b. Batch label inference and replacement attacks in black-boxed vertical federated learning. *arXiv preprint arXiv:2112.05409*.

Liu, Z.; Wang, T.; Huai, M.; and Miao, C. 2024d. Backdoor attacks via machine unlearning. In *AAAI*, 14115–14123.

Long, G.; Tan, Y.; Jiang, J.; and Zhang, C. 2020. Federated learning for open banking. In *Federated learning: privacy and incentive*, 240–254.

Lyu, X.; Han, Y.; Wang, W.; Liu, J.; Wang, B.; Liu, J.; and Zhang, X. 2023. Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning. In *AAAI*, 9020–9028.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 1273–1282.

Nguyen, T. D.; Nguyen, T. A.; Tran, A.; Doan, K. D.; and Wong, K.-S. 2024. Iba: Towards irreversible backdoor attacks in federated learning. In *NeurIPS*.

Ning, R.; Li, J.; Xin, C.; and Wu, H. 2021. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *INFOCOM*, 1–10.

Qin, Z.; Chen, F.; Zhi, C.; Yan, X.; and Deng, S. 2024. Resisting Backdoor Attacks in Federated Learning via Bidirectional Elections and Individual Perspective. In *AAAI*, 14677–14685.

Qiu, P.; Pu, Y.; Liu, Y.; Liu, W.; Yue, Y.; Zhu, X.; Li, L.; Li, J.; and Ji, S. 2024. Integer Is Enough: When Vertical Federated Learning Meets Rounding. In *AAAI*, 14704–14712.

Shen, W.; Ye, M.; and Huang, W. 2024. Resisting Over-Smoothing in Graph Neural Networks via Dual-Dimensional Decoupling. In *ACM MM*, 5800–5809.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *CCS*, 1310–1321.

Song, Y.; Xie, Y.; Zhang, H.; Liang, Y.; Ye, X.; Yang, A.; and Ouyang, Y. 2021. Federated learning application on telecommunication-joint healthcare recommendation. In *ICCT*, 1443–1448.

Sun, Z.; Kairouz, P.; Suresh, A. T.; and McMahan, H. B. 2019. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*.

Tan, Z.; Wan, G.; Huang, W.; and Ye, M. 2024. FedSSP: Federated Graph Learning with Spectral Knowledge and Personalized Preference. In *NeurIPS*.

Turner, A.; Tsipras, D.; and Madry, A. 2018. Clean-label backdoor attacks.

Wang, G.; Gu, B.; Zhang, Q.; Li, X.; Wang, B.; and Ling, C. X. 2024a. A unified solution for privacy and communication efficiency in vertical federated learning. In *NeurIPS*.

Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.-y.; Lee, K.; and Papailiopoulos, D. 2020. Attack of the tails: Yes, you really can backdoor federated learning. In *NeurIPS*, 16070–16084.

Wang, H.; Xu, H.; Li, Y.; Xu, Y.; Li, R.; and Zhang, T. 2024b. FedCDA: Federated Learning with Cross-rounds Divergence-aware Aggregation. In *ICLR*.

Wei, P.; Dou, H.; Liu, S.; Tang, R.; Liu, L.; Wang, L.; and Zheng, B. 2023. Fedads: A benchmark for privacy-preserving cvr estimation with vertical federated learning. In *SIGIR*, 3037–3046.

Wu, Z.; Hou, J.; and He, B. 2024. VertiBench: Advancing feature distribution diversity in vertical federated learning benchmarks. In *ICLR*.

Wu, Z.; Li, Q.; and He, B. 2022. A coupled design of exploiting record similarity for practical vertical federated learning. In *NeurIPS*, 21087–21100.

Yan, Y.; Wang, H.; Huang, Y.; He, N.; Zhu, L.; Xu, Y.; Li, Y.; and Zheng, Y. 2024. Cross-modal vertical federated learning for mri reconstruction. *JBHI*.

Yang, L.; Chai, D.; Zhang, J.; Jin, Y.; Wang, L.; Liu, H.; Tian, H.; Xu, Q.; and Chen, K. 2023. A survey on vertical federated learning: From a layered perspective. *arXiv preprint arXiv:2304.01829*.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *TIST*, 1–19.

Ye, M.; Fang, X.; Du, B.; Yuen, P. C.; and Tao, D. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *CSUR*, 56(3): 1–44.

Ye, M.; Huang, W.; Shi, Z.; Li, H.; and Bo, D. 2024a. Revisiting federated learning with label skew: An overconfidence perspective. *SCIS*.

Ye, M.; Shen, W.; Snezhko, E.; Kovalev, V.; Yuen, P. C.; and Du, B. 2024b. Vertical Federated Learning for Effectiveness, Security, Applicability: A Survey. *arXiv preprint arXiv:2405.17495*.

Ye, M.; Shen, W.; Zhang, J.; Yang, Y.; and Du, B. 2024c. Securereid: Privacy-preserving anonymization for person re-identification. *TIFS*.

Yoo, K.; and Kwak, N. 2022. Backdoor attacks in federated learning by rare embeddings and gradient ensembling. In *EMNLP*.

Yuan, H.; Ma, C.; Zhao, Z.; Xu, X.; and Wang, Z. 2022. A privacy-preserving oriented service recommendation approach based on personal data cloud and federated learning. In *ICWS*, 322–330.

Zhang, J.; and Jiang, Y. 2021. A vertical federation recommendation method based on clustering and latent factor model. In *EIECS*, 362–366.

Zhang, Q.; Gu, B.; Deng, C.; and Huang, H. 2021. Secure bilevel asynchronous vertical federated learning with backward updating. In *AAAI*, 10896–10904.

Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; and Jiang, Y.-G. 2020. Clean-label backdoor attacks on video recognition models. In *CVPR*, 14443–14452.

Zheng, F.; Li, K.; Tian, J.; Xiang, X.; et al. 2020. A vertical federated learning method for interpretable scorecard and its application in credit scoring. *arXiv preprint arXiv:2009.06218*.