PS-MI: Accurate, Efficient, and Private Data Valuation in Vertical Federated Learning

Xiaokai Zhou
School of Computer Science,
Wuhan University
xiaokaizhou@whu.edu.cn

Xiao Yan
Institute for Math & AI,
Wuhan University
yanxiaosunny@gmail.com

Fangcheng Fu
School of Artificial Intelligence,
Shanghai Jiao Tong University
ccchengff@gmail.com

Ziwen Fu
School of Cyber Science and
Engineering, Wuhan University
ziwen.fu@whu.edu.cn

Tieyun Qian
School of Computer Science,
Wuhan University
qty@whu.edu.cn

Yuanyuan Zhu
School of Computer Science,
Wuhan University
yyzhu@whu.edu.cn

Qinbo Zhang
School of Computer Science,
Wuhan University
qinbo_zhang@whu.edu.cn

Bin Cui
School of Computer Science,
Peking University
bin.cui@pku.edu.cn

Jiawei Jiang
School of Computer Science,
Wuhan University
jiawei.jiang@whu.edu.cn

ABSTRACT

Vertical federated learning (VFL) trains models when multiple databases (a.k.a participants) hold different features of the same set of samples. By quantifying each participant's contribution to model training, *data valuation* can prevent hitch-riders and reward the instrumental parties. However, vertical federated data valuation (VFDV) is challenging because it needs to be accurate and efficient while protecting participant data privacy. In this paper, we propose a method meeting all three requirements by using *projection* and *sampling* for *mutual information* estimation (thus dubbed PS-MI). In particular, we first show that the utility of a participant set (a.k.a a *coalition*) can be expressed as the mutual information (MI) between their features and the target labels. MI is favorable because it does not depend on the model to train (i.e., *model-agnostic*) and can be estimated via k-nearest neighbor (KNN). To run KNN, instead of using costly homomorphic encryption to protect data privacy, we apply simple *random projection* to participant features before distance computation. We prove that random projection ensures differential privacy and preserves unbiased distance estimates. Since the contribution of a participant involves many coalitions, we adopt *stratified sampling* to reduce the number of coalitions while controlling estimation variance. To further improve efficiency, we incorporate optimizations including using locality sensitive hashing (LSH) to prune kNN candidates, batching kNN candidate checking for multiple coalitions, and adaptive early termination for utility evaluation. We compare PS-MI with 5 state-of-the-art VFDV methods. The results show that PS-MI yields higher accuracy and shorter running time than the baselines, and the maximum speedup can be 592×.

PVLDB Reference Format:
Xiaokai Zhou, Xiao Yan, Fangcheng Fu, Ziwen Fu, Tieyun Qian, Yuanyuan Zhu, Qinbo Zhang, Bin Cui, and Jiawei Jiang. PS-MI: Accurate, Efficient, and Private Data Valuation in Vertical Federated Learning. PVLDB, 18(10): 3559 - 3572, 2025.
doi:10.14778/3748191.3748215

PVLDB Artifact Availability:
The source code is available at https://github.com/ZhouXiaokay/VF-SV.

1 INTRODUCTION

"*Data is the new oil*" — large-scale, high-quality training data is the foundation of performant machine learning (ML) models. However, in many practical scenarios, the training data is distributed across multiple parties and cannot be shared due to regulatory restrictions [49, 52]. To tackle this problem, federated learning (FL) coordinates multiple parties to collaboratively train ML models while protecting data privacy. Based on the distribution of training data, FL can be categorized into two main types, i.e., *horizontal federated learning* (HFL) and *vertical federated learning* (VFL). In HFL, different parties hold different data samples but share the same feature space, while in VFL, all parties have the same set of data samples but hold different features. In this paper, we focus on VFL since it has attracted research interests from the database community on topics such as efficient training [14, 34, 57], data privacy protection [15, 36, 59], and communication optimization [13, 17, 35].

