ProVFL: Property Inference Attacks Against Vertical Federated Learning

Li Bai, Xinwei Zhang, *Graduate Student Member, IEEE*, Sen Zhang, *Member, IEEE*, Qingqing Ye, *Member, IEEE*, and Haibo Hu, *Senior Member, IEEE*

***Abstract*—Recent studies show that privacy leakages may occur in vertical federated learning (VFL), where parties hold split features of the same samples. While various attacks, including label and feature inference, focus on record-level privacy risks in VFL, few studies delve into the distribution-level privacy threat. In this paper, we explore property inference attacks (PIAs) in VFL, where an adversarial party seeks to deduce global distribution information about a target property in the victim party's training set. Our key observation is that the L_p-norm distribution of intermediate results in VFL could reflect the fraction of the target property in a training set. Inspired by this, we present *ProVFL*, a novel PIA framework involving distribution comparison and correlation augmentation modules. To achieve property inference, we design a distribution comparison module by creating various intermediate-result populations with different proportions, aiming to learn the relationship between L_p-norm distributions and their fractions. Then, we theoretically analyze the factors that contribute to the attack effectiveness and develop a correlation augmentation module based on label replacement and model refinement to amplify property information leakage. Extensive experimental results demonstrate that our attacks can achieve inferences with low estimation errors as low as 1%. This poses the immediate threat of property information leakage from private training data in the VFL setting.**

***Index Terms*—Property inference attacks, vertical federated learning, privacy attacks, defense mechanisms.**

I. Introduction

With consideration for data privacy and security, federated learning (FL) [1] is proposed to train a high-quality model collaboratively across parties without centralizing their data. FL can be categorized into two types based on how data is partitioned within a feature and sample space: horizontal federated learning (HFL) and vertical federated learning (VFL). In contrast to HFL, where parties typically possess data with identical feature spaces, VFL operates on decentralized data sharing the same sample IDs while having different features [2], [3]. Application cases of VFL appear in finance, e-commerce, and healthcare [4], [5], [6]. As an example in Figure 1, for an e-commerce platform and bank institution, VFL offers a solution to merge distinct and distributed training samples, enabling them to develop a powerful loan prediction model without accessing original private records.

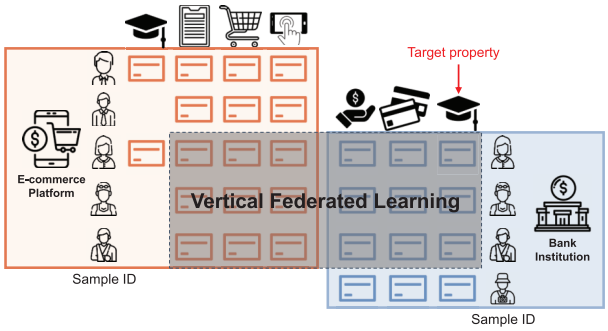


Fig. 1. An example of PIA in a two-party VFL setting. Consider a VFL system where an e-commerce platform and a bank institution collaboratively develop a loan prediction model using their overlapping samples. The e-commerce platform infers the statistical distribution of educational degrees for common users and adjusts advertising policies for this population.

Typically, a VFL system has two types of parties: passive party that solely holds sample features, and active party that supplies both sample labels and features. Each party constructs a bottom model based on its own private dataset, uploads intermediate outputs to a top model controlled by an active party, and subsequently updates the bottom model based on intermediate gradients from the top model [7]. While VFL keeps each party's private data locally, it is susceptible to various privacy attacks. Recent studies have thoroughly investigated these threats, including feature inference attacks [2], [8], [9], [10] and label inference attacks [3], [11], [12], [13]. The former entails adversaries reconstructing private data from model updates exchanged across parties, whereas the latter reveals sensitive label information of training samples [3]. Both attacks target the *record-level* privacy leakage of VFL as they operate on individual training samples.

Different from previous privacy studies, this paper pioneers the investigation of property inference attacks (PIAs) against VFL, which target *distribution-level* privacy leakage. In PIAs, an attacker can infer the characteristics and statistical information about a specific property of interest (i.e., target property) in a training set. Such distribution-level information leakage

Received 3 December 2024; revised 19 May 2025; accepted 15 June 2025. Date of publication 20 June 2025; date of current version 3 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 92270123 and Grant 62372122; in part by the Research Grants Council, Hong Kong, SAR, China, under Grant 15226221, Grant 15208923, and Grant 15224124; and in part by the Open Research Fund of the State Key Laboratory of Blockchain and Data Security, Zhejiang University. The associate editor coordinating the review of this article and approving it for publication was Dr. Paolo Gasti. *(Corresponding author: Haibo Hu.)*

Li Bai, Xinwei Zhang, and Sen Zhang are with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: baili.bai@connect.polyu.hk; xin-wei.zhang@connect.polyu.hk; senzhang@polyu.edu.hk).

Qingqing Ye is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong, and also with the State Key Laboratory of Blockchain and Data Security, Zhejiang University, Zhejiang 310027, China (e-mail: qqing.ye@polyu.edu.hk).

Haibo Hu is with the Department of Electrical and Electronic Engineering and the Research Centre for Privacy and Security Technologies in Future Smart Systems, The Hong Kong Polytechnic University, Hong Kong (e-mail: haibo.hu@polyu.edu.hk).

Digital Object Identifier 10.1109/TIFS.2025.3581743

may cause privacy breach [14], [15] or intellectual property infringement [15], [16], as an example provided in Figure 1. Understanding and exploring such privacy leakage helps shed light on privacy protection in VFL.

Unfortunately, existing works [17], [18] are not directly applicable to the VFL setting due to the following fundamental differences. At first, existing PIAs, designed for centralized learning or HFL, are impractical for VFL as they often require the training of shadow models on the full feature space [14], [15], [16], [17], [19]. However, an adversarial party in VFL has access to partial feature space, which makes it unable to replicate the complete training procedure. Besides, the heterogeneous feature spaces within VFL provide limited information for an adversarial party [18]. This restricts the effectiveness of PIAs based on attribute inference attacks (AIAs) [20], [21], which heavily relies on a strong correlation between the target property and the task label, as well as on the availability of adversarial knowledge [17]. These limitations motivate our contributions towards a specific approach as well as a thorough analysis for VFL.

To bridge this gap, we deeply investigate property information leakage within a partial feature space controlled by a malicious party in VFL. The essence of property inference is that different fractions of the training set on a target property may lead to distinction on model outputs or parameters [14], [15], [16]. Such a distinction can also be reflected in VFL intermediate results (i.e., outputs and gradients). This leads us to discover that the scale of divergences in the L_p-norm distribution of intermediate results is aligned with the fractional gap between two different populations. Building on this, we propose a novel PIA framework for VFL, named ProVFL, which infers the fraction of a target property based on the observed L_p-norm distributions of intermediate results. ProVFL comprises a distribution comparison module and a correlation augmentation module. In the first module, we randomly sample various populations with different fractions and learn the correspondence between L_p-norm distributions of intermediate results. To further enhance the attack effectiveness, we conduct a theoretical analysis of the factors influencing its performance and incorporate the correlation augmentation module. The second module incorporates two methods: (1) a label replacement technique that discreetly substitutes the original labels with the target property, embedding more relevant information into the VFL system, and (2) a model refinement strategy that enhances the attacker's bottom model by utilizing supervised knowledge of the target properties and amplifying distribution disparities between populations. Experiment results demonstrate that both of them are effective in decreasing estimation errors of property inference.

The main contributions of this paper are summarized as follows:

- We propose a novel PIA framework for VFL, named ProVFL. To the best of our knowledge, this is the first work to investigate PIAs within VFL while achieving low estimation errors.
- To achieve property inference, we devise a distribution comparison module by creating various intermediate-result populations with different proportions. This module can learn the relationship between L_p-norm distributions and their fractions.
- Motivated by our theoretical analysis, we develop label replacement and model refinement approaches in the correlation augmentation module to enhance the effectiveness of PIAs. These approaches can amplify property information leakage while improving attack performance.
- We validate the effectiveness of ProVFL on fourteen target properties in five real-world datasets. Also, several defenses are explored to safeguard against such leakage risk.

The rest of the paper is organized as follows. We introduce related work in Section II and the research problem in Section III. Then, following the key observation in Section IV, Section V presents our attack framework and detailed approaches. Section VI shows experiment settings and evaluation results. We also evaluate our attack on a set of defense mechanisms in Section VII. Finally, Section VIII concludes this paper.

II. Related Work

A. Property Inference Attacks

They aim to infer statistical information about a training set, such as the proportion of samples related to a specific target property in the dataset. By constructing a batch of shadow models and training a meta-classifier, various PIAs have been introduced across different architectures, including neural networks [19], [22], generative adversarial networks [23], graph neural networks [17], and diffusion models [24], [25]. Their core idea is that similar training datasets result in comparable model performance, either in terms of model predictions [14], [19] or model parameters [22]. Another method based on AIAs recovers the attribute values of training samples and subsequently summarizes them [23], [25]. This approach relies on the correlation between inferred attributes and labels, as well as the adversarial knowledge available [17], [26].

PIAs have also been extended to collaborative learning contexts [19], [20], [27], [28]. Melis et al. [20] utilize snapshots of the global model to construct a batch property classifier in HFL. The classifier can infer properties related to a subset of the training inputs or detect when a specific property is present in the training data. Like shadow-training methods, [19] examines the disclosure of properties in a multi-party environment through model predictions. These studies focus on PIAs in HFL, while the risk of property leakage in VFL has received little attention. Our work bridges this gap by revealing that an adversarial party can make highly accurate inferences about target properties, highlighting the immediate threat of property information leakage in VFL.

