TOWARDS PRIVACY-GUARANTEED LABEL UNLEARNING IN VERTICAL FEDERATED LEARNING: FEW-SHOT FORGETTING WITHOUT DISCLOSURE

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper addresses the critical challenge of unlearning in Vertical Federated Learning (VFL), a setting that has received far less attention than its horizontal counterpart. Specifically, we propose the first method tailored to *label unlearning* in VFL, where labels play a dual role as both essential inputs and sensitive information. To this end, we repurpose manifold mixup traditionally used as an augmentation technique into a privacy-preserving transformation that disguises label information in the shared embeddings. These augmented embeddings are then subjected to gradient-based label forgetting, effectively removing the associated label information from the model. To recover performance on the retained data, we introduce a recovery-phase optimization step that refines the remaining embeddings. This design achieves effective label unlearning while preserving privacy and maintaining computational efficiency. We validate our method through extensive experiments on diverse datasets, including MNIST, CIFAR-10, CIFAR-100, ModelNet, Brain Tumor MRI, COVID-19 Radiography, and Yahoo Answers demonstrate strong efficacy and scalability. Overall, this work establishes a new direction for unlearning in VFL, showing that re-imagining mixup as a privacy mechanism can unlock practical, privacy-preserving, and utility-preserving unlearning. Our code will be released publicly.

1 INTRODUCTION

Vertical Federated Learning (VFL) (Yang et al., 2019) enables multiple organizations to collaboratively utilize their private datasets in a privacy-preserving manner, even when they share some sample IDs but differ significantly in terms of features. In VFL, there are typically two types of parties: (i) the passive party, which holds the *features*, and (ii) the active party, which possesses the *labels*. VFL has seen the widespread application, especially in sensitive domains like banking and healthcare, where organizations benefit from joint modeling without exposing their raw data (Li et al., 2020).

A fundamental requirement in VFL is the necessity for unlearning, driven by the *"right to be forgotten"* as mandated by regulations such as GDPR[1] and CCPA[2]. While unlearning has been explored in Horizontal Federated Learning (HFL), there has been limited attention to its application in vertical settings. Existing studies on vertical federated unlearning (Li et al., 2024; Wang et al., 2024; Han et al., 2025) primarily address the removal of features when an entire passive client withdraws. In contrast, **this paper focuses on label unlearning**, which is particularly critical in scenarios where labels encode highly sensitive information. For example, in medical diagnostics, one may wish to *erase the label indicating whether a patient is HIV-positive*, as it reveals private health status. Similarly, in credit risk assessment systems, *revoking the label associated with a loan approval decision* helps safeguard against fairness and regulatory concerns.

However, our empirical results (see Sect. 3.2) show that directly applying current popular unlearning methods, such as retraining (Bourtoule et al., 2021; Foster et al., 2024a) or Boundary unlearning (Chen et al., 2023), poses a critical privacy risk. In VFL, the active party holds the labels, while the

[1] `https://gdpr-info.eu/art-17-gdpr/`
[2] `https://oag.ca.gov/privacy/ccpa`

passive party only holds features. Under the threat model, the passive party may act as an adversary and must not learn either the labels or which labels are being unlearned. Yet retraining requires the active party to explicitly identify to the passive party all samples tied to the deleted label (e.g., all "cat" samples) so that they can be excluded from training. This disclosure directly links the passive party's features to the deleted label identities, leading to complete leakage of sensitive information.

To overcome this challenge, we introduce a novel few-shot unlearning framework for VFL that removes labels from both active and passive models while relying only on a small public dataset. To achieve this, we repurpose manifold mixup (Verma et al., 2019) that is traditionally used as an augmentation technique into a privacy-preserving transformation that disguises label information in the shared embeddings. On these transformed embeddings, the active party performs gradient-based forgetting, while inverse gradients allow the passive party to unlearn the corresponding representations locally without accessing raw labels. A final recovery phase restores accuracy on the retained data. Our framework shows that by re-purposing mixup as a vehicle for label concealment, it opens a new direction for achieving efficient and privacy-preserving label unlearning in VFL.

Our proposed unlearning solution offers three key advantages: (i) **Privacy-Preserving Unlearning**: It requires only a small subset of labeled public data, significantly minimizing the risk of label privacy leakage; (ii) **Enhanced Unlearning Effectiveness**: By leveraging the manifold mixup technique, it achieves effective unlearning with minimal data; and (iii) **Computational Efficiency**: The proposed unlearning process is highly efficient, completing within seconds. The primary contributions of this work are as follows:

1. To the best knowledge, this is the first work to address the unlearning of labels in VFL. We first systematically elucidate the label privacy leakage that may occur when directly applying traditional machine unlearning methods for label unlearning in VFL (see Sect. 3.2).
2. We propose a novel few-shot label unlearning method that effectively removes labels from both the active and passive models in VFL using only a limited amount of public data. With this, our approach not only minimizes the risk of label privacy leakage by using a small number of data samples, but also enhances unlearning effectiveness through manifold mixup, ensuring a more robust and efficient unlearning process (see Sect. 4).
3. We conduct extensive experiments on diverse benchmark datasets, including MNIST (image), CIFAR-10 (image), CIFAR-100 (image), ModelNet (image), Brain Tumor MRI (image), COVID-19 Radiography (image) and Yahoo Answers (text). The results demonstrate that our approach rapidly and effectively unlearns target labels, outperforming existing machine unlearning techniques in both efficiency and effectiveness.

2 Related Works

This section reviews the related works in two aspects: machine unlearning & federated unlearning and vertical federated learning & privacy attacks in VFL. Please refer to Appendix A.8 for the comparison of our method with existing studies on Federated Unlearning.

2.1 Machine Unlearning (MU) & Federated Unlearning

Machine unlearning (MU) refers to the process of selectively removing the influences of specific data points from a machine learning (ML) model after it has been trained. This process ensures that the model "forgets" the contributions of specific data, as though the data had never been part of the training process.

MU was first proposed by (Cao & Yang, 2015) to remove specific data from a model without full retraining (Garg et al., 2020; Chen et al., 2021). It includes exact and approximate unlearning. Exact methods like SISA (Bourtoule et al., 2021) and ARCANE (Yan et al., 2022) partition data, train sub-models per partition, and retrain only the affected sections during unlearning. Approximate unlearning includes methods like fine-tuning (Golatkar et al., 2020a; Jia et al., 2023), random label updates (Graves et al., 2021; Chen et al., 2023), noise injection (Tarun et al., 2024; Huang et al., 2021), gradient ascent (Goel et al., 2022; Choi & Na, 2023; Abbasi et al., 2023; Hoang et al., 2024), knowledge distillation (Chundawat et al., 2023b; Zhang et al., 2023b; Kurmanji et al., 2023), and weight scrubbing (Golatkar et al., 2020a;b; 2021; Guo et al., 2020; Foster et al., 2024a). While these rely on data during unlearning, Chundawat et al. (Chundawat et al., 2023a) propose zero-shot unlearning, and Yoon et al. (Yoon et al., 2022) introduce a few-shot approach using model inversion.

In federated unlearning, most of the existing works are focusing in the horizontal environment (Wu et al., 2022; Gu et al., 2024a; Zhao et al., 2023; Romandini et al., 2024; Liu et al., 2024c; Zhang et al., 2023a; Su & Li, 2023; Ye et al., 2024; Gao et al., 2024; Cao et al., 2023b; Yuan et al., 2023; Alam et al., 2024; Li et al., 2023; Halimi et al., 2022; Xia et al., 2023; Wang et al., 2023a; Dhasade et al., 2023; Liu et al., 2022; Zhao et al., 2024; Wang et al., 2022; Gu et al., 2024b; Liu et al., 2021; Cao et al., 2023a). The research on horizontal federated unlearning (HFU) mainly focuses on label (Wang et al., 2022; Zhao et al., 2024), client (Liu et al., 2021; Yuan et al., 2023; Zhang et al., 2023a; Gao et al., 2024; Ye et al., 2024; Su & Li, 2023; Cao et al., 2023a; Wu et al., 2022) and sample unlearning (Liu et al., 2022; Dhasade et al., 2023). While HFU has been widely studied, vertical federated unlearning (VFU) remains underexplored. Existing works mainly focus on passive party unlearning, such as logistic regression (Deng et al., 2023), gradient boosting (Li et al., 2024), and deep learning models with fast retraining (Wang et al., 2024) or backdoor defence (Han et al., 2025). However, little attention has been given to active-party-initiated unlearning with all parties actively engaged in VFL.

2.2 Vertical Federated Learning & Privacy Attack in VFL

Vertical Federated Learning is a collaborative machine learning framework that enables multiple parties or organizations, each possessing **different attributes** of the **same set of users or entities**, to train a shared model without exchanging their raw data (Yang et al., 2019; Liu et al., 2024b). In VFL, privacy is of utmost importance because the participants are typically companies that handle valuable and sensitive user information. Hence, numerous studies have been conducted on privacy-preservation(Yang et al., 2024c; Xie et al., 2024; Wang et al., 2023b), privacy leakage attack (He et al., 2024; Liu et al., 2025; 2024a) and privacy leakage defend (Qiu et al., 2024; Wu et al., 2025).

There are mainly two types of privacy attacks in VFL: i) label inference attacks, and ii) feature inference attacks. Label inference attacks (Fu et al., 2022a; Li et al., 2021) are initiated by the passive party and aim to infer the label possessed by the active party. Feature inference attacks (He et al., 2019; Jin et al., 2021; Luo et al., 2021; Yang et al., 2024b), on the other hand, are initiated by the active party and aim to discover the features of the passive party.

3 Label Leakage During Vertical Federated Unlearning

This section introduces the general setup and discusses the risk of label leakage during the label unlearning process.

3.1 General Setup

VFL training. Vertical Federated Learning allows parties with different user attributes to collaboratively train a model without sharing raw data (Yang et al., 2019; Liu et al., 2024b). The active party holds the labels and active model, while passive parties provide features and passive models, enabling performance gains with privacy preservation. We assume that a VFL setting consists of one active party P_0 and K passive parties $\{P_1, \cdots, P_K\}$ who collaboratively train a VFL model $\Theta = (\theta_1, \cdots, \theta_K, \omega)$ to optimize:

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

in which passive party P_k owns features $x_k = (x_{k,1}, \cdots, x_{k,n})$ and the passive model G_{θ_k}, the active party owns the labels $y = \{y_1, \cdots, y_m\}$ and active model F_ω. Before training, parties use Private Set Intersection (PSI) to align data records by same ID. During training, each passive party k computes a forward embedding H_k from its features and sends it to the active party. The active party concatenates embeddings $\{H_k\}_{k=1}^K$ into H, generates outputs, computes the loss, and updates its model using $\frac{\partial L}{\partial \omega}$. It then sends $\frac{\partial \ell}{\partial H_k}$ to passive parties, which compute $\frac{\partial \ell}{\partial \theta_k} = \frac{\partial \ell}{\partial H_k} \cdot \frac{\partial H_k}{\partial \theta_k}$ to update their models. Please refer to Appendix A.2 for table of notations.

Label unlearning in VFL. When the active party requests to unlearn some sensitive labels y^u, where the corresponding unlearn feature is $\{x_k^u\}_{k=1}^K := \{\{x_{k,i}^u\}_{i=1}^{n_u}\}_{k=1}^K$. The active party aims to remove the influence of y^u on both the active model F_ω and K passive models $\{G_{\theta_k}\}_{k=1}^K$.

Label unlearning in VFL refers to the process of efficiently and securely removing label information from a VFL system. Specifically, the unlearned passive model of passive party k, denoted as θ_k^u, and the unlearned active model, denoted as ω^u, are obtained through the application of an unlearning mechanism $\mathcal{U}$, as follows:

$$\theta_k^u = \mathcal{U}(\theta_k, y^u), \quad \omega^u = \mathcal{U}(\omega, y^u),$$

where θ_k and ω represent the passive model of passive party k and active model before unlearning, respectively, and $\mathbf{g}^u$ are the gradients associated with the unlearned label y^u.

Building on (Bourtoule et al., 2021), label unlearning in VFL should meet three goals: i) **Privacy-Preserving Unlearning**: remove the impact of specific labels without affecting other data; ii) **Enhanced Unlearning Effectiveness**: achieve this with minimal data; and iii) **Computational Efficiency**: avoid full model retraining.

Threat Model. We assume all participating parties are *semi-honest* and do not collude with each other. An adversary (i.e., the passive party) faithfully executes the training protocol but may launch privacy attacks to infer the private labels of the active party.

Assumption. We assume that the passive party possesses corresponding labels for a limited number of features, defined as $\mathcal{D}^p = \{(\mathbf{x}_k^p, \mathbf{y}^p)\}_{k=1}^K = \{\{(x_{k,i}^p, y_i)\}_{i=1}^{n_p}\}_{k=1}^K$, where $n_p << n_u$. This assumption is widely employed in prior works (Fu et al., 2022b; Gu et al., 2023; Zou et al., 2022). Moreover, $\mathcal{D}^p$ is consisted of the data with the unlearned label $\mathcal{D}^{p,u}$ and data with other labels $\mathcal{D}^{p,r}$, i.e., $\mathcal{D}^p = \mathcal{D}^{p,u} \cup \mathcal{D}^{p,r}$.