Vertical Federated Data Valuation (VFDV). In VFL, some parties may hold features that are informative for the model predictions (e.g., the classification labels) while the other parties may not. VFDV quantifies the contribution of (the data from) each party to model training and can serve multiple purposes. ❶ It can protect against hitch-riders or malicious attackers, which contribute low-quality or irrelevant data. ❷ It allows to give quantitative rewards to the parties according to their contributions such that instrumental parties are encouraged to participate. ❸ We can select only the instrumental parties for model training to improve efficiency.

Existing Solutions and Their Limitations. Early researches use heuristic methods for VFDV [22, 24, 31]. They cannot model the

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 18, No. 10 ISSN 2150-8097.
doi:10.14778/3748191.3748215

marginal contribution of a party w.r.t. the other parties and fail to ensure fairness when evaluating the contributions of different parties. In contrast, derived from cooperative game [48], the *Shapley value* (SV) is widely recognized as a fair and principled metric of contribution evaluation. In particular, the SV of a participant (a.k.a party) is defined as its average marginal contribution over all possible subsets of the other participants. Formally, consider a set $\mathcal{P}$ with P participants, and use $v(S)$ to denote the utility a subset of participants $S \subseteq P$, the SV $s(p)$ of participant p is

$$s(p) = \frac{1}{P} \sum_{\mathcal{S} \subseteq \mathcal{P} \backslash p} \binom{P-1}{|\mathcal{S}|}^{-1} \left[v(\mathcal{S} \cup \{p\}) - v(\mathcal{S}) \right]. \quad (1)$$

As a contribution evaluation metric, the SV satisfies favorable axioms such as balance, additivity, symmetry, and zero element, and thus it has been widely used in the database community [6, 23, 63, 64, 66]. However, adopting SV for VFDV poses three challenges:

- *Accuracy*. Using model performance (e.g., test accuracy) as the utility function leads to inconsistent SV estimates across models. As a result, data valuation must be repeated whenever the model changes, which is common in practice for tasks like performance tuning. For example, as shown in Table 1, the SV correlation between logistic regression (LR) and multilayer perceptron (MLP) is only 0.45, and using LR-based SV for MLP participant selection degrades accuracy from 0.8070 to 0.7893.
- *Privacy*. VFL requires to protect participants' data privacy but data valuation needs to exchange information about participant data. This opens the door for malicious participants to infer or steal private data from others through received messages.
- *Efficiency*. Computing SV requires evaluating $2^P - 1$ coalitions for P participants, which is computationally intensive. As shown in Table 1, even with only 4 participants, training a model for evaluating each coalition leads to a long running time.

Multiple techniques are proposed to solve the above concerns. For the first challenge, some works use mutual information as the utility function [22, 24, 31], but these ignore marginal contributions and fail to ensure fairness among participants. For the second challenge, homomorphic encryption (HE) is employed to protect transmitted data [22, 24, 55], but it relies on a trusted third party, cannot defend against collusion, and incurs high costs. For the third challenge, methods such as Monte Carlo sampling [18], Hessian approximation [54], and transfer learning [67] are used to reduce complexity, yet they sacrifice precision and still require repeated evaluations. However, none of them considers accuracy, privacy, and efficiency simultaneously. These limitations lead us to ask:

> *Can we accurately, securely, and efficiently compute the Shapley value for each participant in VFL?*

Our Solutions PS-MI. To this end, we propose a vertical federated data valuation method PS-MI, which achieves estimation accuracy, execution efficiency, and data privacy at the same time.

❶ **Mutual Information (MI) as Model Agnostic Utility.** For accurate data valuation, we first define a general utility function that quantifies the contribution of a coalition of participants as the predictive power of their features for the downstream task. Then, we show that the utility function can be naturally expressed as the Shannon mutual information (MI) between the features of the participants and the target labels for the classification task when the loss function is cross-entropy. The MI utility function is model-agnostic and does not require model training.

Table 1: Consistency of Shapley value (SV) estimations and model accuracy on *Bank* with 4 participants. (i) Pearson correlation coefficient (PCC) between the SVs estimated using the test accuracy of Logistic Regression (LR) and Multilayer Perceptron (MLP); (ii) Model accuracy of LR/MLP when trained with top-1 participant selected according to SVs; (iii) Computation time, *Single* for one coalition, *Total* for all coalitions.