B. Defenses Against Property Inference Attacks

Intuitive defensive techniques, such as adding noise or removing sensitive properties, are not effective at protecting against property leakage. Previous research suggests that differential privacy [29], which is designed for individual privacy protection, offers limited mitigation [14], [15], [19]. Due to inherent correlations across features, removing sensitive

TABLE I
NOTATIONS

Symbol	Description
P_k, K	the k-th party, the number of party
$\mathcal{X}_k$, $\mathcal{Y}$	P_k's feature space, label space
$\mathbf{x}_i \in \mathbb{R}^{1\times d}$	the i-th full sample point
$\mathbf{x}_i^k \in \mathbb{R}^{1\times d_k}$	the i-th partial sample point of P_k
$\mathbf{v}_i^k \in \mathbb{R}^{1\times d^*}$	the output of f_{θ_k} about $\mathbf{x}_i^k$
d	the size of an entire sample space
d_k	the size of a partial sample space of $\mathcal{X}_k$
d^*	size of an intermediate output
D	the entire training set
D_k	P_k's local training set
f_{θ_k}, h_ϕ	P_k's bottom model,the top model
θ_k, ϕ	model parameters of f_{θ_k}, h_ϕ
p	target property
t	predicted fraction
ℓ	loss function

attributes cannot solve property leakage inherently [19]. As for FL, another strategy is to reduce the amount of information exchanged between parties, such as by sharing fewer gradients [20], [30] and adding noise to model updates [22], [27], [31]. However, these techniques fail to provide adequate protection while maintaining acceptable model utility. Hence, there is a lack of rigorous and powerful defenses against distribution-level information leakage [16], [32].

III. Problem Formulation

A. Vertical Federated Learning

Following [7], [33], we consider a typical VFL setting in which K parties $\{P_1, P_2, \ldots, P_K\}$ use N overlapping data samples $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ to train a machine learning model. Each sample $\mathbf{x}_i$ is constructed by joining the distributed features $\mathbf{x}_i^k \in D_k$ across parties and the label $y_i \in \mathcal{Y}$ is provided by the active party.

Upon preparing the training set, P_k utilizes its local dataset to train a bottom model f_{θ_k} parameterized with θ_k. Let $\mathbf{v}_i^k$ denote its intermediate output, i.e., $\mathbf{v}_i^k = f_{\theta_k}(\mathbf{x}_i^k)$. A top model h_ϕ aggregates all intermediate outputs in a certain way, e.g., by concatenation, where ϕ denotes the top model parameter. Let $\mathbf{v}_i = [\mathbf{v}_i^1, \mathbf{v}_i^2, \ldots, \mathbf{v}_i^K]$ denote the concatenated vector of $\mathbf{x}_i$. Subsequently, h_ϕ makes the final prediction as $h_\phi(\mathbf{v}_i)$ and computes corresponding gradients by minimizing the following objective function:

$$\mathcal{L} = \frac{1}{|D|} \sum_{(\mathbf{x}_i, y_i)\in D} \ell\left(h_\phi\left(f_{\theta_1}\left(\mathbf{x}_i^1\right), \ldots, f_{\theta_K}\left(\mathbf{x}_i^K\right)\right), y_i\right), \quad (1)$$

where ℓ is a loss function, e.g., the cross-entropy function. The active party calculates the loss using Eq.(1) and updates the top model using $\nabla_\phi \mathcal{L}$. Next, the active party computes the partial gradients for each bottom model and sends them back. Finally, each party updates its respective bottom model accordingly. Detailed notations can be found in Table I.

B. Record-Level Intermediate Result

Unlike aggregated model updates in HFL, a VFL adversary could acquire fine-grained exchanged information, including record-level intermediate output and gradient. Intuitively, a party P_k can access the intermediate output of a training sample $\mathbf{x}_i^k$ with its bottom model, i.e., $\mathbf{v}_i^k$. Moreover, common sample IDs in VFL expose the intermediate gradients of individual data points to the active party who can obtain record-level intermediate gradients by replaying the optimization process, i.e., $\mathbf{g}_i^k = \nabla_{\mathbf{v}_i^k} \ell(h_\phi(\mathbf{v}_i), y_i)$. However, passive parties only observe aggregated gradients of multiple samples [2] due to lacking control over the top model or access to label information.

Before introducing possible adversarial knowledge, we define AO and AG as *record-level* intermediate outputs and gradients derived from the adversarial party's data points, respectively. Similarly, VO and VG refer to *record-level* intermediate outputs and gradients associated with the victim party's data points.

C. Adversary's Objective

We investigate PIAs in the context of VFL, where an adversarial party P_k infers statistical information about a sensitive attribute owned by a victim party. Formally, given a target property p, a training bottom model f_{θ_k} and external adversarial knowledge Ω, a PIA model can be defined as follows:

$$\mathcal{A} : \{p,\ f_{\theta_k},\ \Omega\} \rightarrow t, \quad (2)$$

where t is the predicted fraction of p in the victim party's training dataset.

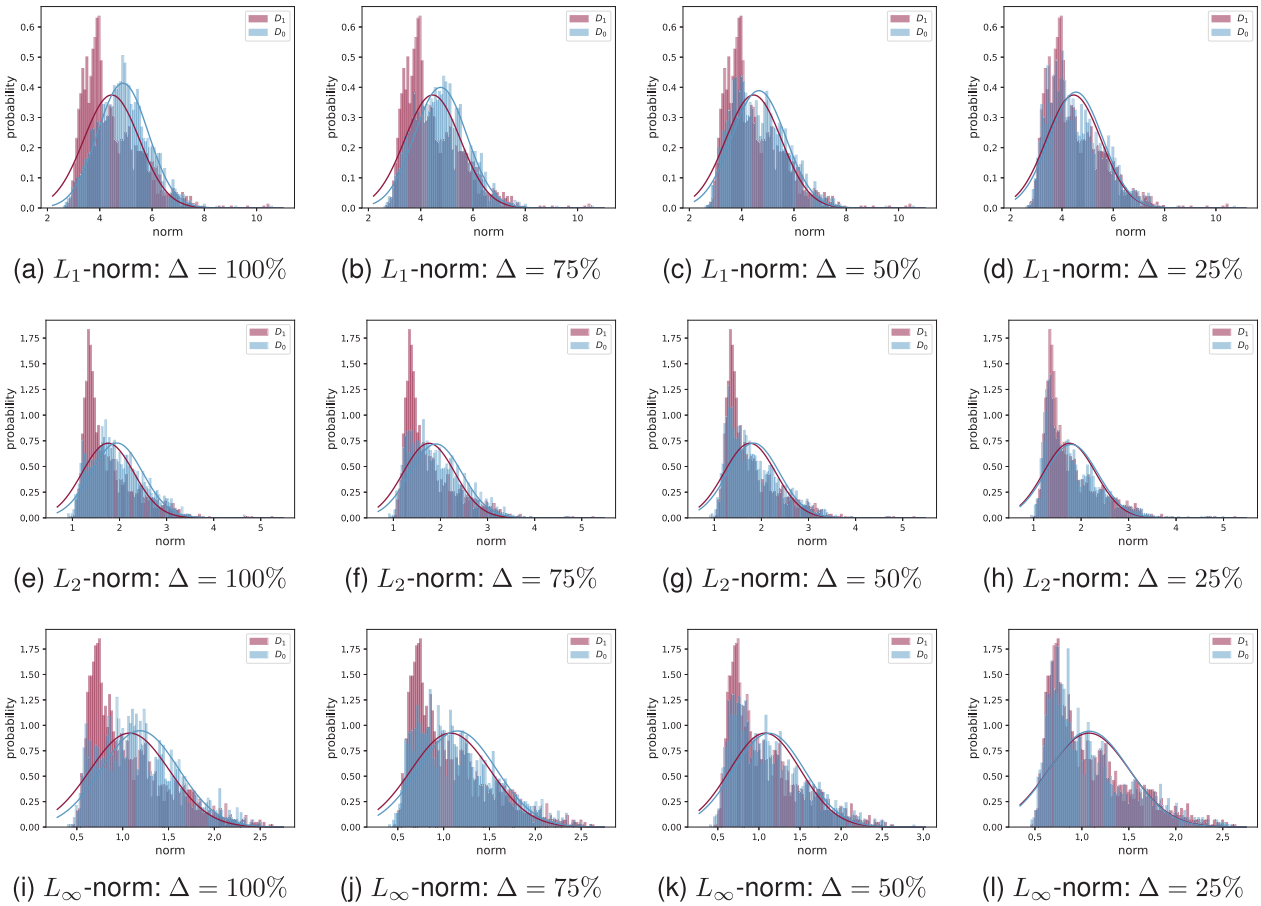
D. Adversary's Capacity

We assume that an attacker can gather some auxiliary data, with each sample labeled as either having or not having the target property. This is an assumption commonly made by other PIAs [15], [16], [22], where a key difference is that we consider those data from the adversarial party's feature space rather than the victim party's, as accessing data from other parties is typically challenging in VFL. To achieve this, an attacker might inquire about inferred properties from a subset of their users or adopt other attacks, e.g., feature inference attacks [2], [34], [35] to reconstruct some noisy samples from the victim party's side in advance. We further discuss and relax this assumption in Section VI.

With auxiliary data, an adversarial party can receive different amounts of information, including knowledge of intermediate results and the top model, to cause property leakage. Considering the adversary's role in VFL, we examine its adversarial knowledge Ω under two cases:

(C1) Active party as the PIA adversary: Given that the active party provides label information and typically possesses more computational resources, it often undertakes the training of the top model [3]. Hence, in this scenario, the adversarial knowledge encompasses four types of intermediate results and a controllable top model, i.e., $\Omega = \{\text{AO}, \text{AG}, \text{VO}, \text{VG}, h_\phi\}$.

(C2) Passive party as the PIA adversary: Compared to the active party, a passive party has access to less adversarial knowledge because its bottom model is the only controllable component. Consequently, in this case, we explore property



(a) L_1-norm: $\Delta = 100\%$ (b) L_1-norm: $\Delta = 75\%$ (c) L_1-norm: $\Delta = 50\%$ (d) L_1-norm: $\Delta = 25\%$

(e) L_2-norm: $\Delta = 100\%$ (f) L_2-norm: $\Delta = 75\%$ (g) L_2-norm: $\Delta = 50\%$ (h) L_2-norm: $\Delta = 25\%$

(i) L_∞-norm: $\Delta = 100\%$ (j) L_∞-norm: $\Delta = 75\%$ (k) L_∞-norm: $\Delta = 50\%$ (l) L_∞-norm: $\Delta = 25\%$

Fig. 2. Different L_p-norm distributions of VO between D_0 and D_1. Δ means the fraction gap of the target property.

leakage of VFL when an attacker relies solely on its record-level intermediate outputs during the training process, i.e., $\Omega = \{\text{AO}\}$.