3.2 Label leakage during unlearning

Unlearning the influence of passive models $\{G_{\theta_k}\}_{k=1}^K$ poses the risk of leaking labels $y^u = \{y_1^u, \ldots, y_{m_u}^u\}$ to passive parties. This arises because the active party must transmit information (e.g., gradients $\mathbf{g}^u = g_1^u, \ldots, g_{n_u}^u$) during the unlearning process, which may enable label inference. Specifically, when unlearning a single label y_1^u, we consider two methods: (i) retraining (Foster et al., 2024a) and (ii) boundary unlearning (Chen et al., 2023). In retraining, the active party must indicate which features to exclude, revealing the label. In boundary unlearning, gradients linked to y_1^u may also expose the label.

Datasets	Membership Leakage Rate (%)	
	Standard Retraining	Ours
CIFAR10	100	14.38
CIFAR100	100	4.04

Table 1: The averaged membership leakage rate evaluated across all class for CIFAR10/100 with ResNet18 (lower better). We set k = 5000 for CIFAR10 and k = 500 for CIFAR100, where k is the total number of samples in each class.

As shown in Tab. 1, retraining inherently reveals the full set of deleted labels to the passive party, resulting in 100% leakage of sensitive information. Moreover, when unlearning multiple labels (m_u), label leakage worsens as the passive party can infer on the transferred gradient by the active party (please refer to Appendix A.3 for detailed experimental results). In contrast, our method which will be detailed next departs from this by (a) Restricting information exchange to embeddings and gradient updates of a small public subset, which provably (Appendix A.1, Bernstein inequality) bounds the residual influence of unlearned data, and (b) Demonstrating empirically that membership/feature linkage leakage is reduced from 100% (retraining) to as low as 14.38%. This demonstrates that the passive party could not reconstruct the full feature–label correspondence of deleted data.

4 The Proposed few-shot label unlearning method

This section outlines the proposed few-shot label unlearning method, as shown in Fig. 1 and Algorithm 1. The approach involves three steps: (1) augmenting the forward embedding with manifold mixup to address limited labeled data (Sect. 4.1); (2) using gradient ascent on the augmented embedding to guide label removal in both passive and active models (Sect. 4.2); and (3) improving model accuracy on the retained labels (Sect. 4.3).

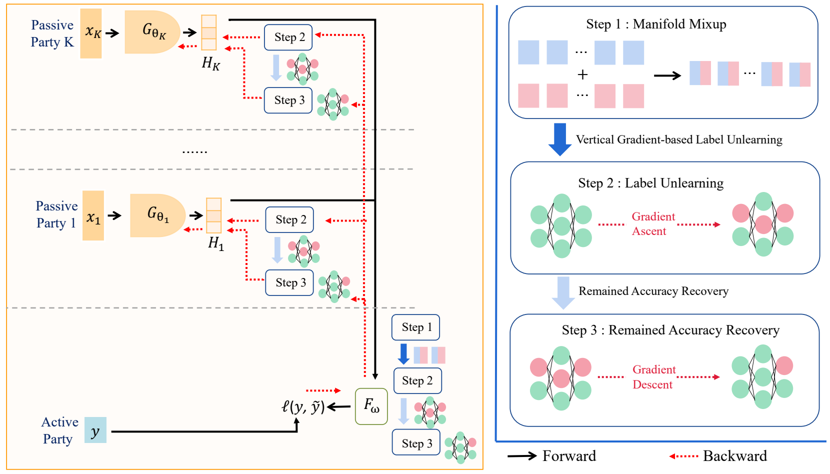


Figure 1: Overview of our proposed few-shot unlearning framework in VFL setting.

4.1 Vertical Manifold Mixup

Due to the label privacy leakage issue (Sect. 3.2), directly applying traditional machine unlearning methods will pose some challenges. We assume that the active party discloses a limited number of labels to the passive party to facilitate the unlearning of a specific label. However, this small labeled dataset with unlearned label, denoted as $\mathcal{D}^{p,u} = \{(x_{1,i}^{p,u}, y_i^{p,u}), \cdots, (x_{K,i}^{p,u}, y_i^{p,u})\}_{i=1}^{n_{p,u}}$, is insufficient for an effective unlearning (see Fig 5). Consequently, this scenario can be framed as a few-shot unlearning problem, wherein a minimal set of labels is employed to unlearn all associated labels. Please refer to Appendix A.4 for an explanation of why a small number of samples (up to 40 per label, as shown in Tab. 8 and 9) is sufficient to achieve performance comparable to using the full dataset.

Drawing inspiration from the few-shot learning principles, we re-purpose the manifold mixup mechanism (Verma et al., 2019) by interpolating hidden embeddings rather than directly mixing the features. We propose a manifold mixup framework for VFL by optimizing the following loss function:

$$\max_{\omega,\theta_1,\cdots,\theta_K} \frac{1}{{n_{p,u}}^2} \sum_{i,j=1}^{n_{p,u}} \ell(F_\omega \circ (\text{Mix}_\lambda(G_{\theta_1}(x_{1,i}^{p,u}), G_{\theta_1}(x_{1,j}^{p,u})),$$
$$\cdots, \text{Mix}_\lambda(G_{\theta_K}(x_{K,i}^{p,u}), G_{\theta_K}(x_{K,j}^{p,u})), \text{Mix}_\lambda(y_i^{p,u}, y_j^{p,u})),$$

where

$$\text{Mix}_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b. \tag{2}$$

The mixed coefficient λ ranges from 0 to 1. The advantage of the manifold mixup approach lies in its ability to flatten the state distributions (Verma et al., 2019). Once each passive party k generates its local embeddings $\{H_k^{p,u} = G_\theta(x_k^{p,u})\}$, these embeddings are transmitted to the active party. The active party then constructs a set of synthesized embeddings $\vec{H}_k^u$ by performing manifold mixup among embeddings originating from the same passive party, ensuring consistency in the λ value while avoiding any coordination among passive parties. Similarly, for the small data with the remaining label $\mathcal{D}^{p,r}$, we also implement the manifold mixup to obtain the $\vec{H}_k^r$ and corresponding label $\vec{y}^r$ for the remaining accuracy recovery step.

4.2 Vertical Gradient-based Label Unlearning

Once the augmented embeddings $\{\vec{H}_1, \dots, \vec{H}_K^u\}$ are generated, a straightforward yet effective strategy is to implement gradient ascent for both the active and passive models using these augmented embeddings. Specifically, the active party concatenates all embeddings $\{\vec{H}_k^u\}_{k=1}^K$ into a single tensor $\vec{H}^u = [\vec{H}_1^u, \dots, \vec{H}_K^u]$, and optimizes it according to the following formulation:

$$\min_\omega \ell(F_\omega(\vec{H}^u), \vec{y}^u) = \ell(F_\omega([\vec{H}_1, \dots, \vec{H}_K^u]), \vec{y}^u), \tag{3}$$

where $\vec{y}^u$ represents the mixture of the representative unlearned labels and η is the learning rate.

Unlearning for active model F_ω. On one hand, the active model undergoes unlearning for active model F_ω via gradient ascent as follows:

$$\omega = \omega + \eta \nabla_\omega \ell(F_\omega(\vec{H}^u), \vec{y}^u). \tag{4}$$

Unlearning for passive model G_{ω_k}. Subsequently, the active party computes the gradients $\frac{\partial \ell}{\partial \vec{H}_k^u}$ in accordance with equation 3 and transmits these gradients to the corresponding passive party k. Finally, the passive party k updates the passive model G_{θ_k} using the following expression:

$$\theta_k = \theta_k + \eta \nabla_{\vec{H}^u} \ell(F_\omega(\vec{H}^u), \vec{y}^u) \cdot \nabla_{\theta_k} \vec{H}_k^u. \tag{5}$$

Theorem 1. *Suppose that both the trained passive model θ and the active model ω achieve a training loss bounded by a small value ϵ. Then, when unlearning a single label, the following holds:*

$$\begin{aligned} \mathbb{E}_{(\vec{H}^u, \vec{y}^u)} \nabla_\omega \ell(\omega; \vec{H}^u, \vec{y}^u) \cdot \mathbb{E}_{(H^u, y^u)} \nabla_\omega \ell(\omega; H^u, y^u) > 0, \\ \mathbb{E}_{(\vec{H}^u, \vec{y}^u)} \nabla_\theta \ell(\theta; \vec{H}^u, \vec{y}^u) \cdot \mathbb{E}_{(H^u, y^u)} \nabla_\theta \ell(\theta; H^u, y^u) > 0, \end{aligned} \tag{6}$$

where $(\vec{H}^u, \vec{y}^u)$ denotes the manifold mixup embeddings and labels of the public data $\mathcal{D}^{p,u}$ associated with the unlearned label, (H^u, y^u) denotes the embeddings and labels of the complete unlearned dataset $\mathcal{D}^u$., and ℓ is the main task loss of VFL.

Theorem 1 indicates that the gradient update direction for label unlearning, when applied to the augmented embeddings of the public unlearned data, is positively aligned with the update direction derived from the embeddings of the entire unlearned dataset. This result suggests that gradient-based label unlearning using only public data is effective and approximates the behavior of unlearning with access to all unlearned data. See proof in Appendix A.1.

Algorithm 1 Our proposed unlearning

Input: Bottom models parameters θ_k of K passive parties, top model parameters ω , unlearn data $\mathcal{D}^{p,u}$, remain data $\mathcal{D}^{p,r}$, learning rate η, unlearn epoch N, batch size b.
Output: Unlearned bottom models parameters θ_k^u, unlearned top model parameters ω^u
for n in N **do**
    for $k = 1$ to K **do**
        Each passive party k generates local embeddings and send it to *Active party*.
    end for
    ▷ *Manifold Mixup*
    Active party generate $\vec{H}_k^u$ and $\vec{H}_k^r$ according to Eq. equation 2.
    ▷ *Gradient-based Label Unlearning*
    Active party updates ω according to Eq. equation 4
    Active party transfers $\frac{\partial \ell}{\partial \vec{H}_k^u}$ to all passive parties.
    for $k = 1$ to K **do**
        Each Passive party k update θ_k as Eq. equation 5.
    end for
    ▷ *Remained Accuracy Recovery*
    Active party updates ω according to Eq. equation 7
    Active party transfers $\frac{\partial \ell}{\partial \vec{H}_k^u}$ to all passive parties.
    for $k = 1$ to K **do**
        Each Passive party k update θ_k as Eq. equation 7.
    end for
end for
Return unlearned passive model $\theta_k \triangleq \theta_k^u$ and unlearned active model $\omega \triangleq \omega^u$.

4.3 Remained Accuracy Recovery

The preceding step focuses solely on unlearning the target labels and does not explicitly account for the model's accuracy on the retained data. As a result, the predictive performance on the remaining dataset may degrade. To address this issue, we introduce a Remained Accuracy Recovery step aimed at enhancing model accuracy on the retained samples. Specifically, using the small dataset with retained labels, denoted as $\mathcal{D}^{p,r}$, we jointly optimize the passive and active models with respect to the main task loss, formulated as follows:

$$\begin{aligned} \omega &= \omega - \eta \nabla_\omega \ell(F_\omega(\vec{H}^r), \vec{y}^r), \\ \theta_k &= \theta_k - \eta \nabla_{\vec{H}^r} \ell(F_\omega(\vec{H}^r), \vec{y}^r) \cdot \nabla_{\theta_k} \vec{H}_k^r. \end{aligned} \tag{7}$$

5 Experimental Results

This section presents the empirical analysis of the proposed method in terms of utility, unlearning effectiveness, time efficiency and some ablation studies.