Proxy	PCC	Acc		Time(s)	
		LR	MLP	Single	Total
LR	0.4516	**0.7893**	0.7893	1930	28950
MLP		0.7740	**0.8070**	2100	31500

❷ **Projection-based MI Estimator.** The MI utility can be estimated by searching the k-nearest neighbors (KNNs) for data samples. To run KNN efficiently, we apply random projection to transform the local features of the participants and add Gaussian noises to the transformed features before distance computation. This improves efficiency by avoiding HE since directly exchanging the transformed distances still satisfies the well-known (ϵ, δ) – differential privacy. Moreover, we also show that this provides accurate MI estimations because the transformed distances match the original distances in expectation.

❸ **Efficiency Optimizations**. To further enhance efficiency, we consider two key aspects, i.e., reducing the number of evaluated coalitions and reducing the per-coalition evaluation cost.

- *Stratified sampling for SV approximation.* Computing the SV of a participant requires to evaluate many coalitions. Instead of enumerating these coalitions, we sample some coalitions to meet a given target for estimation accuracy. Following the Neyman approach [42], we begin with a sample allocation method based on the variances of the coalitions but find that the variances cannot be obtained before evaluating the coalitions, which causes a chicken-egg problem. Using the Popoviciu's inequality [46], we adopt the range of coalition utility to replace variance. Moreover, we also fully reuse the evaluated coalition utilities across participants by reformulating the SV.
- *Implementation optimizations.* A naive implementation of our proposed PS-MI needs to compute the distances of all data samples to a set of query samples for KNN. This is expensive because there can be many data and query samples. To reduce running time, we propose a suite of efficiency optimizations, i.e., *locality sensitive hashing (LSH) for KNN candidate pruning*, *batched candidate checking for each query sample*, and *adaptive termination for coalition utility evaluation.* In particular, for each query sample, LSH can narrow down its KNNs from all data samples to a small candidate set while protecting data privacy; by batching and sharing the hashing for different query samples and coalitions, redundant computations are avoided; adaptive termination saves computation by stopping evaluating a coalition once the accuracy target is achieved.

We conduct extensive experiments to compare our PS-MI with five state-of-the-art VFDV methods on multiple real-world datasets. We show that PS-MI produces accurate SV estimations, reaching a Pearson's correlation coefficient up to 0.99 with the ground-truth SVs. Meanwhile, PS-MI runs significantly faster than the baselines, achieving a maximum speedup of 592×. PS-MI can also benefit downstream model training since using its SV estimations to conduct participant selection usually yields higher model accuracy than using the estimations of the baselines. Moreover, ablation studies verify the effectiveness of our designs and optimizations.

2 PRELIMINARIES

In this section, we introduce differential privacy in Section 2.1 and present the setup of vertical federated data valuation in Section 2.2. Table 2 summarizes the frequently used notations.

2.1 Differential Privacy

Differential privacy (DP) is a rigorous privacy definition of data disclosure that prevents attempts from learning private information about any individual in a data release. A standard notion of differential privacy is $(\epsilon, \delta) -$ DP, which is defined as follows [10].

Definition 1 ($(\epsilon, \delta) -$ differential privacy)**.** A random algorithm $\mathcal{A}$ is $(\epsilon, \delta) -$ differentially private if for any pair of datasets D and D' that differ in one record and for all possible subset O of possible outputs of $\mathcal{A}$, we have $Pr[\mathcal{A}(D) \subseteq O] \leq \exp(\epsilon) Pr[\mathcal{A}(D') \subseteq O] + \delta$ where $Pr[\cdot]$ denotes the probability of an event.

DP requires the output of a randomized algorithm to be approximately the same if any single record is replaced with a new one. The parameter ϵ is called the privacy budget where the smaller ϵ means stronger privacy protection is provided. The Gaussian mechanism, one of the most classic DP mechanisms, adds Gaussian noise to the return of a function f to ensure the result is differentially private. The variance of the noise depends on L_2 sensitivity of f, defined with a pair of neighboring datasets as $\Delta_2 f = \max_{D,D'} \|f(D) - f(D')\|_2$. The Gaussian mechanism is formulated as $\mathcal{M} = f + \mathcal{N}(0, \sigma^2)$, where $\sigma^2 = \frac{2\ln(1.25/\delta)\cdot(\Delta_2 f)^2}{\epsilon^2}$. When f outputs a vector, $\mathcal{M}$ adds independent noises, sampled from $\mathcal{N}(0, \sigma^2)$, to each element of the vector.