IV. Key Observation

In this section, we first provide insights into what reveals property information to attackers in VFL, which motivates us to design a specific PIA for VFL.

The norms of latent outputs and gradients of a neural network hold a substantial amount of semantic information [36], [37], [38], which motivates many research areas, including person recognition [39], [40], [41] and model explanations [42], [43], [44]. We expect that such norms encompass side information about target properties and project intermediate results of training samples closer to those with similar properties.

To validate this conjecture, we present a case study about property information leakage of VFL in a real-world dataset. At first, we classify training samples with the target property as *property samples* and those without as *non-property samples*. We construct two kinds of datasets, D_0 and D_1, each comprising 2,000 sampled data points. Four different D_0 contain {0%, 25%, 50%, 75%} of property samples and the remaining are non-property samples, respectively, while D_1 only contains property samples. Hence the fraction gaps between D_0 and D_1 on the target property are {100%, 75%, 50%, 25%}. Subsequently, we collect the victim party's intermediate outputs (i.e., VO) of each sample in these sets and present their L_1-norm, L_2-norm and L_∞-norm distributions in Figure 2. We find that the difference becomes less pronounced as the fractional gap decreases. For example, the overlap between distributions is significantly greater in the Δ = 25% scenario than in the Δ = 75% case, regardless of the norm option. We further present a quantitative analysis of distances between these distributions in Table II, using KL divergence (KL), JS divergence (JS), and Bhattacharyya coefficient (BC). Regarding for different norm options, the results indicate that the L_1-norm type effectively captures differences between distributions, leading it as the default norm function in the following.

TABLE II
The Distance Comparison of Different Norm Options

Δ	Option	KL ↑	JS ↑	BC ↓
25%	L_1-norm	0.0017	0.0202	0.9684
	L_2-norm	0.0000	0.0012	0.9990
	L_∞-norm	0.0004	0.0095	1.0000
100%	L_1-norm	0.0327	0.0876	0.8692
	L_2-norm	0.0086	0.0458	0.9665
	L_∞-norm	0.0145	0.0592	0.9856

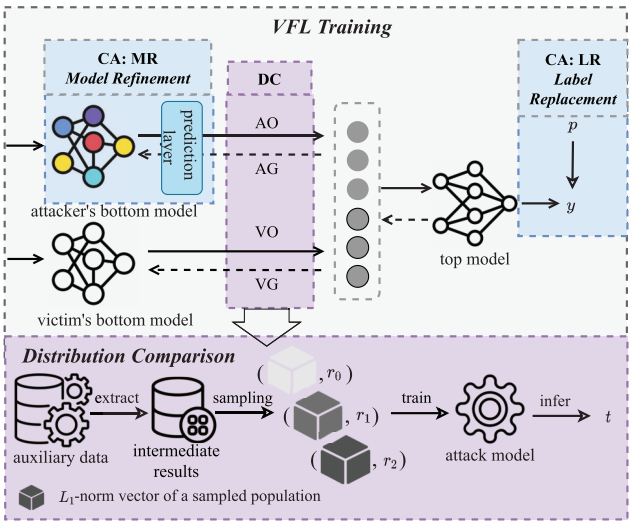


Fig. 3. An overview of ProVFL in a two-party VFL setting. ProVFL involves two main modules: distribution comparison (DC) and correlation augmentation (CA). Within CA, there are two approaches: Label Replacement (LR) and Model Refinement (MR).

The above observation shows that the norms of intermediate results contain property-related information, indicating a correspondence between the L_p-norm distributions of intermediate-result populations and their inherent fractions. Drawing inspiration from this, we leverage distribution-level differences to identify the statistical characteristics of a target property. The following section details the process of our proposed PIAs by learning such correspondence in VFL.

V. Attack Framework

Exchanged intermediate results among VFL parties inadvertently expose property information to possible adversaries during training. Based on that, we propose a novel framework of PIAs in VFL, named ProVFL, in which an adversarial party can identify the exact fraction of a target property owned by other parties with low inference error. As Figure 3 shows, it involves two modules: *distribution comparison* (DC) exploits the correspondence between intermediate-result distributions and fractions of the target property to infer property information, and *correlation augmentation* (CA) increases the property leakage of VFL to boost the attack performance. Within the CA module, we design two approaches, label replacement and model refinement, to achieve better attack performance. We introduce each module of ProVFL in detail below.

A. Distribution Comparison

The core idea behind DC is to create multiple intermediate-result distributions with different fractions and then compare these to the victim party's distribution. To achieve this, DC involves three phases: (1) Distribution generation constructs various intermediate-result populations from auxiliary data. (2) Regression ensemble learns the correspondence to target properties by developing a set of regression models. (3) Property inference identifies the fraction of the victim party's dataset. We assume that an adversarial party initiates a PIA at the e_{SEC}-th training epoch. As Algorithm 1 presents, DC is performed as follows:

1) **Distribution generation** (Lines 1-7). In this process, the adversary aims to construct pairwise training data points for an attack model by sampling intermediate result populations with varying fractions related to the target property.
Given a specific kind of intermediate result to use, such as one of {AO, AG, VO, VG}, the attacker, e.g., P_k, first extracts these from auxiliary dataset D_{aux} and the attacker's training dataset D_k, respectively. Each sample in the auxiliary dataset originates from the adversary's feature space and is additionally labeled to indicate the presence of the target property. Such data can be collected from a subset of users on the adversary's side or inferred using advanced feature inference attacks. We define them as D^{ir}_{aux} and D^{ir}_{k}, where each record corresponds to the intermediate result representation of an individual sample. Next, the adversary creates a fraction candidate set {0%, q%,..., 100%} with a q% interval. For its fraction element r, DC creates an intermediate-result subset where r of the samples corresponds to property samples, while the remaining $1-r$ of the subset belongs to non-property samples. Subsequently, we construct the L1-norm distribution of this subset by concatenating the L_1-norm values of all elements. We use $\hat{\mathbf{z}} = [\|z_1\|_1, \|z_2\|_1, \ldots, \|z_M\|_1] \in \mathbb{R}^{1\times M}$, where $\|\cdot\|_1$ denotes the L_1-norm function and z_i is the intermediate result of a training sample. Finally, we obtain an attack data point $(\hat{\mathbf{z}}, r)$ and then establish an attack training set D_{att} for each kind of intermediate result in a similar way.
2) **Regression ensemble** (Line 8). After preparing the attack training set, DC employs regression models to learn the relationship between L_1-norm distributions and fractions. These regression models serve as the attack model for property inference.
We use g_φ to refer to a regression model with φ representing its parameters and train it to minimize the total sum of absolute errors between predicted and actual fractions in D_{att}, formulated as $\varphi \leftarrow \arg\min_{\varphi} \frac{1}{|D_{\text{att}}|}\sum_{(\hat{\mathbf{z}},r)\in D_{\text{att}}} \ell\left(g_\varphi(\hat{\mathbf{z}}), r\right)$, where ℓ denotes a loss function used in regression tasks, such as mean squared error.
When multiple types of intermediate results are observable in C1, we employ ensemble methods by combining regression models trained on different types of attack training sets to improve overall prediction accuracy. Specifically, each regression model is initially trained independently using a specific type of intermediate result. The predictions from all models are then aggregated into an ensemble by averaging their outputs. This ensemble is ultimately used to predict property inference. For the C2 scenario, only a single regression model is necessary.
3) **Property inference** (Lines 9-13). To estimate the proportion of a target property within the victim party's dataset, an attacker queries the attack model using the

Algorithm 1 Distribution Comparison: Using Intermediate Results to Infer Property Information in VFL

Input: D_k^{ir}: intermediate results of D_k
$D_{\text{aux}}^{\text{ir}}$: intermediate results of D_{aux}
R: size of the attack training set
Q: number of distribution queries
M: sample size per query
q: step size of fraction interval
Output: Predicted property fraction in D_k^{ir}

Split $D_{\text{aux}}^{\text{ir}}$ into property and non-property sets: D_{rep}^{+}, D_{rep}^{-} ;
$D_{\text{att}} \leftarrow []$; // Initialize attack training set
for $i = 1$ **to** R **do**
  $r \leftarrow \text{random\_sampling}(\{0\%, q\%, \ldots, 100\%\}, 1)$;
  $D_0 \leftarrow \text{random\_sampling}(D_{\text{rep}}^{+}, r \cdot M)$; // Sample property instances
  $D_1 \leftarrow \text{random\_sampling}(D_{\text{rep}}^{-}, (1-r) \cdot M)$; // Sample non-property instances
  $x \leftarrow L_1\text{\_norm}(D_0 \cup D_1)$; // Extract distributional representation
  $D_{\text{att}} \leftarrow D_{\text{att}} \,||\, (x, r)$; // Append to attack set
$g_\varphi \leftarrow \text{TRAIN}(g_\varphi; D_{\text{att}})$; // Train the attack model
$Pr \leftarrow []$; // Store predicted property fractions
for $i = 1$ **to** Q **do**
  $D_q \leftarrow \text{random\_sampling}(D_k^{\text{ir}}, M)$;
  $x_q \leftarrow L_1\text{\_norm}(D_q)$;
  $Pr \leftarrow Pr \,||\, g_\varphi(x_q)$
return mean(Pr)