Model	Datasets	Metrics	Accuracy (%)								
			Baseline	Retrain	FT	Fisher	Amnesiac	Unsir	BU	SSD	Ours
ResNet18	MNIST	$\mathcal{D}^r$	99.29	99.33 ± 0.03	98.99 ± 0.05	12.16 ± 0.46	98.16 ± 0.92	84.92 ± 1.13	98.72 ± 0.02	96.50 ± 0.19	**99.05 ± 0.01**
		y^u	99.39	0.00 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	58.83 ± 1.79	**0.00 ± 0.00**	**0.00 ± 0.00**
		ASR	90.61	1.03 ± 0.24	2.92 ± 1.08	**0.11 ± 0.07**	**0.00 ± 0.00**	29.07 ± 7.95	**0.47 ± 0.01**	**0.00 ± 0.00**	**0.35 ± 0.01**
	CIFAR10	$\mathcal{D}^r$	90.61	91.26 ± 0.12	88.16 ± 0.15	54.4 ± 10.77	86.37 ± 0.20	75.02 ± 1.65	72.65 ± 0.55	87.17 ± 0.76	**89.29 ± 0.19**
		y^u	93.10	0.00 ± 0.00	11.00 ± 0.10	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	3.25 ± 0.15	**0.00 ± 0.00**	**0.00 ± 0.00**
		ASR	83.84	25.98 ± 1.27	**15.85 ± 2.33**	50.67 ± 12.51	**1.62 ± 0.54**	76.78 ± 0.44	34.90 ± 1.16	**25.74 ± 2.78**	**17.23 ± 1.00**
	CIFAR100	$\mathcal{D}^r$	71.43	71.03 ± 0.12	66.86 ± 0.73	61.04 ± 8.61	60.05 ± 0.03	59.32 ± 0.14	55.30 ± 0.81	67.25 ± 0.91	**69.96 ± 0.12**
		y^u	83.00	0.00 ± 0.00	12.25 ± 2.25	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	3.50 ± 0.50	**0.00 ± 0.00**	**0.00 ± 0.00**
		ASR	88.40	25.53 ± 3.36	29.30 ± 2.70	28.10 ± 4.10	**2.60 ± 1.30**	73.70 ± 1.70	**6.00 ± 0.60**	**6.80 ± 0.01**	**5.02 ± 1.01**
	ModelNet	$\mathcal{D}^r$	94.26	93.90 ± 0.11	66.64 ± 1.53	28.10 ± 0.69	73.91 ± 1.83	13.51 ± 0.05	24.07 ± 0.27	81.89 ± 0.98	**87.69 ± 0.21**
		y^u	100.00	0.00 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	4.79 ± 1.02	2.00 ± 0.00
		ASR	98.40	0.65 ± 0.05	0.79 ± 0.16	23.48 ± 0.77	1.11 ± 0.16	49.20 ± 1.25	21.16 ± 0.23	0.77 ± 0.03	**0.11 ± 0.03**
	Brain MRI	$\mathcal{D}^r$	97.46	98.81 ± 0.34	81.89 ± 0.82	30.26 ± 0.21	78.29 ± 0.09	73.29 ± 0.09	64.19 ± 0.91	85.93 ± 0.09	**93.79 ± 0.17**
		y^u	97.29	0.00 ± 0.00	4.33 ± 0.49	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	3.67 ± 0.14	1.00 ± 0.92	0.56 ± 0.18
		ASR	82.32	48.37 ± 1.31	**24.07 ± 0.16**	51.77 ± 0.83	**24.29 ± 0.91**	**27.32 ± 3.04**	58.36 ± 0.92	52.35 ± 0.76	**45.71 ± 0.99**
	COVID-19 Radiography	$\mathcal{D}^r$	92.82	93.85 ± 0.12	73.84 ± 0.73	40.98 ± 4.89	81.76 ± 0.77	72.30 ± 0.37	66.13 ± 0.69	81.11 ± 0.15	**92.35 ± 0.77**
		y^u	73.33	0.00 ± 0.00	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	10.83 ± 1.48	6.11 ± 1.92	**0.00 ± 0.00**	**0.00 ± 0.00**
		ASR	88.97	46.67 ± 2.74	**20.51 ± 2.28**	49.27 ± 0.99	**13.16 ± 3.92**	**37.52 ± 5.32**	**18.12 ± 2.63**	**36.17 ± 0.02**	**39.21 ± 0.11**
Vgg16	CIFAR10	$\mathcal{D}^r$	89.50	90.27 ± 0.19	88.69 ± 0.08	15.93 ± 4.82	84.67 ± 0.22	74.74 ± 0.72	82.69 ± 0.1	87.19 ± 0.39	**88.97 ± 0.18**
		y^u	91.10	0.00 ± 0.00	4.25 ± 1.05	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	2.85 ± 0.05	**0.00 ± 0.00**	0.13 ± 0.04
		ASR	81.66	33.10 ± 1.86	**21.84 ± 2.66**	42.25 ± 6.23	**2.36 ± 0.86**	**21.75 ± 2.41**	34.53 ± 0.65	**29.19 ± 0.01**	**31.33 ± 0.36**
	CIFAR100	$\mathcal{D}^r$	65.48	65.32 ± 0.32	59.92 ± 0.56	35.42 ± 1.95	55.83 ± 0.13	55.78 ± 0.59	52.21 ± 0.00	64.11 ± 0.73	**64.24 ± 0.09**
		y^u	77.00	0.00 ± 0.00	2.50 ± 0.25	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	3.00 ± 0.00	1.19 ± 0.03	1.00 ± 0.00
		ASR	87.20	42.13 ± 2.73	**34.50 ± 4.30**	**40.70 ± 3.50**	**3.10 ± 0.15**	42.70 ± 0.70	**18.20 ± 0.11**	**39.11 ± 0.91**	**20.28 ± 0.67**

Table 2: Accuracy of $\mathcal{D}^r$, y^u and ASR for each unlearning method across ResNet18 and Vgg16 models in single-label unlearning.

5.1 Experiment Setup

Datasets & Models. We conduct experiments on seven datasets: MNIST (Lecun et al., 1998), CIFAR10, CIFAR100 (Krizhevsky et al., 2009), ModelNet (Wu et al., 2015), Brain Tumor MRI (Wang et al., 2024), COVID-19 Radiography (Rahman, 2022) and Yahoo Answers dataset (Fu et al., 2022a). We adopt ResNet18 (He et al., 2016) on the datasets MNIST, CIFAR10, CIFAR100, ModelNet, Brain Tumor MRI and COVID-19 Radiography. We adopt MixText (Chen et al., 2020) on the Yahoo Answers dataset. We do extend our experiments with Vgg16 (Simonyan & Zisserman, 2015) on the dataset CIFAR10 and CIFAR100. Experiments are repeated over five random trials, and results are reported as mean and standard deviation. We conduct experiments on a single NVIDIA A100 GPU.

VFL Setting & Unlearning Scenarios. We simulate a VFL setting with one active model owner and 1–8 passive model owners. In *single-label unlearning*, one label is removed from all datasets; in *two-label unlearning*, two labels are removed from CIFAR10/100; and in *multi-label unlearning*, four labels are removed from CIFAR100. Appendix A.9 summarizes model names, VFL configurations, datasets, unlearned labels and hyper-parameters used for unlearning.

Evaluations Metrics. We evaluate the utility of unlearning by measuring the accuracy of remaining data, $\mathcal{D}^r$ before and after unlearning. The higher accuracy on $\mathcal{D}^r$ indicates stronger utility. Stronger utility indicates that more information of $\mathcal{D}^r$ is being preserved.

To assess unlearning effectiveness, we use a basic Membership Inference Attack (MIA) from (Shokri et al., 2017) to measure the Attack Success Rate (ASR) and the prediction accuracy on the unlearned label y^u before and after unlearning. MIA determines whether a data point was part of the model's training. Lower accuracy on y^u suggests more effective unlearning. A 0% ASR may indicate the Streisand effect (Golatkar et al., 2020a), where the model consistently mispredicts all y^u samples as a single incorrect label. Ideally, the ASR should be slightly lower than that of a retrained model, signaling successful unlearning without revealing extra information.

Time efficiency is measured by the runtime of each method: the shorter, the better. An effective unlearning method should: a) preserve $\mathcal{D}^r$ accuracy; b) reduce y^u accuracy to near 0%; c) achieve an ASR slightly below that of a retrained model; and d) run quickly.

Baselines. We compare our method with the following baselines: Retrain, Fine-Tuning (Golatkar et al., 2020a; Jia et al., 2023), Fisher Forgetting (Golatkar et al., 2020a), Amnesiac Unlearning

(Graves et al., 2021), UNSIR (Tarun et al., 2024), Boundary Unlearning (Chen et al., 2023) and SSD (Foster et al., 2024b). The implementation details of each baseline are in Appendix A.9.

5.2 Experimental Results

5.2.1 Utility Guarantee

To assess the utility of our proposed method, we evaluate accuracy on $\mathcal{D}^r$ before and after unlearning (Tab. 2). An effective unlearning method should retain as much information as possible from $\mathcal{D}^r$.

From Tab. 2, we observe: i) Fine-tuning preserves $\mathcal{D}^r$ well but has low unlearning effectiveness (see Sect. 5.2.2); ii) Fisher forgetting severely degrades $\mathcal{D}^r$ accuracy; iii) Amnesiac's random mislabeling shifts decision boundaries, hurting $\mathcal{D}^r$, especially in label-rich datasets like CIFAR100 and ModelNet; iv) UNSIR's repair step fails to fully retain $\mathcal{D}^r$, causing some degradation; v) Boundary unlearning yields inconsistent results across datasets and models; vi) While SSD preserves $\mathcal{D}^r$ reasonably well, it underperforms slightly compared to our approach; vii) In contrast, our method consistently achieves strong unlearning and retention performance.

5.2.2 Unlearning Effectiveness

For unlearning effectiveness, we run MIA to evaluate if the unlearned model leaks any information about the y^u and measure the accuracy of y^u before and after unlearning.

From Tab. 2, we observe: i) Fine-tuning performs poorly on CIFAR10/100; ii) Fisher forgetting, Amnesiac, and UNSIR effectively reduce y^u accuracy to 0.00%; iii) Boundary unlearning is inconsistent across datasets and models, with mixed results; iv) SSD shows strong unlearning effectiveness across all settings; v) In contrast, our method consistently achieves effective unlearning across all settings.

From Tab. 2, we further observe: i) Fine-tuning yields consistent ASR; ii) Fisher forgetting often shows high ASR; iii) Amnesiac consistently has low ASR; iv) UNSIR shows high ASR in most cases; v) Boundary unlearning has relatively stable ASR; vi) SSD achieves strong ASR performance across most datasets, with the exception of ModelNet and Brain MRI; and vii) our method consistently achieves strong ASR across all settings.

5.2.3 Time Efficiency

Figure 2 shows single-label unlearning times (in seconds) on CIFAR10 with ResNet18 across varying numbers of passive parties: i) Methods using the full dataset or $\mathcal{D}^r$ (e.g., Fine-Tuning, Amnesiac, Fisher Forgetting) have high execution time; ii) Methods using only $\mathcal{D}_u$, like Boundary Unlearning, are faster; iii) Our solution has the lowest execution time (16x - 1200x lower).

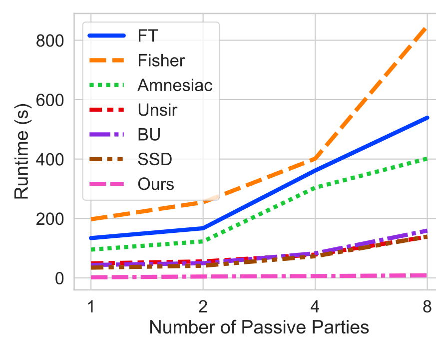


Figure 2: The runtime(s) of each unlearning method.

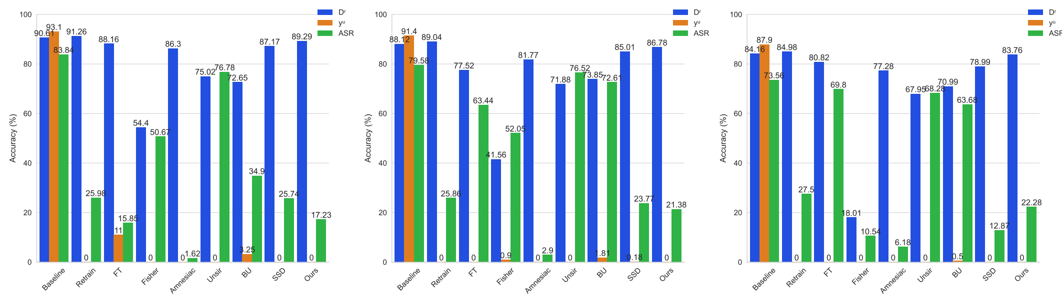
In addition, the manifold mixup step is executed on the embeddings of the same passive party (see Fig. 1 and Algo. 1 of the Sect. 4). As a result, the unlearning time increases linearly with the number of passive parties. The unlearning times of different methods are compared for varying numbers of passive parties in Fig. 2, demonstrating that our method remains the most efficient compared to the alternatives.

5.2.4 Other Modality

Beyond image datasets, our method also maintains strong performance on text datasets. This demonstrates its robustness across modalities, which is particularly relevant since VFL is widely adopted in e-commerce scenarios where text is the primary modality.

Metrics	Accuracy (%)		
	Baseline	Retrain	Ours
$\mathcal{D}^r$	62.92	63.14 ± 0.45	61.01 ± 0.91
y^u	41.63	0.00 ± 0.00	1.41 ± 0.35

Table 3: Single-label unlearning scenario on Yahoo Answer dataset with MixText architecture.



(a) Two Passive Parties (b) Four Passive Parties (c) Eight Passive Parties

Figure 3: Accuracy of $\mathcal{D}^r$, y^u and ASR for each unlearning method across ResNet18 model in single-label unlearning on different number of passive parties.

Table 3 shows the effectiveness of our method in preserving the accuracy of the retained data ($\mathcal{D}^r$) while substantially reducing the accuracy of the unlearned data (y^u). For example, the accuracy of the unlearned data drops sharply from 41.63% to 1.41%, whereas the accuracy of the retained data decreases by less than 2%. Other baseline methods are excluded from direct comparison as they are designed primarily for image datasets and do not generalize well to text-based tasks.

5.3 Ablation Study

In this section, we conduct an ablation study on the effectiveness of different sizes of $\mathcal{D}^{p,u}$, our method across varying numbers of passive parties and different privacy-preserving VFL mechanisms.