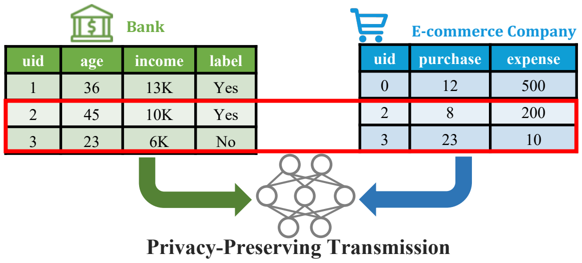
2.2 Vertical Federated Shapley Value

Data Layout. Let $\mathcal{P}$ be a participant set with P participants and a dataset $\mathcal{D} = \{X, Y\}$ with N data samples. $X \in \mathbb{R}^{N\times F}$ is the joint feature space where F is the dimension of the joint feature space. In the VFL setting, this joint feature space X is vertically partitioned over different participants — each participant $p \in \mathcal{P}$ holds a subset of features (columns) of X, denoted by X_p, such that: $X = [X_1, \cdots, X_p, \cdots, X_P]$. Here $[., \cdots, .]$ denotes the concatenation operation. The local dataset for each participant $p \in \mathcal{P}$ is $\mathcal{D}_p = X_p \in \mathbb{R}^{N\times F_p} = \{x_i^p : i \in [N]\}$. Only one participant called the leader participant holds the label set $Y = \{y_i : i \in [N]\}$. Figure 1 shows an example of VFL, where a bank wants to build a fraud detection model with an e-commerce company.

Data Alignment. In VFL, the data samples among participants are assumed to be *vertically aligned* [3, 15, 16]. In other words, the overlapping samples from participants (e.g., instances 2 and 3 in Figure 1) have been extracted and organized in the same order prior to training. This can be achieved by the private set intersection (PSI) technique [5, 21, 45, 47]. Then, participants can use the same random seed to sample synchronized mini-batches..

Table 2: The summary of frequently used notations.

Symbol	Description
N	the number of samples
$\mathcal{P}$	the set with P participants
X_p	feature matrix of participant p
$\mathcal{S}$	the coalition of participants
$v(\cdot)$	utility function
R	Gaussian random projection matrix
ψ	Gaussian noise matrix
Q	the query set
$s(p)$	Shapley value of participant p



uid	age	income	label
1	36	13K	Yes
2	45	10K	Yes
3	23	6K	No

uid	purchase	expense
0	12	500
2	8	200
3	23	10

Figure 1: An illustration of vertical federated learning (VFL).

Problem Formulation The Shapley value is a broadly adopted concept in collaborative game theory for evaluating a participant's contribution to a coalition. Consider a set $\mathcal{P}$ consisting of P participants. A utility function $v : 2^P \to \mathbb{R}$ maps each possible coalition $\mathcal{S} \subseteq \mathcal{P}$ to a real number that describes the utility of a coalition. Shapley value measures the expectation of marginal contribution by participant $p \in \mathcal{P}$ in all possible coalitions. That is, $s(p) = \frac{1}{P}\sum_{\mathcal{S}\subseteq\mathcal{P}\setminus p}\binom{P-1}{|\mathcal{S}|}^{-1}[v(\mathcal{S}\cup\{p\}) - v(\mathcal{S})]$. The formula can be rewritten in expectation: $s(p) = \frac{1}{P!}\sum_{\pi\in\Pi(\mathcal{P})}[v(\mathcal{S}_\pi \cup \{p\}) - v(\mathcal{S}_\pi)]$ where $\Pi(\mathcal{P})$ is the set of all permutations of participants, and $\mathcal{S}_\pi$ is the set of participants that precede p. A coalition of P participants can form in $P!$ orders. The Shapley value of each participant is the average of the marginal contributions over all the possible orders.