Algorithm 2 Passive and Active Property Inference Attacks for Two-Party VFL

Input: $\mathbf{x}^1$: features from passive party
$\mathbf{x}^2$, y: features and labels from active party
p: property of samples, η: learning rate
f_{θ_1}: local model of passive party
f_{θ_2}: local model of active party
h_ϕ: top model owned by active party
$f_{\theta_\mathcal{T}}$: auxiliary model for inference attack
Output: Trained models f_{θ_1}, f_{θ_2}, and h_ϕ

for *each epoch* **do**
  for *each mini-batch* $(\mathbf{x}^1, \mathbf{x}^2, y)$ **do**
    $y \leftarrow p$; // LR: Replace y with property p for attack
    $\mathbf{v}_1 \leftarrow f_{\theta_1}(\mathbf{x}^1)$; // Passive party forward pass
    $\mathbf{v}_2 \leftarrow f_{\theta_2}(\mathbf{x}^2)$; // Active party forward pass
    $\mathbf{v} \leftarrow [\mathbf{v}_1; \mathbf{v}_2]$; // Concatenate representations
    $\mathcal{L} \leftarrow \ell(h_\phi(\mathbf{v}), y)$; // Active party computes loss
    Compute gradients: $\nabla_{h_\phi}\mathcal{L}$, $\nabla_{\mathbf{v}_1}\mathcal{L}$, $\nabla_{\mathbf{v}_2}\mathcal{L}$;
    Update f_{θ_1} using $\nabla_{\mathbf{v}_1}\mathcal{L}$; // Passive party update
    Update f_{θ_2}, h_ϕ using $\nabla_{\mathbf{v}_2}\mathcal{L}$, $\nabla_{h_\phi}\mathcal{L}$; // Active party update
    $D^{\text{ir}} \leftarrow \{\mathbf{v}_1, \mathbf{v}_2, \nabla_{\mathbf{v}_1}\mathcal{L}, \nabla_{\mathbf{v}_2}\mathcal{L}\}$; // DC: Store for passive attack DC
    $h \leftarrow f_{\theta_\mathcal{T}}(\mathbf{v}_1)$; // MR: Attack by passive party; use $\mathbf{v}_2$ for active
    $\mathcal{L}_o \leftarrow \ell(h, p)$; // MR: Compute loss via Eq. (12).
    Update $f_{\theta_\mathcal{T}}$ and f_{θ_1} ; // MR: Update attack model
return f_{θ_1}, f_{θ_2}, h_ϕ

L_1-norm distributions of the training samples they can access.

The attacker randomly selects M data points from D_k^{ir} to construct a distribution vector, uses it to query the attack model, and obtains the predicted fraction. To enhance the robustness and accuracy of property inference, we generate Q distribution vectors and use the average of their predicted fractions as the final result. When multiple types of intermediate results are accessed, the attacker performs the above process independently for each type. This involves obtaining distribution vectors for all kinds of intermediate results and ultimately querying the ensemble model to achieve property inference.

B. Theoretical Analysis

Intuitively, enhancing the representational capabilities is advantageous for inference attacks related to the target property, as it enables an adversary to more clearly differentiate between property and non-property samples. We now analyze and demonstrate this point from a theoretical perspective. Following previous works [15], [16], we reduce exact property estimation into a binary classification task to distinguish two worlds. Then, a simplified PIA is to classify the following worlds:

World 0: The VFL has an intermediate-result distribution D_0 whose proportion of the target property is t_0.

World 1: The VFL has an intermediate-result distribution D_1 whose proportion of the target property is t_1.

We introduce *distinguishing accuracy* (*DAcc*) to measure how well an adversary can distinguish between D_0 and D_1. This metric signifies the probability of an adversary accurately discerning between two distributions when it can access two datasets randomly sampled from D_0 or D_1, respectively. Now, we provide an upper bound of the distinguishing accuracy as follows.

Theorem 1 (The Upper Bound of Distinguishing Accuracy): Given that t^* represents the prior probability of a sample having the target property, q is the fraction difference between

two datasets, the distinguishing accuracy is bounded as:

$$DAcc \leq \frac{1+\sqrt{1-\left(1+\frac{q(c-t^*)}{t^*(1-c)}\right)^{-N}}}{2}, \tag{3}$$

where $c = \Pr[p = 1|z]$ denotes the conditional probability of an observed intermediate result owning the target property, and N denotes the size of an intermediate-result dataset.

Proof: Let Z denote a particular type of intermediate results, i.e., $Z \in \{\text{AO}, \text{AG}, \text{VO}, \text{VG}\}$ and $z \in Z$ as one of its elements. According to the corresponding probability density function, two distributions D_0 and D_1 can be expressed as follows:

$$\begin{aligned} D_0 &: \rho_0 = t_0 \Pr[z|p=1] + (1-t_0)\Pr[z|p=0], \\ D_1 &: \rho_1 = t_1 \Pr[z|p=1] + (1-t_1)\Pr[z|p=0]. \end{aligned} \tag{4}$$

Without loss of generality, we consider the case when $q = t_0 - t_1$. Then, given $t^* = \Pr[p = 1]$, and according to Eq.(4), we have:

$$\begin{aligned} \rho_0 &= (t_1+q)\frac{\Pr[z]\Pr[p=1|z]}{\Pr[p=1]} + (1-t_1-q)\frac{\Pr[z]\Pr[p=0|z]}{\Pr[p=0]} \\ &= (t_1+q)\frac{\Pr[z]c}{t^*} + (1-t_1-q)\frac{\Pr[z](1-c)}{1-t^*}. \end{aligned} \tag{5}$$

Similarly,

$$\rho_1 = t_1\frac{\Pr[z]c}{t^*} + (1-t_1)\frac{\Pr[z](1-c)}{1-t^*}. \tag{6}$$

Combining Eq.(5) and Eq.(6), we can derive:

$$\begin{aligned} \frac{\rho_0}{\rho_1} &= 1 + \frac{q(c-t^*)}{t_1(c-t^*)+t^*(1-c)} \\ &\leq 1 + \frac{q(c-t^*)}{t^*(1-c)}. \end{aligned} \tag{7}$$

The KL-divergence distance d_{KL} between ρ_0 and ρ_1 can be written as

$$\begin{aligned} d_{KL}(\rho_0 \| \rho_1) &= \int \rho_0(z)\log\left(\frac{\rho_0(z)}{\rho_1(z)}\right)\mathrm{d}z \\ &\leq \int \rho_0(z)\log\left(1+\frac{q(c-t^*)}{t^*(1-c)}\right)\mathrm{d}z \\ &= \log\left(1+\frac{q(c-t^*)}{t^*(1-c)}\right)\int \rho_0(z)\,\mathrm{d}z \\ &= \log\left(1+\frac{q(c-t^*)}{t^*(1-c)}\right). \end{aligned} \tag{8}$$

Based on the derivation above and previous work [32], for two datasets D_0' and D_1' containing N data points from D_0 and D_1, respectively, we achieve that:

$$\begin{aligned} DAcc &\leq \frac{1+d_{TV}(D_0', D_1')}{2} \\ &\leq \frac{1+\sqrt{1-\mathrm{e}^{-d_{KL}(D_0'\|D_1')}}}{2} \\ &\leq \frac{1+\sqrt{1-\left(1+\frac{q(c-t^*)}{t^*(1-c)}\right)^{-N}}}{2}, \end{aligned} \tag{9}$$

where $d_{TV}(\cdot,\cdot)$ is the total variation distance between two probability measures.□

Based on it, we derive Corollary 1, which further indicates potential enhancement strategies by increasing the correlation between observed representations and the target property.

Corollary 1: The upper bound of the distinguishing accuracy is a monotone increasing function about the posterior probability of an observed representation owning the target property.

Proof: Let $f(x) = \sqrt{1-(1+x)^{-N}}$, where $x = \frac{q(c-t^*)}{t^*(1-c)}$. Given that t^* is a constant and q is fixed, it is clear that as x increases, the function $f(x)$ also increases. We find the first derivative of x with respect to c as follows:

$$\frac{\partial x}{\partial c} = \frac{qt^*(1-t^*)}{[t^*(1-c)]^2} > 0. \tag{10}$$

Consequently, x monotonically increases with increasing c, resulting in the increase of $f(x)$.□

Theorem 1 presents an upper bound on the attack power of an adversary who seeks to distinguish between two distributions, while Corollary 1 points out that the bound arises with the conditional probability of an observed representation owning the target property. It implies avenues for improving the attack performance by controlling the correlation between observed representations and target property. Motivated by these insights, we present two enhancement approaches in the following module.

C. Correlation Augmentation

Based on the theoretical evidence, we introduce label replacement (LR) and model refinement (MR) in this module for two adversarial cases C1 and C2. The core idea is that an adversary transforms intermediate results into highly indicative representations of the target property and intentionally amplifies the disparity between the distributions of intermediate results.

1) Label Replacement: Inspired by existing poisoning attacks [15], [16], label replacement involves an attacker substituting the original task label and thus infusing supervised property information in the entire VFL system. Specifically, an adversary acted by the active party, e.g., P_k, possesses the capability to control label information. Before the VFL training, it secretly replaces the task label y_i with the property label p_i for each sample $\mathbf{x}_i^k$ in auxiliary data, where $p_i = 1$ if it belongs to a property sample, otherwise $p_i = 0$. Subsequently, the VFL system undergoes supervised learning about the target property, intensifying the association between intermediate results and the target property.

This method does not need access to the top model's training since the label manipulation can be carried out independently. Yet, only the active party with auxiliary data is empowered to use this enhancement. In addition, since most VFL systems are developed on extensive datasets, poisoning a small fraction of training data has a negligible impact on overall performance as demonstrated in our experiments later.

2) Model Refinement: Another approach to enhance the effectiveness of PIAs is to induce the adversary's bottom model to leak more property information through supervised learning and output constraints.

During the training process, the adversary controls its bottom model entirely. It at first adds randomly initialized layers on top of the bottom model, and refines the bottom model by learning supervised knowledge from auxiliary data. Formally, an adversarial party P_k introduces a prediction layer $f_{\theta_\mathcal{T}}$ with parameters $\theta_\mathcal{T}$. Using the intermediate output of its bottom model as input, $f_{\theta_\mathcal{T}}$ is used to make decisions for a binary property classification task. Then, the supervised learning objective on the adversary party's side is to minimize:

$$\mathcal{L}_s = \frac{1}{|D_{\text{aux}}|} \sum_{(\mathbf{x}_i^k, p_i) \in D_{\text{aux}}} \ell \left(f_{\theta_\mathcal{T}} \left(f_{\theta_k} \left(\mathbf{x}_i^k \right) \right), p_i \right), \tag{11}$$

where ℓ denotes as a loss function. By optimizing $\mathcal{L}_s$, more information related to the target properties is exposed to the adversarial party.