Module-Omission Experiments: We conduct module-omission experiments to quantify the contribution of individual components to overall performance. We evaluate four experimental variants: (i) standard gradient ascent using all data samples, (ii) few-shot gradient ascent under the same data samples as our method, (iii) few-shot gradient ascent with mixup, and (iv) our proposed approach. Please refer to Appendix A.5 for complete results.

Evaluation for different numbers of passive parties: The robustness of our approach is unaffected by the number of passive parties. Experiments with varying participant counts (see Fig. 3) show consistent performance across all settings, demonstrating that scalability is preserved.

Evaluation for different privacy preserving VFL methods: Our method remains effective when applied to privacy-preserving VFL models. To validate this, we evaluate performance under Differential Privacy and Gradient Compression, two mechanisms that are widely adopted in VFL training (Fu et al., 2022b). The results confirm that our approach maintains robustness, underscoring its practicality for real-world deployments. Please refer to Appendix A.6 for a complete result.

Multi-label Unlearning: Similarly, our method remains highly effective and maintains strong utility even under more challenging conditions (*i.e.* unlearning multiple labels), underscoring its reliability across diverse scenarios. Please refer to Appendix A.7 for complete results.

6 Conclusion

This paper presents a pioneering approach to label unlearning within the VFL domain, addressing a critical gap in the existing literature. By introducing a few-shot unlearning method that leverages manifold mixup, we effectively mitigate the risk of label privacy leakage while ensuring efficient unlearning from both active and passive models. Our systematic exploration of potential label privacy risks and extensive experimental validation on benchmark datasets underscore the proposed method's efficacy and rapid performance. Ultimately, this work not only advances the understanding of unlearning in VFL but also sets the stage for further innovations in privacy-preserving collaborative machine learning practices.

References

Ali Abbasi, Chayne Thrash, Elaheh Akbari, Daniel Zhang, and Soheil Kolouri. Covarnav: Machine unlearning via model inversion and covariance navigation. *arXiv preprint arXiv:2311.12999*, 2023.

Manaar Alam, Hithem Lamri, and Michail Maniatakos. Get rid of your trail: Remotely erasing backdoors in federated learning. *IEEE Transactions on Artificial Intelligence*, 2024.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pp. 141–159. IEEE, 2021.

Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1366–1383, 2023a. doi: 10.1109/SP46215.2023.10179336.

Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1366–1383. IEEE, 2023b.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015. doi: 10.1109/SP.2015.35.

Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2147–2157, 2020.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21. ACM, November 2021. doi: 10.1145/3460120.3484756. URL `http://dx.doi.org/10.1145/3460120.3484756`.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7766–7775, 2023.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Dasol Choi and Dongbin Na. Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems. *arXiv preprint arXiv:2311.02240*, 2023.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023a. doi: 10.1109/TIFS.2023.3265506.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7210–7217, 2023b.

Zihao Deng, Zhaoyang Han, Chuan Ma, Ming Ding, Long Yuan, Chunpeng Ge, and Zhe Liu. Vertical federated unlearning on the logistic regression model. *Electronics*, 12(14), 2023. ISSN 2079-9292. doi: 10.3390/electronics12143182. URL `https://www.mdpi.com/2079-9292/12/14/3182`.

Akash Dhasade, Yaohong Ding, Song Guo, Anne-marie Kermarrec, Martijn De Vos, and Leijie Wu. Quickdrop: Efficient federated unlearning by integrated dataset distillation. *arXiv preprint arXiv:2311.15603*, 2023.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024a.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12043–12051, 2024b.

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)*, pp. 1397–1414, 2022a. ISBN 978-1-939133-31-1.

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)*, pp. 1397–1414, 2022b.

Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 2024.

Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 373–402. Springer, 2020.

Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640*, 2022.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9304–9312, 2020a.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *16th European Conference on Computer Vision (ECCV)*, pp. 383–398, 2020b.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 792–801, 2021.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021.

Hanlin Gu, Jiahuan Luo, Yan Kang, Lixin Fan, and Qiang Yang. Fedpass: privacy-preserving vertical federated deep learning with adaptive obfuscation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 3759–3767, 2023.

Hanlin Gu, WinKent Ong, Chee Seng Chan, and Lixin Fan. Ferrari: Federated feature unlearning via optimizing feature sensitivity. *Advances in Neural Information Processing Systems*, 37:24150–24180, 2024a.

Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. Unlearning during learning: an efficient federated machine unlearning method. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4035–4043, 2024b.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3832–3842, 2020.

Anisa Halimi, Swanand Ravindra Kadhe, Ambrish Rawat, and Nathalie Baracaldo Angel. Federated unlearning: How to efficiently erase a client in fl? In *International Conference on Machine Learning*, 2022.

Mengde Han, Tianqing Zhu, Lefeng Zhang, Huan Huo, and Wanlei Zhou. Vertical federated unlearning via backdoor certification. *IEEE Transactions on Services Computing*, 2025.

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*, pp. 770–778, 2016.

Ying He, Zhili Shen, Jingyu Hua, Qixuan Dong, Jiacheng Niu, Wei Tong, Xu Huang, Chen Li, and Sheng Zhong. Backdoor attack against split neural network-based vertical federated learning. *IEEE Transactions on Information Forensics and Security*, 19:748–763, 2024. doi: 10.1109/TIFS.2023.3327853.

Zecheng He, Tianwei Zhang, and Ruby B. Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pp. 148–162. Association for Computing Machinery, 2019. ISBN 9781450376280. doi: 10.1145/3359789.3359824. URL `https://doi.org/10.1145/3359789.3359824`.

Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4819–4828, 2024.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. *arXiv preprint arXiv:2101.04898*, 2021.

Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36:51584–51605, 2023.

Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. *Advances in neural information processing systems*, 34:994–1006, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987, 2023.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Bowen Li, Jian Zhang, Jie Li, and Chentao Wu. Securecut: Federated gradient boosting decision trees with efficient machine unlearning. In *International Conference on Pattern Recognition*, pp. 350–365. Springer, 2024.

Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. Subspace based federated unlearning. *arXiv preprint arXiv:2302.12448*, 2023.

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

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. *arXiv preprint arXiv:2102.08504*, 2021.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–10, 2021. doi: 10.1109/IWQOS52092.2021.9521274.

Junlin Liu, Xinchen Lyu, Qimei Cui, and Xiaofeng Tao. Similarity-based label inference attack against training and inference of split learning. *IEEE Transactions on Information Forensics and Security*, 19:2881–2895, 2024a. doi: 10.1109/TIFS.2024.3356821.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024b. doi: 10.1109/TKDE.2024.3352628.

Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. IEEE, May 2022. doi: 10.1109/infocom48880.2022.9796721. URL http://dx.doi.org/10.1109/INFOCOM48880.2022.9796721.

Yige Liu, Che Wang, Yiwei Lou, Yongzhi Cao, and Hanpin Wang. Attackers are not the same! unveiling the impact of feature distribution on label inference attacks. *IEEE Transactions on Information Forensics and Security*, 20:71–86, 2025. doi: 10.1109/TIFS.2024.3498464.

Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024c.

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)*, pp. 181–192, 2021. doi: 10.1109/ICDE51399.2021.00023.

Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2218–2227, 2020.

Pengyu Qiu, Xuhong Zhang, Shouling Ji, Chong Fu, Xing Yang, and Ting Wang. Hashvfl: Defending against data reconstruction attacks in vertical federated learning. *IEEE Transactions on Information Forensics and Security*, 19:3435–3450, 2024. doi: 10.1109/TIFS.2024.3356164.

Tawsifur Rahman. Covid-19 radiography database, Mar 2022. URL https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database.

Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2024. ISSN 2162-2388. doi: 10.1109/tnnls.2024.3478334. URL http://dx.doi.org/10.1109/TNNLS.2024.3478334.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pp. 1–10, 2023. doi: 10.1109/INFOCOM53939.2023.10229075.

Ayush K. Tarun, Vikram S. Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2024. ISSN 2162-2388. doi: 10.1109/tnnls.2023.3266233. URL http://dx.doi.org/10.1109/TNNLS.2023.3266233.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pp. 6438–6447. PMLR, 2019.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, pp. 622–632, 2022.

Pengfei Wang, Zhaohong Yan, Mohammad S. Obaidat, Zhiwei Yuan, Leyou Yang, Junxiang Zhang, Zongzheng Wei, and Qiang Zhang. Edge caching with federated unlearning for low-latency v2x communications. *IEEE Communications Magazine*, pp. 1–7, 2023a. doi: 10.1109/MCOM.001.2300272.

Zhousheng Wang, Geng Yang, Hua Dai, and Chunming Rong. Privacy-preserving split learning for large-scaled vision pre-training. *IEEE Transactions on Information Forensics and Security*, 18: 1539–1553, 2023b. doi: 10.1109/TIFS.2023.3243490.

Zichen Wang, Xiangshan Gao, Cong Wang, Peng Cheng, and Jiming Chen. Efficient vertical federated unlearning via fast retraining. *ACM Transactions on Internet Technology*, 24(2):1–22, 2024.

Chen Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. *arXiv preprint arXiv:2201.09441*, 2022.

Xiaodong Wu, Henry Yuan, Xiangman Li, Jianbing Ni, and Rongxing Lu. Evaluating security and robustness for split federated learning against poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 20:175–190, 2025. doi: 10.1109/TIFS.2024.3490861.

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*, pp. 1912–1920, 2015.

Hui Xia, Shuo Xu, Jiaming Pei, Rui Zhang, Zhi Yu, Weitao Zou, Lukun Wang, and Chao Liu. Fedme2: Memory evaluation & erase promoting federated unlearning in dtmn. *IEEE Journal on Selected Areas in Communications*, 41(11):3573–3588, 2023. doi: 10.1109/JSAC.2023.3310049.

Chulin Xie, Pin-Yu Chen, Qinbin Li, Arash Nourian, Ce Zhang, and Bo Li. Improving privacy-preserving vertical federated learning by efficient communication with admm. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 443–471. IEEE, 2024.

Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4006–4013. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/556. URL `https://doi.org/10.24963/ijcai.2022/556`. Main Track.

Minmin Yang, Weiheng Chai, Jiyang Wang, and Senem Velipasalar. Simplimix: A simplified manifold mixup for few-shot point cloud classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3668–3677, 2024a.

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.

Ruikang Yang, Jianfeng Ma, Junying Zhang, Saru Kumari, Sachin Kumar, and Joel J. P. C. Rodrigues. Practical feature inference attack in vertical federated learning during prediction in artificial internet of things. *IEEE Internet of Things Journal*, 11(1):5–16, 2024b. doi: 10.1109/JIOT.2023.3275161.

Yunbo Yang, Xiang Chen, Yuhao Pan, Jiachen Shen, Zhenfu Cao, Xiaolei Dong, Xiaoguo Li, Jianfei Sun, Guomin Yang, and Robert Deng. Openvfl: A vertical federated learning framework with stronger privacy-preserving. *IEEE Transactions on Information Forensics and Security*, 19: 9670–9681, 2024c. doi: 10.1109/TIFS.2024.3477924.

Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. Heterogeneous decentralised machine unlearning with seed model distillation. *CAAI Transactions on Intelligence Technology*, 9 (3):608–619, 2024.

Youngsik Yoon, Jinhwan Nam, Hyojeong Yun, Jaeho Lee, Dongwoo Kim, and Jungseul Ok. Few-shot unlearning by model inversion. *arXiv preprint arXiv:2205.15567*, 2022.

Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. Federated unlearning for on-device recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pp. 393–401, 2023.

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18:4732–4746, 2023a. doi: 10.1109/TIFS.2023.3297905.

Xulong Zhang, Jianzong Wang, Ning Cheng, Yifu Sun, Chuanyao Zhang, and Jing Xiao. Machine unlearning methodology based on stochastic teacher network. In *International Conference on Advanced Data Mining and Applications*, pp. 250–261, 2023b.

Yang Zhao, Jiaxi Yang, Yiling Tao, Lixu Wang, Xiaoxiao Li, and Dusit Niyato. A survey of federated unlearning: A taxonomy, challenges and future directions. *arXiv preprint arXiv:2310.19218*, 2023.

Yian Zhao, Pengfei Wang, Heng Qi, Jianguo Huang, Zongzheng Wei, and Qiang Zhang. Federated unlearning with momentum degradation. *IEEE Internet of Things Journal*, 11(5):8860–8870, 2024. doi: 10.1109/JIOT.2023.3321594.

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*, 10(6):1016–1027, 2022.

A APPENDIX

This appendix provides additional theoretical insights, extended experimental results and further implementation details to support the main submission. We begin with a theoretical analysis of unlearning effectiveness in Sect. A.1, followed by a table of notations in Sect. A.2 for reference. Sect. A.3 provides an explanation of label leakage during the unlearning process in VFL, while Sect. A.4 discusses why our method can achieve strong results with only a small number of data samples. Sect. A.5 highlights the importance of each component in our framework through a module ablation study. In addition, Sect. A.6 presents ablation experiments on applying our method to privacy-preserved VFL models. Sect. A.7 presents the experimental results for two-label and multi-label unlearning scenarios, showcasing the effectiveness of our solution in handling more complex unlearning tasks. This section also includes a detailed performance analysis of each baseline across three key metrics: (i) accuracy on $\mathcal{D}^r$, (ii) accuracy on y^u and (iii) ASR score. Sect. A.8 presents a comparative table of our method against existing Federated Unlearning approaches, highlighting the capability of our method to preserve label privacy during the unlearning process. Finally, in Sect. A.9, we provide detailed information about the experiment setup, including model architectures, unlearning scenarios, baseline methods, datasets, and the data distribution among parties.