Besides supervised property information, the difference between intermediate results of property and non-property samples is another factor for a successful PIA. Intuitively, a wider distribution gap between property and non-property samples results in easier property inference. To exaggerate this distinction, we propose an output loss function for updating the adversary's bottom model as follows:

$$\mathcal{L}_o = -\frac{1}{|D_{\text{aux}}|} \sum_{(\mathbf{x}_i^k, p_i) \in D_{\text{aux}}} p_i' \left\| f_{\theta_k} \left(\mathbf{x}_i^k \right) \right\|_1, \tag{12}$$

where $\|\cdot\|_1$ is the L_1-norm function and $p_i' = -1$ for a non-property sample, and 0 otherwise. $\mathcal{L}_o$ increases the norm value of property samples and suppresses the norm value of non-property samples, making their intermediate output distributions more dispersed.

Finally, by combining the aforementioned two loss functions, the attacker aims to minimize the final objective function as follows:

$$\mathcal{L} = \mathcal{L}_s + \lambda \cdot \mathcal{L}_o, \tag{13}$$

where λ is a hyperparameter for balancing two terms. The attacker's bottom model is updated along with the optimization process. Conventional optimization techniques like stochastic gradient descent can be applied to solve the first term $\mathcal{L}_s$. Regarding the second term $\mathcal{L}_o$, we follow previous work [45] to optimize it and then update the adversary's bottom model.

In summary, both LR and MR approaches benefit from supervised learning, enhancing the representative capacity of intermediate results regarding the target property. MR goes further by making distribution gaps more distinguishing by restricting the attacker's bottom model outputs. Additionally, MR operates across different adversary's capacities, allowing arbitrary parties in VFL, whether passive or active, to enhance attack performance with the help of auxiliary data. Therefore, for C1, we apply both MR and LR approaches simultaneously, while for the C2 scenario, we utilize MR solely. We present the complete attack pipeline involving the three modules in Algorithm 2.

VI. Experiment

A. Experimental Setting

1) Datasets: As most of the datasets used in VFL research are tabular type [7], we evaluate our attack on five popular tabular datasets as follows:

1) *Adult* [46], [47]: The Adult dataset comprises 44,355 records of US census data. Following the exclusion of *final_weight* and duplicate entries, we frame a classification task aimed at predicting whether an individual's salary exceeds \$50,000. Within this dataset, we use *sex*, *race*, and *workclass* as private attributes vulnerable to potential inference attacks.
2) *Census* [16], [46]: As a repository of US census information, the Census dataset offers greater depth than the Adult dataset, comprising 299,285 records with 41 features. Similar to the Adult dataset, we utilize a binary classifier trained on the Census data to predict whether an individual's income exceeds \$50,000 annually.
3) *Bankmk* [16], [46]: The Bankmk dataset, also known as Bank Marketing, comprises 45,211 records detailing a banking institution's marketing campaigns. A binary classification model is trained on this dataset to forecast whether a client has subscribed to a term deposit. In this analysis, we designate *month*, *marital*, and *contact* as private attributes.
4) *Health*: The Health dataset, called Health-Heritage, encompasses over 60,000 medical records. In line with prior research [19], we omit the *Year* and *MemberID* attributes before utilizing it to train a binary classifier aimed at predicting *max_CharIsonIndex*. In this context, we consider *Sex* and *AgeAtFirstClaim* as the target properties vulnerable to inference by potential adversaries.
5) *Lawschool*[1] [48], [49]: The Lawschool dataset, sourced from the Law School Admissions Council, comprises 96,584 records detailing various attributes of law students, including gender, LAST score, and GPA, among others. It constitutes a binary classification task to predict whether a student will secure admission. Within this dataset, we designate *race*, *resident*, and *gender* as attributes susceptible to inference.
6) *Texas* [50]: The Texas hospital dataset consists of inpatient stay records collected from multiple healthcare facilities, based on the Hospital Discharge Data from 2006 to 2009. It contains over 67,000 samples, each represented by 6,169 binary features, and covers 100 output classes. The dataset is characterized by its high dimensionality and extreme sparsity. In our experiments, we consider two different attributes from this dataset as target properties.

To simulate the VFL scenario, we evenly distribute the feature space of training data to different parties by default. Table III presents the statistic information and model utility of various datasets in VFL.

2) Selection of Target Properties: Our experiments include sixteen target properties across six datasets, all intrinsically

[1] https://www.kaggle.com/datasets/danofer/law-school-admissions-bar-passage

TABLE III
DATASET DESCRIPTION AND PERFORMANCE

Dataset	Feature	Encoded	Class	Size	AUC
Adult	14	111	2	44,355	0.9039
Census	41	511	2	299,285	0.9418
Bankmk	16	51	2	45,211	0.9134
Health	17	110	2	61,700	0.9994
Lawschool	7	39	2	96,584	0.9376
Texas	6169	6169	1000	67,330	0.9544

TABLE IV
TARGET PROPERTIES CONSIDERED

Dataset	Attribute	#Num	Target	Fraction	Abbr.
Adult	sex	3	Male	66.21%	A-sex
	race	5	White	84.27%	A-race
	workclass	9	Private	67.52%	A-work
Census	sex	2	Female	52.05%	C-sex
	race	5	Black	10.20%	C-race
	education	17	Bachelor	9.94%	C-edu
Bankmk	month	12	May	30.45%	B-month
	marital	3	Married	60.19%	B-mart
	contact	3	Telephone	6.43%	B-cont
Health	sex	3	Female	16.13%	H-sex
	age	10	80+	2.68%	H-age
Lawschool	race	4	Black	8.63%	L-race
	resident	2	0.0	69.08%	L-res
	gender	2	0.0	44.27%	L-sex
Texas	Unknown-1	2	1.0	55.08%	T-u1
	Unknown-2	2	1.0	75.48%	T-u2

private attributes such as gender, race, and age. Following previous studies [16], we comprehensively explore various fractions from small ratios (below 10%) to large ratios (above 50%), including binary and categorical attributes. Table IV provides each sensitive attribute, the number of optional values in the selected attribute, target property, true fraction, and corresponding abbreviations.

3) Baseline: We choose *AIA* as our main baseline where an adversary estimates property information by inferring the specific value of the sensitive attribute and summarizing all results [17], [51]. We adopt a three-layer neural network as the binary classifier fed into intermediate results and make decisions about the target property. The predicted fraction is the ratio between the number of the predicted target property and the scale of a training dataset. Additionally, we explore other approaches, such as using the original training samples as input in AIA and simulating shadow model-based PIA within the VFL setting [15], [16], [27].

4) Model Architecture: Both the top and bottom models in VFL are constructed with their own neural networks. They utilize ReLU as the activation function and SGD as the optimizer. Specifically, the top model incorporates Sigmoid as its final layer and cross-entropy loss as its loss function. The aggregation of all intermediate outputs from the bottom models serves as the input for the top model. In our experiments, we mainly consider a two-party VFL while discussing multi-party VFL settings in the ablation study. Besides, we use XGBoost [52] as the default regression model.

TABLE V
ATTACK PERFORMANCE (MAE) OF VARIOUS ATTACKS UNDER DIFFERENT ADVERSARY'S CAPACITIES

Property	C1			C2		
	AIA	ProVFL-B	ProVFL-M	AIA	ProVFL-B	ProVFL-M
A-sex	0.1623	0.0186	**0.0097**	0.0181	0.0164	**0.0084**
A-race	0.2693	0.0236	**0.0143**	0.0388	0.0209	**0.0130**
A-work	0.2516	**0.0276**	0.0280	0.1777	0.0483	**0.0240**
C-sex	0.2127	**0.0175**	0.0221	0.0365	0.0229	**0.0184**
C-race	0.1913	0.0310	**0.0246**	0.2156	0.0444	**0.0255**
C-edu	0.1024	0.0141	**0.0106**	0.2149	**0.0125**	0.0214
B-month	0.2263	**0.0287**	0.0362	0.2189	0.1135	**0.0364**
B-mart	0.1370	0.0372	**0.0204**	0.0995	0.0600	**0.0325**
B-cont	0.2566	0.1058	**0.0236**	0.4039	0.1967	**0.0571**
H-sex	0.2853	**0.0067**	0.0119	0.0221	**0.0148**	0.0235
H-age	0.3564	0.0290	**0.0204**	0.0961	0.0161	**0.0130**
L-race	0.2066	0.0449	**0.0306**	0.2145	0.0704	**0.0281**
L-res	0.1930	0.0858	**0.0448**	0.1135	0.0893	**0.0500**
L-sex	0.1599	0.0337	**0.0185**	0.0161	0.0269	**0.0120**
T-u1	0.0386	0.0415	**0.0238**	0.0253	0.0241	**0.0195**
T-u2	0.1268	0.0475	**0.0251**	0.1490	0.0802	**0.0572**

Bold/Underline: the first/second smallest MAE in each case.

5) Training and Attack Setting: For each dataset, we use 80% of data for training and the remaining 20% for testing in VFL. By default, we set D_{aux} with 2000 property samples and 2000 non-property samples (i.e., 1.3%-9.0% of original datasets). We configure the attack training set size $|R|$ to 200, the sampling size M to 2000, the fraction interval q to 1, and the number of distribution queries Q to 100. We perform PIAs in the penultimate epoch of the training process. All experiments are reported on the average results of ten trials, and the source code is available.[2]

6) Evaluation Metric: Following previous work [23], we adopt mean absolute error (MAE) as the criteria to evaluate the attack performance. It is the average absolute difference between ground-truth fractions and predicted values. A lower MAE value indicates stronger attack performance.

B. Experimental Results

We validate the effectiveness of ProVFL in two adversarial cases: when the active party is the PIA adversary (C1) and when one of passive parties acts the PIA adversary (C2) during the VFL training process. For simplicity, we denote passive attacks as **ProVFL-B** when the adversary solely employs DC to infer target properties and active attacks as **ProVFL-M** when the adversary maliciously adopts LR or MR to enhance the performance of DC.

1) Performance of Passive Property Inference Attacks Through Distribution Comparison: We first analyze the effectiveness of DC from the attack performance of ProVFL-B in both cases. Table V shows that ProVFL-B in C1 outperforms

[2]https://github.com/BaiLibl/ProVFL

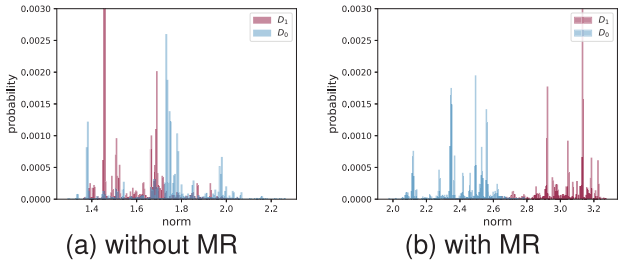


Fig. 4. L_1-norm distribution of AO between property (D_1) and non-property samples (D_0) on the A-sex property.

the baseline across all properties, significantly decreasing MAEs by nearly an order of magnitude (2.2× to 42.6×). Moreover, when employing AO solely in C2, ProVFL also achieves lower MAEs among most target properties. We also note that the baseline in C2 performs admirably, which results from the attack features used in the baseline coincidentally aligning well with the target property. Take L-sex as an example. The baseline can construct a high-quality classifier, achieving over 95% classification accuracy in practice. The performance difference can be attributed to inherent motivations: the baseline conducts inference at a record level, whereas our method performs property inference from a distribution-level perspective. Generally, it is easier to discern differences within an entire subset than within individual samples. Besides, when comparing ProVFL-B in different adversarial scenarios, the adversary in C1 often achieves better attack performance than in C2, especially in the Bankmk dataset. It demonstrates that DC effectively combines distribution information from different types of intermediate results and thus achieves low inference errors.

2) Performance of Active Property Inference Attacks Through Correlation Augmentation: When an adversarial party adopts enhancement approaches, the inference estimation errors can be further reduced. Table V shows that an adversary party can achieve lower estimation errors for most target properties when it uses LR and MR approaches in C1. And for an adversary in C2, solely using the MR strategy further decreases the inference errors of target properties by 1.2× to 3.4×. Although ProVFL-B performs better in some properties, e.g., C-edu and H-sex, their differences are less than 1% MAE. These results demonstrate that CA enhances the performance of ProVFL-B and lowers inference errors regardless of the adversary's role. We also observe that regarding target properties from Texas, although our proposed method still outperforms the baselines, the improvement of CA is less significant compared to other properties. This is partly because, for multi-class samples, we only apply MR to enhance the performance of ProVFL-B. Moreover, the number of task classes is substantially larger than the number of target attributes, resulting in limited supervised information being embedded in the intermediate representations. Besides, when comparing ProVFL-M in both cases, we observe that an adversary with less adversarial knowledge in C2 can sometimes achieve better attack performance, e.g., Adult dataset. We conjecture that this is because MR intentionally makes the attacker's AO more distinguishable and leads to significantly less overlap between their L_1-norm distributions, as shown in Figure 4. However, such a difference may be reduced when considering four types of intermediate results in C1.

TABLE VI

THE PERFORMANCE OF USING THE RAW FEATURES UNDER DIFFERENT LEVELS OF CORRELATIONS

Property	Attacks (MAE)		Correlation (PCC)	
	AIA*	ProVFL-B	p	$\mathcal{Y}$
B-month	0.1569	0.0287	0.4259	-0.1050
B-mart	0.0753	0.0372	0.2875	-0.0554
B-cont	0.2894	0.1058	0.1637	0.0131

TABLE VII

ATTACK PERFORMANCE (ACCURACY) BETWEEN SM-PIA AND PROVFL-B

Property	Values	SM-PIA	ProVFL-B
A-sex	30% vs. 50%	0.50	0.97
A-work	50% vs. 60%	0.55	0.88
A-race	5% vs. 10%	0.50	0.85

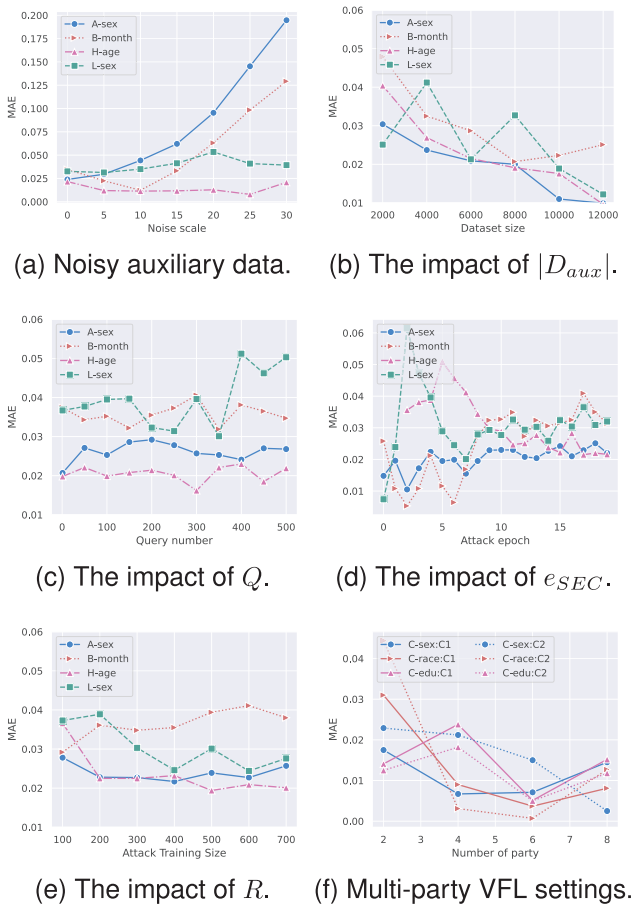
TABLE VIII

ATTACK PERFORMANCE (MAE) AND UTILITY (AUC) ON MISALIGNED AUXILIARY DATASETS

Property	(C1) ProVFL-B	(C2) ProVFL-B	Utility
A-sex	0.0091	0.0125	0.9016
A-race	0.0178	0.0247	0.9017
A-work	0.0661	0.0537	0.9016

3) Other Baselines: Apart from the AIA baseline above, we additionally compare two possible methods with our ProVFL. 1) Using raw features rather intermediate results in AIA (named *AIA**): The raw features on the adversarial party's side can be used to deduce property information due to inherent correlation. Similar to the baseline, we use these features as the inputs of the binary classifier. Table VI presents the attack performance, maximum correlation with the target property in adversarial party's features, and correlation with task labels using Pearson correlation coefficient (PCC). Although there is a high correlation, the AIA-based approach remains ineffective for property inference on the Bankmk dataset, e.g., B-month. In contrast, ProVFL demonstrates robustness by not relying heavily on correlations with the task label or other attributes, e.g., B-cont. 2) Shadow model-based PIA (named *SM-PIA*): We attempt shadow model training methods [15], [16], [27] by replicating the complete training process across the entire feature space. We adapt it and ProVFL-B to distinguish two predefined fraction values as shown in Table VII. As expected, ProVFL-B effectively handles both distributions regardless of the fraction gap, whereas SM-PIA fails to differentiate between them. In fact, the SM-PIA method assumes that an adversary has access to the full feature space, which is impractical for a party limited to its own feature subset in VFL.

4) Relaxed Assumptions on Auxiliary Data: Our above attacks are conducted under the assumption that auxiliary data



(a) Noisy auxiliary data. (b) The impact of $|D_{aux}|$.

(c) The impact of Q. (d) The impact of e_{SEC}.

(e) The impact of R. (f) Multi-party VFL settings.

Fig. 5. Impact of various settings in ProVFL-B.

TABLE IX
ATTACK PERFORMANCE (MAE) UNDER DIFFERENT CA STRATEGIES

Property	w/o MR	w/o LR	with MR+LR
A-sex	0.0137/0.8917	0.0144/0.9009	0.0097/0.8931
A-race	0.0165/0.8950	0.0318/0.9017	0.0143/0.8960
A-work	0.0340/0.8907	0.0158/0.9009	0.0280/0.8907
B-month	0.0473/0.9129	0.0193/0.9057	0.0362/0.9047
B-mart	0.0226/0.9108	0.0313/0.8440	0.0204/0.7850
B-cont	0.0492/0.9045	0.1027/0.8840	0.0236/0.8360

/ means attack performance (MEA)/model utility (AUC).

TABLE X
AVERAGE TIME COST OF VARIOUS ATTACKS ON ADULT DATASET

Attack	AIA	DC (ProVFL-B)	DC+MR	DC+MR+LR (ProVFL-M)
Time (s)	31.8	4.4	17.2	17.2

TABLE XI
ATTACK PERFORMANCE (MAE) OF PROVFL-B AND AIA ON THE NUS-WIDE DATASET FOR DIFFERENT TARGET PROPERTIES

Adversary Access	Method	Target Property		
		Grass	Animal	Water
Text Data	AIA	0.2997	0.1491	0.1853
	ProVFL-B	**0.0834**	**0.0324**	**0.0615**
Image Data	AIA	0.1814	0.1830	0.2120
	ProVFL-B	**0.0450**	**0.0273**	**0.0795**

are clean and aligned with other parties' samples. We now relax this assumption and discuss two general cases: **1)** *Noisy auxiliary data:* an adversary may use noisy samples to conduct ProVFL, e.g., by using feature inference attacks [8], [9], [53] to recover a handful of the victim's records. As shown in Figure 5a, we observe that for A-sex and H-age properties, the inference errors increase as the noise scale rises, but as for the other two properties, their attack performance remains relatively stable. Overall, when 10% auxiliary samples are mislabeled, the inference errors increase by 2%. It indicates that ProVFL is robust for noisy auxiliary data and maintains similar attack performance. **2)** *Misaligned auxiliary data:* An adversary cannot observe intermediate results during training if their sample IDs do not overlap with those of VFL parties. To simulate this scenario, we randomly replace the adversary's aligned training data with these samples prior to the VFL process. Table VIII demonstrates that ProVFL can successfully infer property information without significant performance deterioration in this scenario. The above results suggest that ProVFL remains robust and effective across different types of auxiliary data.