A.1 THEORETICAL ANALYSIS FOR UNLEARNING EFFECTIVENESS

Theorem 2. *Suppose that the trained passive model θ and active model ω achieve near-zero training loss. Then, when unlearning a single label, the following holds:*

$$\begin{aligned}\mathbb{E}_{(\vec{H}^u,\vec{y}^u)}\nabla_\omega \ell(\omega;\vec{H}^u,\vec{y}^u)\cdot\mathbb{E}_{(H^u,y^u)}\nabla_\omega\ell(\omega;H^u,y^u)>0,\\ \mathbb{E}_{(\vec{H}^u,\vec{y}^u)}\nabla_\theta \ell(\theta;\vec{H}^u,\vec{y}^u)\cdot\mathbb{E}_{(H^u,y^u)}\nabla_\theta\ell(\theta;H^u,y^u)>0,\end{aligned} \tag{8}$$

where $(\vec{H}^u,\vec{y}^u)$ denotes the manifold mixup embeddings and labels of the public data $\mathcal{D}^{p,u}$ associated with the unlearned label, (H^u,y^u) denotes the embeddings and labels of the complete unlearned dataset $\mathcal{D}^u$., and ℓ is the main task loss of VFL.

Proof. In simplified analysis, we set the loss based on the small-size public data $\mathcal{D}^{u,p}$, the manifold mixup of $\mathcal{D}^{u,p}$, and the original unlearned data $\mathcal{D}^u$ are ℓ_1, ℓ_{mix} and ℓ_2 respectively. We consider a two-layer linear neural network: $f(x)=\omega(\theta x)$. The loss function is the Mean Squared Error (MSE): $\ell=\frac{1}{2|D|}\sum_{i\in D}\|f(x_i)-y_i\|^2$.

For a public unlearned sample (x_i^u,y_i^u), let $H_i^u=\theta x_i^u$ be the hidden representation, $z_i=\omega H_i^u$ be the output, $e_i=z_i-y_i^u$ be the error. The gradients are:

$$\nabla_\omega\ell_i=e_i{H_i^u}^\top,\quad \nabla_\theta\ell_i=\omega^\top e_i{x_i^u}^\top. \tag{9}$$

Averaging the gradients over the any dataset $\mathcal{D}^{u,p}$ are:

$$\nabla_\omega\ell=\frac{1}{|D|}\sum_i e_i{H_i^u}^\top,\quad \nabla_\theta\ell=\frac{1}{|D|}\sum_i\omega^\top e_i{x_i^u}^\top \tag{10}$$

Given two samples (x_i^u,y_i^u), (x_j^u,y_j^u), and a mixup coefficient $\lambda\in[0,1]$, we have:

$$\begin{aligned}H_{\text{mix}}=\lambda H_i^u+(1-\lambda)H_j^u,\quad y_{\text{mix}}=\lambda y_i^u+(1-\lambda)y_j^u,\\ z_{\text{mix}}=\omega H_{\text{mix}}=\lambda z_i+(1-\lambda)z_j^u e_{\text{mix}}=z_{\text{mix}}-y_{\text{mix}}=\lambda e_i+(1-\lambda)e_j\end{aligned} \tag{11}$$

Then we have the gradients on ω for manifold mixup:

$$\begin{aligned}\nabla_\omega\ell_{\text{mix}}&=e_{\text{mix}}H_{\text{mix}}^\top\\ &=(\lambda e_i+(1-\lambda)e_j)(\lambda H_i^u+(1-\lambda){H_j^u}^\top)\\ &=\lambda^2 e_i{H_i^u}^\top+\lambda(1-\lambda)(e_i{H_j^u}^\top+e_j{H_i^u}^\top)+(1-\lambda)^2e_j{H_j^u}^\top\end{aligned}$$

The gradients on θ for manifold mixup:

$$\begin{aligned}\nabla_\theta \ell_{\text{mix}} &= \nabla_h \ell_{\text{mix}} \cdot \nabla_\theta H_{\text{mix}} \\ &= \omega^\top e_{\text{mix}}(\lambda x_i^{u\top} + (1-\lambda) x_j^{u\top}) \\ &= \lambda^2 \omega^\top e_i x_i^{u\top} + \lambda(1-\lambda)(\omega^\top e_i x_j^{u\top} + \omega^\top e_j x_i^{u\top}) + (1-\lambda)^2 \omega^\top e_j x_j^{u\top}\end{aligned}$$

Since

$$\mathbb{E}[eh^\top] = \nabla_\omega \ell_1, \quad \mathbb{E}[\omega^\top e x^\top] = \nabla_\theta \ell_1$$

We have:

$$\begin{aligned}\mathbb{E}[\nabla_\omega \ell_{\text{mix}}] = 2\mathbb{E}[\lambda^2]\nabla_\omega \ell_1 + 2(\mathbb{E}[\lambda] - \mathbb{E}[\lambda^2])\mathbb{E}[e]\mathbb{E}[h]^\top \\ \mathbb{E}[\nabla_\theta \ell_{\text{mix}}] = 2\mathbb{E}[\lambda^2]\nabla_\theta \ell_1 + 2(\mathbb{E}[\lambda] - \mathbb{E}[\lambda^2])\omega^\top\mathbb{E}[e]\mathbb{E}[x]^\top\end{aligned}$$

Due to the near-zero training loss, $\mathbb{E}[e]$ tends to be zero, we obtain the averaged gradients of ω and θ on the public dataset $\mathcal{D}^{u,p}$

$$\mathbb{E}[\nabla_\omega \ell_{\text{mix}}] \sim 2\mathbb{E}[\lambda^2]\nabla_\omega \ell_1, \quad \mathbb{E}[\nabla_\theta \ell_{\text{mix}}] \sim 2\mathbb{E}[\lambda^2]\nabla_\theta \ell_1 \tag{12}$$

Since the $\mathcal{D}^{p,u}$ is the subset of the total unlearned dataset $\mathcal{D}^u$, we can leverage the Chebyshev's Inequality to obtain $\nabla_\theta \ell_1 \cdot \nabla_\theta \ell_2 > 0$ and $\nabla_\omega \ell_1 \cdot \nabla_\omega \ell_2 > 0$ with the probability $1 - O(\frac{1}{n_{p,u}})$, where $n_{p,u}$ is the data size of $\mathcal{D}^{p,u}$. Combining Eq.equation 12, we can obtain

$$\mathbb{E}[\nabla_\omega \ell_{\text{mix}}] \cdot \mathbb{E}[\lambda^2]\nabla_\omega \ell_2 > 0, \quad \mathbb{E}[\nabla_\theta \ell_{\text{mix}}] \cdot \mathbb{E}[\lambda^2]\nabla_\theta \ell_2 > 0, \tag{13}$$

which completes the proof. □

Theorem 2 indicates that the gradient update direction for label unlearning, when applied to the augmented embeddings of the public unlearned data, is positively aligned with the update direction derived from the embeddings of the entire unlearned dataset. This result suggests that gradient-based label unlearning using only public data is effective and approximates the behavior of unlearning with access to all unlearned data.

A.2 Table of Notation

Table 4 summarises all notations used throughout the paper for clarity.

Notation	Meaning
F_ω, G_{θ_k}	Active model and k_{th} passive model
w^u, θ_k^u	Unlearned active model and unlearned k_{th} passive model
K	The number of passive party
λ	Mixed coefficient
η	Learning rate
N	Unlearning epochs
$\mathbf{x}_k$	Private features own by k_{th} passive party
y	Private label owned by active party
y^u	The unlearn labels
$\{x_k^u\}$	The unlearned feature for client k corresponding to the y^u
x_k^p	The known features for client k corresponding to the y^u
H_k	Forward embedding of passive party k
$\vec{H}_k^u, \vec{H}_k^r$	Augmented forward embedding of passive party k of unlearn data and remain data.
g_k'	Gradient on the embedding H_k'.

Table 4: Table of Notations

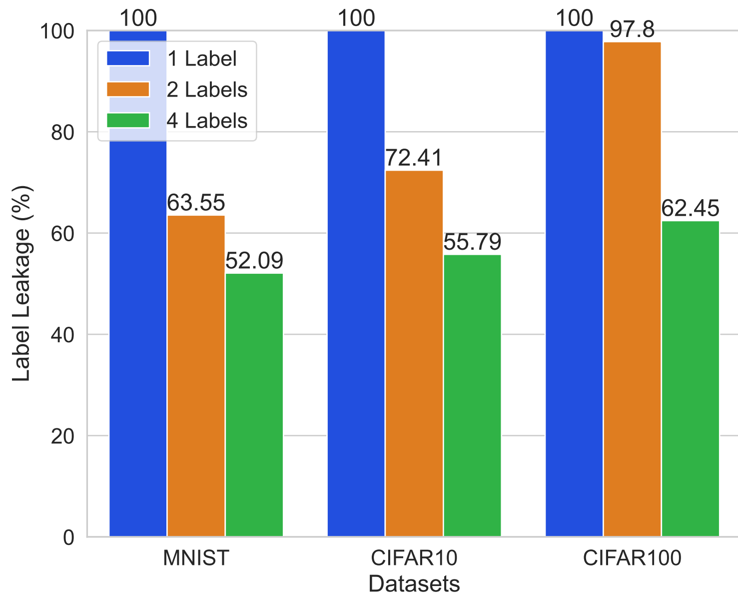


Figure 4: Illustration of label leakage (%) with Boundary unlearning in VFL using ResNet18 model on different number of labels and datasets.

A.3 Label Leakage During Unlearning

Figure 4 shows the label leakage (in %) of Boundary Unlearning in VFL settings for varying numbers of unlearning labels. Single label unlearning shows 100% label leakage since passive parties can immediately infer that all samples involved belong to the same label. Moreover, when unlearning multiple labels (m_u), label leakage worsens as the passive party can infer on the transferred gradient by the active party. Increasing the number of labels, therefore, results in a larger portion of data being exposed during the unlearning process. For instance, with four labels from CIFAR-100, a total of 62.45% of label leakage is exposed. In boundary unlearning, the passive party can infer label information from the gradients $\mathbf{g}^u$ sent by the active party. Specifically, the passive party employs clustering on $\mathbf{g}^u$ to derive m_u clusters by optimizing the following objective function:

$$\min \sum_{g_i \in \mathcal{C}_j} \sum_{j=1}^{m_u} |g_i^u - \bar{g}_j^u|, \tag{14}$$

where $\mathcal{C}_j$ denotes the set of points assigned to cluster j, and $\bar{g}_j^u$ represents the centroid of cluster j. Consequently, the passive party can deduce the labels of the features in $\mathcal{X}$.

A.4 Why Tiny Public Set Suffices

Our goal is *unlearning*, not full re-training. Thus, we only need a small parameter shift to cancel the target label's influence. Below, we quantify three reasons why a much smaller public set $D^{u,p}$ can achieve this.

1. **Variance reduction via manifold mixup.** Manifold mixup augments each public example into an infinite family of virtual points, reducing the gradient-estimator variance proportionally to the number of synthetic mixtures. Recent few-shot studies show that manifold-mixup greatly reduces sample requirements. For example, (Mangla et al., 2020) achieves a higher accuracy than prior baselines on four vision benchmarks using only 5 to 20 samples per class. Similarly, SimpliMix (Yang et al., 2024a) achieves strong gains in few-shot 3-D point-cloud classification.
2. **Tiny update magnitude.** The pre-unlearning model already fits the data (*loss* ≈ 0). As shown by (Chen et al., 2019), when the source and target domains are similar, even simple few-shot methods can perform well with minimal fine-tuning. Hence, once the gradient

direction is reliable, only a small update is needed. In other words, because unlearning starts from a model that already fits the data, only the label-specific contribution must be removed, making the required parameter displacement is tiny. In the quadratic loss setting analysed in our theoretical analysis (Appendix A.1), the required step length scales with the gradient norm. Once the direction is correct, even a single mini-batch update can eliminate the label.

3. **Exponential concentration of the gradient direction.** Extending Eq. equation 12 in our Appendix A.1 with the vector Bernstein inequality (see Section 2.8 of (Vershynin, 2018)) gives
$$\Pr\bigl[\cos\angle(\nabla_\omega \ell_{\text{mix},i}, \nabla_\omega \ell) \le 1-\varepsilon\bigr] \;\le\; \exp\bigl(-c\,|D^{u,p}|\,\varepsilon^2\bigr),$$
where c depends on $\mathbb{E}[\lambda^2]$. Thus, the chance that the mixed-sample gradient deviates substantially from the full-data gradient vanishes exponentially with the number of public samples; a few dozen already give high-probability alignment.

   Specifically, the underlying tail bound for the sum of augmented embeddings follows precisely the Bernstein inequality (Vershynin, 2018):
$$P\{|\textstyle\sum_i X_i| \ge t\} \;\le\; 2\exp[-c\min\{t^2/\textstyle\sum_i \|X_i\|_{\psi_1}^2,\, t/\max_i \|X_i\|_{\psi_1}\}],$$
where $X_i = \cos\angle(\nabla_\omega \ell_{\text{mix},i}, \nabla_\omega \ell)$ and $\ell_{\text{mix},i}$ represents i_{th} mixture gradients.

Intuition. Each augmented sample acts like an arrow pointing toward the true unlearning direction. Applying manifold mixup to only a small number of data samples generates thousands of synthetic gradient directions. Averaging these directions cancels out random variance, as guaranteed by the Bernstein inequality. Because our model already starts near the solution, a short step along this accurately aligned direction suffices to complete unlearning.

A.5 Module-Omission Experiments

This set of experiments was conducted using the ResNet18 architecture on the CIFAR-10 dataset under a single-label unlearning scenario. In Fig. 5, GA-A (Gradient Ascent-All) refers to the use of all 5000 samples from y^u to perform gradient ascent. GA-S (Gradient Ascent-Small) uses only 40 data samples from y^u, matching our method's setting, but without the Manifold Mixup and Remained Accuracy Recovery modules. GA-S + Manifold Mixup, which corresponds to our method without the Remained Accuracy Recovery component. "Ours" refers to our full method, which also uses 40 data samples but incorporates all modules.

As shown in the Fig. 5:

1. GA-A effectively unlearns y^u but significantly degrades the performance on $\mathcal{D}^r$.
2. GA-S preserves $\mathcal{D}^r$ well but fails to effectively unlearn y^u.
3. GA-S + Manifold Mixup demonstrates improved unlearning performance but still results in moderate degradation on $\mathcal{D}^r$.
4. Our full method achieves both successful unlearning of y^u and strong preservation of performance on $\mathcal{D}^r$, validating the contribution of the Remained Accuracy Recovery module.

A.6 Evaluation for different privacy preserving VFL methods

We evaluate our unlearning method under two privacy preserving VFL methods: (i) Differential Privacy (Fu et al., 2022b) and (ii) Gradient Compression (Fu et al., 2022b). Fig. 6 presents the effectiveness of our solution on both methods across different levels of variance Gaussian noise and compression ratio. Higher Gaussian noise levels and greater gradient compression ratios enhance privacy protection in VFL but lead to increased performance degradation. It shows that even for a large compression ratio and noise level, our proposed method can still unlearn effectively, while the utility of the vertical training decreases significantly.

A.7 Two-label and multi-label unlearning

Tables 5 and 6 present the experimental results obtained from the two-label and multi-label unlearning scenarios, respectively.

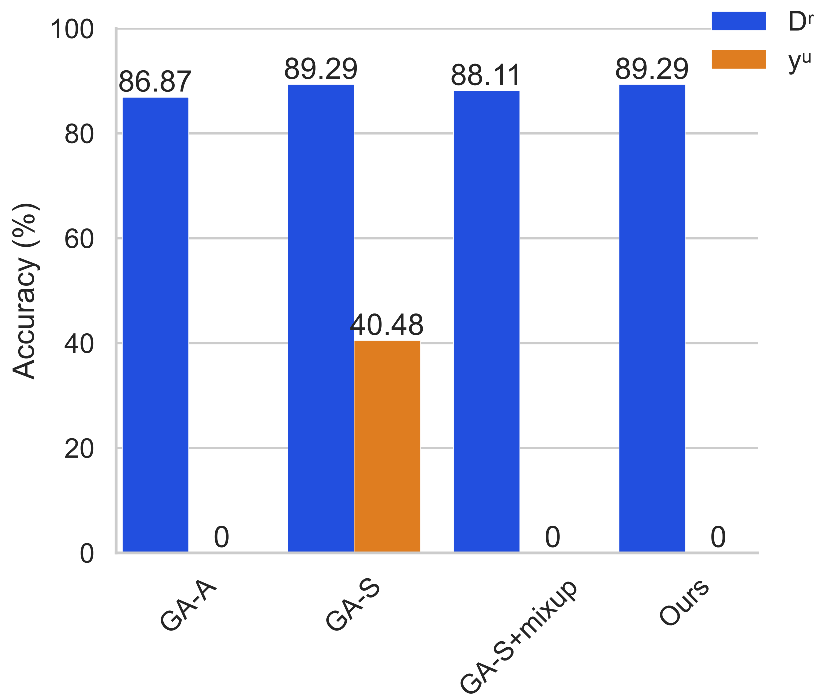


Figure 5: Comparison of the utility and unlearning effectiveness on different sizes of $\mathcal{D}^{p,u}$.

Model	Datasets	Metrics	Accuracy (%)								
			Baseline	Retrain	FT	Fisher	Amnesiac	Unsir	BU	SSD	Ours
ResNet18	CIFAR10	$\mathcal{D}^r$	91.48	91.74 ± 0.01	**90.63 ± 0.57**	31.25 ± 2.23	86.16 ± 0.82	74.48 ± 0.06	81.64 ± 0.56	88.17 ± 0.81	88.75 ± 0.36
		y^u	88.40	0.00 ± 0.00	41.15 ± 1.55	49.55 ± 0.40	**0.00 ± 0.00**	**0.00 ± 0.00**	19.90 ± 0.85	4.55 ± 0.31	0.43 ± 0.11
		ASR	79.61	21.66 ± 0.64	**13.22 ± 0.37**	25.60 ± 0.08	**1.84 ± 0.13**	41.79 ± 1.35	35.40 ± 1.54	35.60 ± 0.17	24.21 ± 1.04
	CIFAR100	$\mathcal{D}^r$	71.56	71.21 ± 0.13	66.04 ± 0.58	53.56 ± 2.54	59.52 ± 0.03	58.02 ± 0.37	56.37 ± 0.39	66.12 ± 2.98	**67.93 ± 0.26**
		y^u	71.00	0.00 ± 0.00	38.00 ± 0.01	25.20 ± 5.75	**0.00 ± 0.00**	**0.00 ± 0.00**	13.00 ± 0.01	**0.00 ± 0.00**	1.17 ± 0.29
		ASR	88.60	21.60 ± 0.85	**19.20 ± 1.20**	48.90 ± 0.54	**6.50 ± 0.40**	54.83 ± 0.44	**13.70 ± 0.90**	**8.03 ± 0.54**	**6.10 ± 0.66**
Vgg16	CIFAR10	$\mathcal{D}^r$	89.80	91.13 ± 0.03	88.09 ± 0.35	47.53 ± 2.38	86.16 ± 0.19	71.50 ± 0.07	88.67 ± 0.22	86.99 ± 0.99	**88.82 ± 0.39**
		y^u	89.10	0.00 ± 0.00	28.55 ± 0.33	13.10 ± 0.28	**0.00 ± 0.00**	**0.00 ± 0.00**	19.08 ± 0.53	3.25 ± 1.09	**0.00 ± 0.00**
		ASR	82.64	28.31 ± 1.23	**17.75 ± 2.22**	68.43 ± 1.14	**1.67 ± 0.01**	46.21 ± 0.72	**11.72 ± 0.07**	**5.27 ± 0.01**	**28.27 ± 1.51**
	CIFAR100	$\mathcal{D}^r$	65.75	65.59 ± 0.17	60.79 ± 0.37	35.24 ± 2.21	57.86 ± 0.81	56.04 ± 0.44	50.02 ± 0.18	58.97 ± 0.05	**63.40 ± 0.13**
		y^u	58.50	0.00 ± 0.00	11.75 ± 1.25	11.00 ± 4.85	**0.00 ± 0.00**	**0.00 ± 0.00**	3.25 ± 0.25	**0.00 ± 0.00**	**0.00 ± 0.00**
		ASR	73.60	30.55 ± 0.05	**22.75 ± 1.05**	32.60 ± 1.17	**3.45 ± 0.65**	52.40 ± 0.80	**27.90 ± 1.20**	**8.30 ± 0.09**	**30.47 ± 3.11**

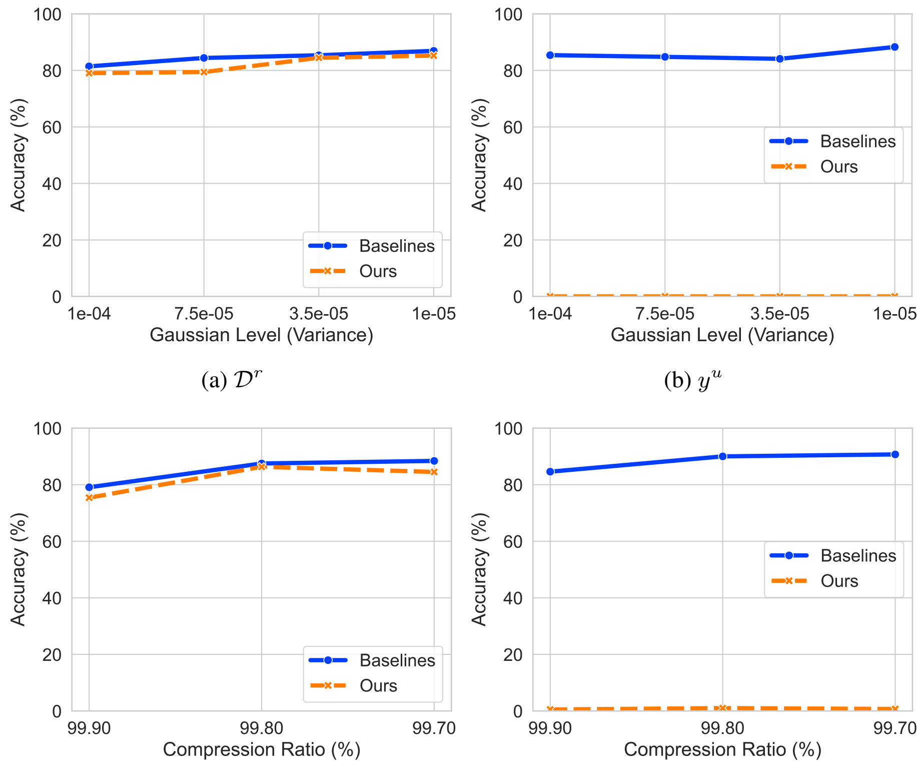
Table 5: Accuracy of $\mathcal{D}^r$, y^u and ASR for each unlearning method across ResNet18 and Vgg16 models in two-labels unlearning

Table 5 provides a detailed breakdown of the performance metrics and outcomes observed during the two-label unlearning process across various datasets and architectures. It highlights the accuracy scores of $\mathcal{D}^r$, y^u and ASR when unlearning was applied to labels *"0"* and *"2"*, demonstrating the utility guarantees and the effectiveness of the baseline unlearning methods.

Similarly, Tab. 6 summarizes the results of the multi-label unlearning experiments, where labels *"0"*, *"2"*, *"5"*, and *"7"* were unlearned for all datasets and architectures. This table captures critical metrics such as the accuracy scores of $\mathcal{D}^r$, y^u and ASR reflecting the outcomes of the multi-label unlearning process.

A.7.1 Utility Guarantee

From Tab. 5, the FT method shows performance consistent with the single-label unlearning scenario. It preserves $\mathcal{D}^r$ well on datasets with a small number of labels (*e.g.*, CIFAR-10) but performs



(a) $\mathcal{D}^r$ (b) y^u

(c) $\mathcal{D}^r$ (d) y^u

Figure 6: Comparison of remained accuracy $\mathcal{D}^r$ and unlearned accuracy y^u under Gaussian noise variance (a, b) and gradient compression ratio (c, d). Subplots (a, c) represent the remained accuracy $\mathcal{D}^r$, while (b, d) show the unlearned accuracy y^u.

Model	Datasets	Metrics	Accuracy (%)								
			Baseline	Retrain	FT	Fisher	Amnesiac	Unsir	BU	SSD	Ours
ResNet18	CIFAR100	$\mathcal{D}^r$	71.53	71.91 ± 0.12	67.16 ± 0.13	54.79 ± 1.04	59.09 ± 0.54	59.05 ± 0.38	48.96 ± 0.04	67.09 ± 1.11	**69.97 ± 0.03**
		y^u	72.00	0.00 ± 0.00	33.87 ± 0.88	45.38 ± 1.13	**0.00 ± 0.00**	**0.00 ± 0.00**	15.00 ± 0.25	0.11 ± 0.07	0.33 ± 0.14
		ASR	86.65	16.95 ± 0.35	18.23 ± 1.63	62.78 ± 3.93	**6.05 ± 1.19**	68.63 ± 1.83	38.35 ± 0.75	**3.33 ± 0.54**	**12.40 ± 1.12**
Vgg16	CIFAR100	$\mathcal{D}^r$	65.83	65.66 ± 0.08	60.92 ± 0.08	36.55 ± 1.07	57.26 ± 0.18	56.86 ± 0.26	47.04 ± 0.32	61.01 ± 0.55	**64.44 ± 0.12**
		y^u	60.25	0.00 ± 0.00	7.63 ± 0.13	28.75 ± 1.25	**0.00 ± 0.00**	**0.00 ± 0.00**	7.13 ± 0.11	**0.00 ± 0.00**	1.17 ± 0.25
		ASR	75.80	27.20 ± 0.75	**24.38 ± 3.13**	55.20 ± 3.75	**4.80 ± 0.05**	32.83 ± 0.58	29.70 ± 0.03	**10.98 ± 0.59**	27.93 ± 0.29

Table 6: Accuracy of $\mathcal{D}^r$, y^u and ASR for each unlearning method across ResNet18 and Vgg16 models in multi-labels unlearning

poorly on datasets with many labels (*e.g.*, CIFAR-100). The Fisher Forgetting method consistently demonstrates poor preservation of $\mathcal{D}^r$ across all datasets and model architectures. The Amnesiac approach performs similarly to FT, achieving strong preservation of $\mathcal{D}^r$ on datasets with fewer labels (e.g., CIFAR-10) but showing reduced performance on datasets with a larger label space (e.g., CIFAR-100). The Unsir and BU methods also follow this trend, maintaining good preservation on simpler datasets like CIFAR-10, but experiencing greater degradation in $\mathcal{D}^r$ accuracy across all datasets compared to FT and Amnesiac. SSD demonstrates strong preservation of $\mathcal{D}^r$ across all experiments, with the exception of the CIFAR100 dataset under the VGG16 architecture.