5) Performance and Time Cost Analysis of LR and MR Approaches: Table IX illustrates the attack performance and model utility of different enhancement approaches in C1. This suggests that neither LR nor MR plays a dominant role in CA, yet utilizing both consistently enhances attack performance without much loss of model utility. Additionally, while CA introduces adjustments to VFL, it does not significantly increase time costs, as shown in Table X. Vanilla DC has a lower computational overhead than the baseline, owing to its reliance on a regression model with small training costs. Regarding the CA module, LR incurs negligible additional computational overhead for VFL training. Although MR would require more time for the optimization process, the total cost is still lower than the baseline.

6) Extension to Other Data Types: To validate the board generalizability of our proposed method, we conduct experiments on the multi-label image-text dataset NUS-WIDE [54], which contains 634 low-level image features and 1,000 textual tag features. Each type of feature is assigned to a separate client. We focus on the top-5 most frequent labels in our experiments. Unlike tabular data with structured feature columns, image and text data are inherently high-dimensional and unstructured. Therefore, we designate one type of label as the target property and use the remaining labels as the learning objectives for the VFL models. We perform our attacks on both feature types using ProVFL-B in C2, with the adversary holding image and text features, respectively. The results in Table XI further demonstrate the effectiveness of our proposed method across different data modalities.

TABLE XII
IMPACT OF DIFFERENT Q ON ATTACK PERFORMANCE (MAE)

Property	0.1	1	5	10
A-sex	0.0257	0.0186	0.0232	0.0239
A-race	0.0250	0.0236	0.0264	0.0287
A-work	0.0374	0.0276	0.0315	0.0355

TABLE XIII
ATTACK PERFORMANCE (MAE) AMONG REGRESSION MODELS

Property	XGBoost	DTR	LNR	SVR
A-sex	0.0175	0.0228	0.0223	0.0197
A-race	0.0229	0.0297	0.0243	0.0508
A-work	0.0231	0.0250	0.0258	0.0459

TABLE XIV
IMPACT OF λ ON ATTACK PERFORMANCE (MAE) AND UTILITY (AUC)

λ	0.0	0.5	1.0	2.0
B-month	0.1135	0.0300	0.0364	0.0205
B-mart	0.0600	0.0389	0.0325	0.0297
B-cont	0.1967	0.0545	0.0571	0.0323
Utility	0.9134	0.8924	0.8779	0.8270

TABLE XV
IMPACT OF NON-IID VFL SETTING ON ATTACK PERFORMANCE (MAE)

r	0.1	0.3	0.5	0.7	0.9
C-sex	0.0206	0.0141	0.0175	0.0273	0.0575
C-race	0.0552	0.0488	0.0310	0.0473	0.1169
C-edu	0.0275	0.0133	0.0141	0.0147	0.0330

C. Ablation Study

In this section, we investigate the effects of different hyperparameter settings on attack performance, including the auxiliary dataset size $|D_{aux}|$, the number of distribution query Q, the attack epoch e_{SEC}, the attack training set size R, the fraction interval q, the architecture of regression models, and multi-party VFL settings. Considering the active party as the adversary, we delve into them in ProVFL-B across four target properties with distinct fractions, ranging from a larger ratio (e.g., A-sex: 66.21%) to a smaller one (e.g., H-age: 2.68%).

1) The Impact of Different $|D_{Aux}|$: DC depends on an auxiliary dataset to prepare different populations of intermediate results. Consequently, we examine it to discern its impact on the attack's effectiveness by varying the size of the auxiliary dataset, as illustrated in Figure 5b. We observe that increasing the number of auxiliary samples improves the attack performance since the attack model can more effectively capture the differences across property distributions with more auxiliary samples.

2) The Impact of Different Q: We examine the effect on estimation errors by adjusting the number of distribution queries employed for inferring property information. Likewise, we carry out our ablation study on different levels of target properties in Figure 5c. In the beginning, as more queries are conducted, the attack performance improves by reducing the influence of outlier distributions randomly sampled from the training set. However, this improvement diminishes with further increases in the number of queries.

3) The Impact of Different e_{SEC}: We now explore the impact on attack performance by altering the epoch at which an adversary executes a PIA, as depicted in Figure 5d. We find a noticeable fluctuation when adversaries use intermediate outputs or gradients in the first several epochs. Nevertheless, the attack performance stabilizes as the attack epoch progresses. For instance, the MAE difference between two separate epochs is approximately 0.01 after the 10th epoch. As the training epoch progresses, intermediate gradients and intermediate outputs demonstrate stability as the model converges, especially in the later stages of the training process. Hence, we suggest adversaries conduct an inference attack during the late epochs instead of at the beginning of training.

4) The Impact of Different R: We fix the size of the auxiliary dataset at 4000 and vary the size of the attack training set from 100 to 500. Figure 5e plots the corresponding attack performance. Upon surpassing a set size of 400, the improvement in attack performance becomes minimal. The estimation errors among target properties fluctuate within 0.01.

5) The Impact of Different q: We adopt the default setting of ProVFL-B in experiment settings but vary the interval of the fraction candidate set. Table XII shows the attack performance under different q settings. When the fraction interval is 1, we can construct a distribution of appropriate granularity, achieving the lowest inference error.

6) The Impact of Regression Models: Our previous experiments employ an XGBoost as the attack model to infer property information. Now, we explore others to understand the effect of different regression models, including decision tree regression (DTR) [55], linear regression (LNR) [56], and support vector regression (SVR) [57]. Table XIII presents the performance of three regression models on the Adult dataset. It indicates that the non-linear XGBoost and DTR can more effectively capture the alignment of L_1-norm distribution and property fractions.

7) The Impact of Different λ: We explore the changes in the attack performance and VFL utility under different λ settings when using MR strategy, as presented in Table XIV. Upon introducing the MR module, our attacks demonstrate much lower estimation errors on the Bankmk dataset. Although a larger λ setting tends to improve attack performance, an attacker has to balance the trade-off between property leakage and model utility.

8) The Impact of Non-IID VFL Setting: The experiments presented above were conducted under an even split of the feature space among parties. We now extend our investigation to non-IID feature distributions [58] and examine their impact on the proposed methods. Specifically, we consider scenarios where the feature space is unevenly partitioned, with different parties holding varying numbers of features. We define r as the ratio of the number of features held by the victim party to that of the other party. Table XV shows that non-IID data in

the feature space may reduce the property leakage, particularly when the victim party holds a large number of features.

9) The Impact of Multi-Party VFL Setting: We present the attack performance in the multi-party VFL setting on the Census dataset due to its extensive feature space. Each party is allocated an equal portion of feature space. As Figure 5f shows, we present inference results on three target properties in C1 and C2. We observe that estimation errors become lower as the number of parties increases. This observation aligns with an intuition that, with a fixed feature space, more parties result in a smaller partial feature space, and the target property then becomes more dominant in the victim party's feature space.

VII. Defense

In this section, we design several mechanisms to mitigate property leakage during the VFL training process, including common strategies as well as customized mitigation techniques specifically against our framework. We validate the defense performance against ProVFL-B in C1.

A. D1. Add Noises to Gradient

A typical defensive strategy to mitigate information leakage in FL is adding noise into gradients [3], [20], [27]. In a vertical case, a trusted third party can add Laplacian noise to intermediate gradients before sending them to parties. We set the noise scale as the hyperparameter to adjust the effectiveness of this defense.

B. D2. Adopt Privacy-Preserving Deep Learning

Following [3], we adopt a hybrid defense framework involving differential privacy [59], [60], [61] and gradient compression to mitigate property leakage. The r_G fraction of intermediate gradients are processed in the training process. A defender can vary r_G to trade-off between defense performance and model utility.

C. D3. Adopt Randomized Responses in Both Propagations

A recent state-of-the-art defense [31] for Split learning is proposed to disrupt knowledge transfer in both directions through a flexible, unified solution. It introduces a newly designed activation function, named R^3eLU, that transforms private smashed data in the forward pass and partial loss in the backward pass into randomized responses, respectively. We consider the privacy budget ϵ as the hyperparameter in our reproduction.

D. D4. Shuffle the Training Data

The correlation between intermediate results and a target property is a key factor of information leakage. A possible defense is to disrupt such a correlation by shuffling the victim's training data. Specifically, the victim party randomly disorders some of his training data, which causes misalignment between its property information and the attacker's intermediate results. We treat the shuffling ratio of this mitigation as a hyperparameter.

TABLE XVI
Defense Performance Against ProVFL-B on Adult Dataset: MAE vs. AUC Trade-off

	Setting	Value	A-sex	A-race	A-work	AUC
None	-	-	0.0186	0.0236	0.0276	0.9039
D1	Noise scale	10e-3	0.0219	0.0221	0.0370	0.9028
		10e-2	0.0239	0.0293	0.0407	0.9027
		10e-1	0.0185	0.0315	0.0499	0.8353
D2	r_G	0.50	0.0255	0.0349	0.0400	0.9009
		0.25	0.0219	0.0422	0.0410	0.8318
		0.10	0.017	0.0542	0.0428	0.7983
D3	ϵ	1.0	0.2154	0.3528	0.1605	0.5229
		4.0	0.1934	0.3355	0.1485	0.6303
		10.0	0.2058	0.3335	0.1248	0.6634
D4	Shuffling ratio	0.10	0.0211	0.0290	0.0387	0.8989
		0.20	0.0135	0.0383	0.0453	0.8948
		0.50	0.0083	0.0459	0.0351	0.8785
D5	Withdrawal ratio	0.10	0.0230	0.0281	0.0535	0.9014
		0.20	0.0617	0.0655	0.0828	0.9011
		0.50	0.1479	0.1345	0.1617	0.8979

TABLE XVII
Defense Performance Against ProVFL-B on Texas Dataset: MAE vs. Accuracy Trade-off

	Setting	Value	T-u1	T-u2	ACC
None	-	-	0.0415	0.0475	0.5607
D5	Withdrawal ratio	0.10	0.0777	0.0840	0.5480
		0.20	0.0717	0.1294	0.5458
		0.50	0.1276	0.1591	0.5511

E. D5. Withdraw Aligned Data Before Training

This defense indirectly changes the training data accessed by an adversarial party. After data index alignment, we can process intersecting features to hide the true statistic information of a sensitive attribute. To achieve this, the victim party deliberately withdraws part of intersecting samples containing the target property and thus changes its distribution in the training process. We consider different withdrawal ratios to validate its effectiveness.