In contrast, our method consistently achieves strong unlearning utility while perfectly preserving $\mathcal{D}^r$ across all experimental settings.

From Tab. 6, the FT method shows similar behavior to the single-label and two-label unlearning scenarios, with poor preservation of $\mathcal{D}^r$ on datasets that have a large number of labels (*e.g.*, CIFAR-100). The Fisher Forgetting method consistently performs poorly in preserving $\mathcal{D}^r$ across all datasets and model architectures. The Amnesiac approach also mirrors the performance of FT, showing reduced preservation on datasets with many labels. Similarly, the Unsir and BU methods exhibit even greater degradation in $\mathcal{D}^r$ performance across all datasets compared to FT and Amnesiac. SSD yields comparable results in both single-label and two-label unlearning scenarios, showing strong preservation of $\mathcal{D}^r$, though its performance is slightly inferior to ours.

In contrast, our method demonstrates strong unlearning utility and achieves near-perfect preservation of $\mathcal{D}^r$ across all experimental settings, even in multi-label unlearning scenarios and for datasets with a large number of labels.

In conclusion, our experimental results across *single-label*, *two-label*, and *multi-label* unlearning tasks demonstrate that our method consistently delivers the best balance between utility and privacy preservation. For example, unlike FT, which maintains $\mathcal{D}^r$ accuracy only on datasets with few labels (*e.g.*, CIFAR-10) but suffers on CIFAR-100, our method achieves near-perfect preservation of $\mathcal{D}^r$ regardless of label complexity. Fisher Forgetting performs poorly across all datasets and architectures, leading to significant drops in retained accuracy. Amnesiac Unlearning introduces instability through random relabeling of $\mathcal{D}_u$, causing accuracy degradation in high-label datasets, while Unsir and BU show even greater declines, especially in multi-label settings. SSD shows good preservation of $\mathcal{D}^r$ across different settings but its performance is slightly inferior to ours. In contrast, our method maintains high $\mathcal{D}^r$ accuracy (often above 98%) while reducing unlearned data accuracy close to random chance, consistently across all scenarios. These results provide strong evidence that our approach offers superior robustness, effectiveness, and generalizability for practical, privacy-preserving unlearning.

A.7.2 Unlearning Effectiveness

We evaluate unlearning effectiveness using two key metrics: the accuracy on y^u and the Attack Success Rate (ASR). A lower accuracy on y^u indicates more effective unlearning. However, an ASR of 0% may signal the presence of the Streisand effect (Golatkar et al., 2020a), where the model consistently misclassifies all y^u samples into the same incorrect label, which indicates potentially leaking information about the unlearned label. Ideally, the ASR should be slightly lower than that of a retrained model, reflecting successful unlearning without revealing leakage. If the model consistently assigns all y^u to a specific wrong label, it might expose that the model has treated y^u uniquely. It suggests the model has not truly *"forgotten"* the data but developed an unusual bias against it.

Analysis on y^u : This paragraph provides a detailed analysis of each baseline method's performance with respect to the y^u accuracy metric.

From Tab. 5, both FT and Fisher Forgetting methods exhibit poor unlearning effectiveness in the two-label unlearning scenario. These methods perform effectively only in the single-label unlearning context. In contrast, both the Amnesiac and Unsir methods consistently achieve perfect unlearning effectiveness in the two-label unlearning scenario. The BU method shows a decline in unlearning effectiveness compared to its performance in single-label unlearning, indicating challenges in handling scenarios with two labels. SSD demonstrates strong unlearning effectiveness on the CIFAR100 dataset but fails to reduce the accuracy of y^u to 0.00% on CIFAR10 dataset. Meanwhile, our method continues to demonstrate exceptional unlearning effectiveness, maintaining its outstanding performance in the two-label unlearning scenarios, further solidifying its robustness and efficiency.

From Tab. 6, the FT and Fisher Forgetting methods exhibit similarly poor unlearning effectiveness in the multi-label unlearning scenario, comparable to their performance in the two-label unlearning context. These methods are effective only in the single-label unlearning scenario. In contrast, the Amnesiac and Unsir methods consistently achieve perfect unlearning effectiveness in the multi-label scenario, mirroring their performance in single-label and two-label contexts. The BU method performs similarly to the two-label scenario in the multi-label context, showing limited unlearning effectiveness. SSD shows good unlearning effectiveness on multi-label scenario. Meanwhile, our method continues to demonstrate exceptional unlearning effectiveness, maintaining outstanding performance in multi-label scenarios, further reinforcing its robustness and efficiency.

In conclusion, our method demonstrates superior performance across all unlearning scenarios. Specifically, it consistently achieves near-perfect preservation of retained data $\mathcal{D}^r$, often maintaining accuracy drops below 2% while reducing the accuracy of unlearned labels to near-random levels (*e.g.*, from 41.63% to 1.41%). In contrast, FT and Fisher Forgetting fail to scale beyond simple settings: they perform adequately only in single-label scenarios but suffer substantial degradation in $\mathcal{D}^r$ accuracy when applied to two-label or multi-label tasks, especially on datasets like CIFAR-100. Amnesiac and Unsir do achieve strong unlearning effectiveness across all scenarios, but at the cost of destabilizing the decision boundaries, which leads to marked drops in $\mathcal{D}^r$ accuracy for more complex datasets. BU also shows deteriorating performance as task complexity increases. SSD performs well in unlearning, but shows minor weakness on CIFAR10 under the two-label setting. Only our method balances both objectives, that is effectively unlearning target labels while preserving the rest, across all datasets, architectures, and label complexities. These concrete results affirm our approach as the most robust, generalizable, and practically deployable solution for modern unlearning needs.

Analysis on ASR : This paragraph discusses how each baseline method performs in terms of ASR, highlighting their strengths and weaknesses.

From Tab. 5 and 6, FT demonstrates excellent performance, achieving ASR scores that are consistently lower but close to those of the retrain model across all datasets, with the exception of the multi-labels scenario involving the ResNet18 architecture and the CIFAR100 dataset. In that case, the ASR score is only slightly higher than that of the retrain model. In contrast, Fisher Forgetting performs poorly in both the two-labels and multi-labels scenarios, exhibiting high ASR scores across all experiments, which indicates its ineffectiveness in mitigating adversarial risks. The Amnesiac method shows a consistent pattern of very low ASR scores across all experiments, potentially suggesting vulnerabilities to the Streisand effect, as observed in previous analyses. The Unsir method performs similarly to Fisher Forgetting, also achieving high ASR scores across all experiments, further highlighting its ineffectiveness. Boundary Unlearning continues to exhibit inconsistent performance throughout the experiments, failing to deliver reliable results. SSD continues to exhibit its good performance throughout the experiments, with consistently achieving ASR scores lower than and close to those of the retrain model. On the other hand, our method demonstrates outstanding performance on many datasets, achieving ASR scores that are consistently lower than and close to those of the retrain model. This highlights our method's effectiveness and reliability in addressing adversarial risks while maintaining robust unlearning outcomes.

In conclusion, the results clearly demonstrate the effectiveness and robustness of our method in mitigating adversarial risks. Across various datasets, our approach consistently achieves ASR scores that are not only low but often even lower than those of the retrained models, indicating strong unlearning without compromising security. While FT and SSD perform reasonably well in most scenarios, other methods such as Fisher Forgetting, Unsir, and Boundary Unlearning exhibit limited effectiveness, with high ASR scores and inconsistent performance. Although the Amnesiac method achieves very low ASR scores, it raises concerns about potential vulnerabilities, such as the Streisand effect. In contrast, our method strikes the optimal balance between unlearning effectiveness and adversarial robustness, establishing it as the most reliable and secure solution among all evaluated approaches.

A.7.3 Further Analysis of Baselines

Fine-Tuning (FT): FT demonstrates adequate performance in utility preservation for datasets with a smaller number of labels (*e.g.*, CIFAR10), its effectiveness diminishes sharply as the dataset complexity increases, failing to preserve $\mathcal{D}^r$ effectively for datasets with a large number of labels (*e.g.*, CIFAR100). Additionally, its unlearning effectiveness is limited to single-label scenarios, highlighting its inability to handle more complex unlearning contexts.

Fisher Forgetting: This method struggles across all dimensions, exhibiting poor preservation of $\mathcal{D}^r$ irrespective of the dataset or architecture. It also fails to achieve unlearning effectiveness in two-label and multi-labels unlearning scenario, with high adversarial success rates (ASR) that compromise its ability to address adversarial risks. This consistent underperformance underscores the method's lack of robustness in diverse scenarios.

Amnesiac Unlearning: While the Amnesiac method demonstrates strong unlearning effectiveness across various scenarios, its utility preservation declines significantly when applied to datasets with a large number of labels, such as CIFAR-100. This is primarily due to its approach of randomly assigning incorrect labels to $\mathcal{D}_u$, which introduces unpredictable shifts in decision boundaries and undermines reliability in more complex settings. Additionally, the method consistently yields very low ASR scores across all experiments. This implies that, although the low ASR scores indicate

Method	Type of FL	Unlearning Types	Unlearning Targets	Protection for Unlearned Data
FedEraser (Liu et al., 2021)	Horizontal	Exact	Client	✗
FRU (Yuan et al., 2023)		Approximate	Client	✗
FedRecovery (Zhang et al., 2023a)		Approximate	Client	✗
VeriFI (Gao et al., 2024)		Approximate	Client	✗
HDUS (Ye et al., 2024)		Approximate	Client	✗
KNOT (Su & Li, 2023)		Approximate	Client	✗
FedRecover (Cao et al., 2023a)		Approximate	Client	✗
Knowledge Distillation (Wu et al., 2022)		Approximate	Client	✗
Discriminative Pruning (Wang et al., 2022)		Approximate	Class	✗
MoDe (Zhao et al., 2024)		Approximate	Class	✗
Rapid Retraining (Liu et al., 2022)		Exact	Sample	✗
QuickDrop (Dhasade et al., 2023)		Approximate	Sample	✗
FedAU (Gu et al., 2024b)		Approximate	Class, Sample & Client	✗
Ferrari (Gu et al., 2024a)		Approximate	Feature	✗
Fast Retraining (Wang et al., 2024)	Vertical	Exact	Passive Party	✗
SecureCut (Li et al., 2024)		Approximate	Instance & Passive Party	✗
Constraint Imposing (Deng et al., 2023)		Approximate	Passive Party	✗
Backdoor Certification (Han et al., 2025)		Approximate	Passive Party	✗
Ours		**Approximate**	**Label**	✓

Table 7: Comparison of our method with existing studies on Federated Unlearning. This table demonstrates that our method is the ***only one*** that ensures privacy protection for $\mathcal{D}_u$ during the unlearning process. To the best of our knowledge, it is also the first approach to tackle label unlearning in VFL while safeguarding the unlearned label throughout the process.

strong unlearning, they may also point to a susceptibility to the Streisand effect, as observed in prior analyses.

Unsir and BU Methods: Both methods show similar limitations, with significant degradation in utility preservation as dataset complexity increases. Although these methods achieve moderate unlearning effectiveness, they exhibit high ASR scores, indicating their vulnerability to adversarial risks. The inconsistency of their performance across scenarios highlights their lack of generalizability and robustness.

SSD: SSD demonstrates strong performance across all three evaluation metrics, namely accuracy on $\mathcal{D}^r$, accuracy on y^u and ASR score, although it remains marginally worse than our method in each case. Moreover, SSD incurs a higher computational cost, requiring longer runtime compared to our approach (see Fig. 2 in Sect. 5.2.3), which further limits its practical efficiency.

Overall, the experimental results highlight the key limitations of existing methods, such as FT, Fisher Forgetting, Amnesiac, Unsir, BU and SSD in striking a balance between utility preservation, unlearning effectiveness, and low ASR scores. While some methods may perform well in one aspect, they often fall short in others, especially when faced with complex or multi-label scenarios. In contrast, our method consistently delivers strong unlearning performance, near-perfect utility preservation, and robust ASR scores across all datasets and settings. This comprehensive and consistent performance firmly establishes our approach as the most reliable, effective, and scalable solution for tackling diverse and challenging unlearning tasks.

A.8 Related Works

Table 7 compares our method with existing studies on Federated Unlearning. Most methods target *Horizontal Federated Learning (HFL)* and focus on *Client* unlearning. A smaller set of methods addresses *Vertical Federated Learning (VFL)*, which typically involves specific unlearning for passive parties. **Privacy protection for unlearned data** is generally lacking, as shown by the prevalence of ✗ across methods. Notably, it uniquely offers **protection for unlearned data**, setting it apart from previous approaches.

Hyper-parameters	Single-label								
	Resnet18-MNIST	Resnet18-CIFAR10	Resnet18-CIFAR100	Resnet18-ModelNet	Resnet18-Brain Tumor MRI	Resnet18-COVID-19 Radiography	Vgg16-CIFAR10	Vgg16-CIFAR100	MixText-Yahoo Answer
Optimization Method	SGD	SGD	SGD	SGD	SGD	SGD	SGD	SGD	SGD
Unlearning Rate	2e-7	2e-7	5e-7	5e-7	6e-6	2e-7	2e-7	1e-7	7e-7
Recovery Rate	2e-7	2e-7	5e-7	5e-7	6e-6	2e-7	2e-7	1e-7	7e-7
Unlearning Epochs	10	20	10	4	4	5	17	9	30
Number of $\mathcal{D}^{p,u}$ Data Samples	40	40	30	30	15	40	40	30	30
Number of $\mathcal{D}^{p,r}$ Data Samples	3	3	3	3	3	3	3	3	3
Batch Size	32	32	32	32	32	32	32	32	32
Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9

Table 8: Hyperparameters used for *single-label* unlearning in our proposed solution. In this single-label unlearning scenario, all datasets are processed using the same optimization method, stochastic gradient descent (SGD), with consistent parameters: a batch size of 32, a weight decay of 5e-4, and a momentum of 0.9. The recovery rate, which is the learning rate during the recovery phase, matches the unlearning rate for all datasets. Additionally, a minimal number of unlearning epochs (a maximum of 30) and unlearning data samples ($\mathcal{D}^{p,u}$, capped at 40) are utilized. Across all experiments, the number of recovery data samples ($\mathcal{D}^{p,r}$) used in the recovery phase is uniformly set to 3.

A.9 Experiment Setup

This section provides a detailed information on our experimental settings. Tables 8 and 9 summarize the hyper-parameters for our unlearning method. Table 10 summarizes the model name, VFL framework settings, datasets and unlearn labels involved in each unlearning scenario.

Baselines: We implement the baselines with the following details.

Retrain: The dataset $\mathcal{D}^r$ is being divided among K parties and the VFL model is retrained from scratch using the same hyperparameters as the baseline.

Fine-Tuning (Golatkar et al., 2020a; Jia et al., 2023): The dataset $\mathcal{D}^r$ is being divided among K parties and the baseline VFL model is fine-tuned for 5 epochs with a learning rate of 0.01.

Fisher Forgetting (Golatkar et al., 2020a): The dataset $\mathcal{D}^r$ is being divided among K parties and the fisher information matrix (FIM) is being used to inject Gaussian noise to perturb the VFL baseline models towards exact unlearning. For every backward propagation, once the gradient is calculated, each party inject Gaussian noise into their respective VFL baseline model to perturb the baseline model toward exact unlearning.

Amnesiac (Graves et al., 2021): The dataset is divided among K parties and the baseline VFL model is retrained for 3 epochs, with the active party relabeling y^u to an incorrect random label.

Unsir (Tarun et al., 2024): The dataset is being divided among K parties. Each passive party introduces a noise matrix to the $\mathcal{D}_u$ image features they own. The noise added image features are used to impair each VFL baseline model, which is then repaired using the clean $\mathcal{D}^r$.

Boundary Unlearning (Chen et al., 2023): The dataset $\mathcal{D}_u$ is being divided among K parties. Passive parties create adversarial examples from the $\mathcal{D}_u$ image features they own, and the active party assigns a new nearest incorrect adversarial label to shrink the $\mathcal{D}_u$ to the nearest incorrect decision boundary.

SSD (Foster et al., 2024b): The dataset is divided among K parties and each parties modify their local model weights with the gradients of full data and $\mathcal{D}_u$.

Datasets: We conduct experiments on seven widely used datasets: MNIST, CIFAR10, CIFAR100, ModelNet, Brain Tumor MRI, COVID-19 Radiography and Yahoo Answers. For the MNIST, CIFAR10/100, Brain Tumor MRI and COVID-19 Radiography datasets. For all datasets, except ModelNet and Yahoo Answer, each image is split into two equal feature segments, with each segment assigned to a different passive party. For the ModelNet dataset, two 2D multi-view images are rendered for each 3D mesh model by placing virtual cameras at evenly spaced positions around the model's centroid. Each passive party is then assigned one of these rendered view. For Yahoo Answers dataset, each sample (*i.e.* a single paragraph of text) is splitted into two segments, with each passive party receiving one segment, ensuring that no single party has access to the complete text.

Figures 7 and 8 illustrate how image features of the dataset are being split among the passive party. In this setup, we consider a scenario involving two passive parties. For all image datasets, except ModelNet, each image's features are evenly split into two halves, as illustrated in Fig. 7. As an example in CIFAR10 (*i.e.*, class = bird), each passive party is assigned one half of the image (bird)

Hyper-parameters	Two-label				Multi-label	
	Resnet18-CIFAR10	Resnet18-CIFAR100	Vgg16-CIFAR10	Vgg16-Cifar100	Resnet18-CIFAR100	Vgg16-CIFAR100
Optimization Method	SGD	SGD	SGD	SGD	SGD	SGD
Unlearning Rate	1e-6	9e-7	1e-6	9e-7	3e-6	1e-6
Recovery Rate	1e-6	9e-7	1e-6	9e-7	3e-6	1e-6
Unlearning Epochs	17	15	17	5	25	17
Number of $\mathcal{D}^{p,u}$ Data Samples	40	20	40	20	15	15
Number of $\mathcal{D}^{p,r}$ Data Samples	3	3	3	3	3	3
Batch Size	32	32	32	32	64	64
Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Momentum	0.9	0.9	0.9	0.9	0.9	0.9

Table 9: Hyperparameters used for *two-label* and *multi-label* unlearning in our proposed solution.In these unlearning scenarios, all datasets are processed using the same optimization method, stochastic gradient descent (SGD), with consistent parameters: a weight decay of 5e-4, and a momentum of 0.9. For the two-label unlearning scenario, a batch size of 32 is used, while a batch size of 64 is employed for the multi-label unlearning scenario. The recovery rate, which is the learning rate during the recovery phase, matches the unlearning rate for all datasets. Additionally, a minimal number of unlearning epochs (a maximum of 25) and unlearning data samples ($\mathcal{D}^{p,u}$, capped at 40) are utilized. Across all experiments, the number of recovery data samples ($\mathcal{D}^{p,r}$) used in the recovery phase is uniformly set to 3.

Scenarios	Models	Model of Passive Party	Model of Active Party	Datasets	Unlearn Labels
Single-label Unlearning	ResNet18	20 Conv	1 FC	MNIST, CIFAR10, CIFAR100, ModelNet, COVID-19 Radiography	0
	ResNet18	20 Conv	1 FC	Brain Tumor MRI	2
	Vgg16	13 Conv	3 FC	CIFAR10, CIFAR100	0
	MixText	12 Brt	4 FC	Yahoo Answer	6
Two-label Unlearning	ResNet18	20 Conv	1 FC	CIFAR10, CIFAR100	0, 2
	Vgg16	13 Conv	3 FC	CIFAR10, CIFAR100	0, 2
Multi-label Unlearning	ResNet18	20 Conv	1 FC	CIFAR100	0, 2, 5, 7
	Vgg16	13 Conv	3 FC	CIFAR100	0, 2, 5, 7

Table 10: Models and datasets involved in each unlearning scenarios. FC: Fully-connected layer. Conv: Convolutional layer. Brt: Bert layer. In the single-label unlearning scenario, we perform unlearning for label *"0"* on the MNIST, CIFAR10, CIFAR100, ModelNet, and COVID-19 Radiography datasets using the ResNet18 architecture. Additionally, label *"0"* is unlearned on the CIFAR10 and CIFAR100 datasets using the Vgg16 architecture. For the Brain Tumor MRI dataset, we unlearn label *"2"* using the ResNet18 architecture. Similarly, on the Yahoo Answer dataset, label *"6"* is unlearned using the MixText architecture. In the two-label unlearning scenario, labels *"0"* and *"2"* are unlearned across all datasets and architectures. In the multi-label unlearning scenario, we extend the unlearning process to include labels *"0"*, *"2"*, *"5"*, and *"7"* across all datasets and architectures.

features. This means that neither party has access to the full set of image features for any given data point, ensuring that the data remains fragmented and partially hidden from each individual party. This approach is specifically designed to enhance privacy and security by ensuring that no single party has access to the full information of the original image. This partitioning allows collaborative computation or machine learning tasks to be performed while upholding data confidentiality.

For the ModelNet dataset, a different approach is used to distribute data between the two passive parties as illustrated in Fig. 8. Specifically, each 3D mesh model is rendered into two distinct 2D images, each capturing a unique view of the object from different perspectives. Then, these two generated 2D views are assigned to separate passive parties, with each party exclusively receiving one view. As a result, no single party has access to both perspectives of the 3D mesh model, preserving information separation. This strategy not only safeguards data privacy but also enables collaborative analysis or training, ensuring that each party's contribution remains independent and incomplete.

Figure 9 illustrates how text features from the Yahoo Answers dataset are split between the passive parties. In this process, each paragraph is divided into two separate segments, with each segment representing a portion of the original text. Each passive party receives access to only one of these segments, ensuring that no single party can view the entire paragraph. This approach helps preserve the confidentiality of the dataset.

MNIST dataset contains images of handwritten digits. MNIST dataset comprises 60,000 training examples and 10,000 test examples. Each example is represented as a single-channel image with dimensions of 28×28 pixels, categorized into one of 10 labels.

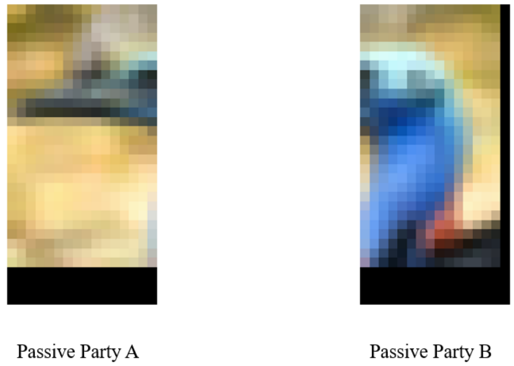


Figure 7: Illustration of how features from the CIFAR-10 dataset are split between passive parties. Passive Party A has access to one segment of the image, such as the left half of the original image's pixel data, while Passive Party B has access to the other segment, such as the right half. This division ensures that no single party has access to the complete image.

Figure 8: Illustration of how features from the ModelNet dataset are split between passive parties. Passive Party A is assigned one view of the 3D model, rendered from a specific angle (*e.g.*, an upper back-facing perspective of the airplane model), while Passive Party B receives a different view of the same model, rendered from another angle (*e.g.*, an upper right-side perspective). This setup ensures that no single party has full access to the complete 3D information.

CIFAR10/100 dataset contain 60,000 images (32×32 pixels, three color channels). CIFAR10 dataset includes 10 labels, with 50,000 for training and 10,000 for testing. The CIFAR100 dataset is similar but has 100 labels, each with 600 images, divided into 500 for training and 100 for testing.

ModelNet dataset is a widely-used 3D shape classification and retrieval benchmark, which contains 127,915 3D CAD models from 662 object categories.

Brain Tumor MRI is commonly used in healthcare scenarios. The Brain Tumor MRI dataset consists of 7,023 human brain MRI images categorized into four labels: Glioma, Meningioma, No Tumor, and Pituitary.

COVID-19 Radiography is a publicly available dataset on Kaggle, designed for research and development in medical imaging, specifically for detecting COVID-19 through chest X-rays. This dataset consists of chest X-ray images categorized into four classes, which are COVID, Normal, Viral Pneumonia and Lung Opacity.

Yahoo Answers is a dataset designed for text classification tasks, comprising 10 labels (topics) such as "Society & Culture", "Science & Mathematics", "Health", "Education & Reference", among others. Each label contains 140,000 training samples and 6,000 testing samples. For simplicity, we utilized 5,000 training samples and 2,000 testing samples from each label.

can a loan corporation garnish your wages? yes they can read your contracts very carefully before signing also k

Passive Party A

now your rights as a consumer in your state go to your states website and search for the fconsumer rights section.

Passive Party B

Figure 9: Illustration of how text from the Yahoo Answers dataset is split between passive parties. Each passive party is assigned a different portion of the text, ensuring that no single party has access to the complete content. For example, the full paragraph is: *can a loan corporation garnish your wages? yes they can read your contracts very carefully before signing also know your rights as a consumer in your state go to your states website and search for the consumer rights section.”* Passive Party A receives the first segment (*e.g.*, *can a loan corporation garnish your wages? yes they can read your contracts very carefully before signing also k”*), while Passive Party B receives the remaining portion (*e.g.*, *“now your rights as a consumer in your state go to your states website and search for the consumer rights section.”*). This division maintains the confidentiality of the full paragraph.