We evaluate the above five mitigation mechanisms to defend our proposed method, where an attacker utilizes his bottom model and four kinds of intermediate results for property inference. Table XVI presents the defense performance across various configurations. D1 and D2 offer slight mitigation against property leakage, accompanied by a clear degradation in model utility. Although D3 significantly reduces property leakage by simultaneously perturbing all intermediate results, it results in substantial utility loss across various target properties. Furthermore, such defense becomes less effective or even saturated as ϵ increases. Additionally, D4 proves ineffective in safeguarding against ProVFL, as shuffling merely disrupts alignment with the victim's data without altering the overall distribution of properties in the training set. As for D5, it diminishes the effectiveness of ProVFL with a small impact on model utility by removing property samples from aligned

data, and thus alters the original property distribution. To further validate the defensive performance of D5, we evaluate it on a multi-classification task using the Texas dataset. As shown in Table XVII, D5 effectively mitigates the performance of ProVFL-B while incurring minimal loss in classification accuracy.

This adjustment significantly weakens the adversary's ability to infer property information, offering a straightforward yet effective defense to safeguard sensitive private information.

VIII. Conclusion

VFL is a widely used framework in various applications, but its distribution-level privacy leakage has not been thoroughly explored. In this paper, we reveal the risk of property leakage in VFL by introducing ProVFL, a novel framework designed to perform property inference with low estimation errors. ProVFL is based on the empirical observation that the L_p-norm distribution of intermediate results unintentionally leaks private property information to potential adversarial parties. To address this issue, we first design a distribution comparison method for property inference and analyze its effectiveness from a theoretical standpoint. We further propose correlation augmentation to enhance the leakage of property information. Our experimental results demonstrate that ProVFL performs effectively across various target properties, highlighting how shared intermediate results in VFL can easily expose private information. Additionally, we explore existing defenses against such leakage and propose a simple yet effective mitigation approach for our attacks. We hope this work provides valuable insights into the protection of private statistical information in VFL.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54, 2017, pp. 1273–1282.

[2] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "CAFE: Catastrophic data leakage in vertical federated learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 994–1006.

[3] C. Fu et al., "Label inference attacks against vertical federated learning," in *Proc. 31st USENIX Security Symp. (USENIX Security)*, 2022, pp. 1397–1414.

[4] P. Chen, X. Du, Z. Lu, J. Wu, and P. C. K. Hung, "EVFL: An explainable vertical federated learning for data-oriented artificial intelligence systems," *J. Syst. Archit.*, vol. 126, May 2022, Art. no. 102474.

[5] J. Zhang and Y. Jiang, "A vertical federation recommendation method based on clustering and latent factor model," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci. (EIECS)*, Sep. 2021, pp. 362–366.

[6] T. Chen, X. Jin, Y. Sun, and W. Yin, "VAFL: A method of vertical asynchronous federated learning," 2020, *arXiv:2007.06081*.

[7] Y. Liu et al., "Vertical federated learning: Concepts, advances, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3615–3634, Jul. 2024.

[8] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, "Feature inference attack on model predictions in vertical federated learning," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 181–192.

[9] P. Ye, Z. Jiang, W. Wang, B. Li, and B. Li, "Feature reconstruction attacks and countermeasures of DNN training in vertical federated learning," 2022, *arXiv:2210.06771*.

[10] Y. Hu et al., "Is vertical logistic regression privacy-preserving? A comprehensive privacy analysis and beyond," 2022, *arXiv:2207.09087*.

[11] O. Li et al., "Label leakage and protection in two-party split learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022, pp. 1–27.

[12] J. Sun, X. Yang, Y. Yao, and C. Wang, "Label leakage and protection from forward embedding in vertical federated learning," 2022, *arXiv:2203.01451*.

[13] S. Kariyappa and M. K. Qureshi, "ExPLoit: Extracting private labels in split learning," in *Proc. IEEE Conf. Secure Trustworthy Mach. Learn. (SaTML)*, Feb. 2023, pp. 165–175.

[14] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *Int. J. Secur. Netw.*, vol. 10, no. 3, pp. 137–150, 2015.

[15] S. Mahloujifar, E. Ghosh, and M. Chase, "Property inference from poisoning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 1120–1137.

[16] H. Chaudhari, J. Abascal, A. Oprea, M. Jagielski, F. Tramèr, and J. Ullman, "SNAP: Efficient extraction of private properties with poisoning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 400–417.

[17] X. Wang and W. H. Wang, "Group property inference attacks against graph neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 2871–2884.

[18] T. Castiglia, S. Wang, and S. Patterson, "Flexible vertical federated learning with heterogeneous parties," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 17878–17892, Dec. 2024.

[19] W. Zhang, S. Tople, and O. Ohrimenko, "Leakage of dataset properties in multi-party machine learning," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 2687–2704.

[20] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 691–706.

[21] C. Song and V. Shmatikov, "Overlearning reveals sensitive attributes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–12.

[22] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, "Property inference attacks on fully connected neural networks using permutation invariant representations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 619–633.

[23] J. Zhou, Y. Chen, C. Shen, and Y. Zhang, "Property inference attacks against GANs," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2022, pp. 1–18.

[24] H. Hu and J. Pang, "PriSampler: Mitigating property inference of diffusion models," 2023, *arXiv:2306.05208*.

[25] X. Luo, Y. Jiang, F. Wei, Y. Wu, X. Xiao, and B. C. Ooi, "Exploring privacy and fairness risks in sharing diffusion models: An adversarial perspective," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8109–8124, 2024.

[26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[27] Z. Wang, Y. Huang, M. Song, L. Wu, F. Xue, and K. Ren, "Poisoning-assisted property inference attack against federated learning," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 4, pp. 3328–3340, Jul./Aug. 2023.

[28] D. Pasquini, G. Ateniese, and M. Bernaschi, "Unleashing the tiger: Inference attacks on split learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 2113–2129.

[29] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[30] H. Yang, Y. Wang, and B. Li, "Individual property inference over collaborative learning in deep feature space," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[31] Y. Mao, Z. Xin, Z. Li, J. Hong, Q. Yang, and S. Zhong, "Secure split learning against property inference, data reconstruction, and feature space hijacking attacks," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)*, vol. 14347, 2023, pp. 23–43.

[32] A. Suri and D. Evans, "Formalizing and estimating distribution inference risks," *Proc. Privacy Enhancing Technol.*, vol. 2022, no. 4, pp. 528–551, 2022.

[33] P. Qiu, X. Zhang, S. Ji, C. Fu, X. Yang, and T. Wang, "HashVFL: Defending against data reconstruction attacks in vertical federated learning," 2022, *arXiv:2212.00325*.

[34] X. Jiang, X. Zhou, and J. Grossklags, "Comprehensive analysis of privacy leakage in vertical federated learning during prediction," *Proc. Privacy Enhancing Technol.*, vol. 2022, no. 2, pp. 263–281, 2022.

[35] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 148–162.

[36] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1426–1435.

[37] W. Wang, Y. Shi, S. Chen, Q. Peng, F. Zheng, and X. You, "Norm-guided adaptive visual embedding for zero-shot sketch-based image retrieval," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1106–1112.

[38] C. Sohler and D. P. Woodruff, "Subspace embeddings for the L_1-norm with applications," in *Proc. 43rd Annu. ACM Symp. Theory Comput. (STOC)*, 2011, pp. 755–764.

[39] D. Chen, S. Zhang, J. Yang, and B. Schiele, "Norm-aware embedding for efficient person search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12615–12624.

[40] Y. Guo and L. Zhang, "One-shot face recognition by promoting underrepresented classes," 2017, *arXiv:1707.05574*.

[41] Y. Wang et al., "Orthogonal deep features decomposition for age-invariant face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 738–753.

[42] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. Workshop Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–8.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[44] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[45] H. Chen et al., "Data-free learning of student networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3514–3522.

[46] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[47] R. Kohavi, "Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, vol. 96, 1996, pp. 202–207.

[48] L. F. Wightman, "LSAC national longitudinal bar passage study. LSAC research report series," ERIC, Tech. Rep., 1998.

[49] M. Vero, M. Balunović, D. I. Dimitrov, and M. Vechev, "TabLeak: Tabular data leakage in federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2023, pp. 35051–35083.

[50] T. D. of State Health Services. (2006). *Texas Hospital Inpatient Discharge Public Use Data File*. [Online]. Available: https://www.dshs.texas.gov/thcic/hospitals/Inpatientpudf.shtm

[51] C. Song and A. Raghunathan, "Information leakage in embedding models," in *Proc. 27th ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2020, pp. 377–390.

[52] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[53] H. Weng et al., "Practical privacy attacks on vertical federated learning," 2020, *arXiv:2011.09290*.

[54] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, Jul. 2009, pp. 1–9.

[55] W. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[56] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Mach. Learn.*, vol. 85, nos. 1–2, pp. 41–75, Oct. 2011.

[57] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[58] K. Jahani, B. Moshiri, and B. H. Khalaj, "A survey on data distribution challenges and solutions in vertical and horizontal federated learning," *J. Artif. Intell., Appl., Innov.*, vol. 1, no. 2, pp. 55–71, 2024.

[59] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1310–1321.

[60] L. Wang, Q. Ye, H. Hu, X. Meng, and K. Huang, "LDP-purifier: Defending against poisoning attacks in local differential privacy," in *Proc. Int. Conf. Database Syst. Adv. Appl. (DASFAA)* (Lecture Notes in Computer Science), vol. 14853. Springer, 2024, pp. 221–231.

[61] J. Cai, Q. Ye, H. Hu, X. Liu, and Y. Fu, "Boosting accuracy of differentially private continuous data release for federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 10287–10301, 2024.