The SV is arguably the most widely studied scheme for data valuation [23, 33, 62]. It is the only existing measure that satisfies all the four fundamental requirements of fair reward allocation, including balance, symmetry, additivity, and zero element [48]. For formal statements of these axioms, see our technical report [11]. These desirable properties motivate us to adopt the SV for VFDV.

In this work, the goal of VFDV is thus to estimate the Shapley value $\hat{s}_p$ of each $p \in \mathcal{P}$ in VFL. We formalize this goal as follows.

Definition 2. Given a set of participants $\mathcal{P}$, VFDV aims to estimate the Shapley value $\hat{s}_p$ for each $p \in \mathcal{P}$ based on a utility function $v : 2^{\mathcal{P}} \to \mathbb{R}$. The estimation must satisfy (ϵ, δ)-differential privacy with respect to each participant's local data.

3 THE PS-MI METHOD

To achieve accurate, secure, efficient vertical federated data valuation, we aim to answer the following questions.

- *How to formulate a model-agnostic utility function?*
- *How to securely compute the utility for each coalition?*
- *How to reduce the required number of evaluated coalitions?*

To achieve model-agnostic data valuation, we first show that the utility function can be expressed as the mutual information (MI) between the participants' local features and the labels in Section 3.1. Exact Shapley value computation requires estimating the mutual information for $2^P - 1$ coalitions, and thus the total computational complexity is $C_{eval} * O(2^P)$, where C_{eval} is the per-coalition evaluation cost. To reduce per-coalition evaluation cost C_{eval}, we propose an efficient but privacy-preserving mutual information estimator in Section 3.2. To reduce the number of evaluated coalitions, we design a stratified sampling method to efficiently approximate the Shapley value of the participants in Section 3.3. The core task then becomes identifying the k-nearest neighbors for the query dataset corresponding to each coalition, as shown in Figure 2. To enable efficient implementation, we design a series of optimization strategies: LSH-based pruning to narrow candidate sets in Section 4.1; batched candidate checking to avoid redundant computation across coalitions in Section 4.2; adaptive termination to stop early once accuracy is sufficient in Section 4.3.

3.1 Utility Formalization: Mutual Information

To achieve effective data valuation, it is crucial to determine how to measure the utility of a data coalition. As mentioned above, the utility of a coalition should be model-agnostic. Our basic idea is to quantify the reduction in risk over the mean prediction of a given set of participants. Below we first provide a general definition of the utility function v in vertical federated data valuation.

Definition 3. Let $\ell(\cdot, \cdot)$ represent the loss function, and let $G(S)$ denote the set of predictors used to predict the label Y, trained over the participant subset $\mathcal{S} \subseteq \mathcal{P}$. Given the utility function $v : 2^P \to \mathbb{R}$, the utility of $\mathcal{S}$ is defined to be:

$$v(S) = \min_{g \in G(\emptyset)} \mathbb{E}\Big[\ell\big(g(X_{\emptyset}), Y\big)\Big] - \min_{g \in G(S)} \mathbb{E}\Big[\ell\big(g(X_{\mathcal{S}}), Y\big)\Big]$$

The left term is the loss achieved with the mean prediction when the model g is trained on the empty set and the right term is the loss achieved using the features $X_{\mathcal{S}}$ over the participant coalition $\mathcal{S}$. The utility function quantifies the predictive power that g learns from the features $X_{\mathcal{S}}$. Such a definition is widely used in importance measurements and explainability [4, 7, 20]. We observe that, for a classification model trained with cross-entropy loss, the utility function v is equivalent to the Shannon mutual information (MI) between the features $X_{\mathcal{S}}$ over $\mathcal{S}$ and the labels Y. The proof of Lemma 1 is provided in our technical report [11].

Lemma 1. For a classification problem, let C be the number of classes. If the loss function is $\ell = -\sum_{c=1}^{C} y_c \log(p_c)$, it follows that $v(S) = I(S; Y) = H(Y) - H(Y|X_{\mathcal{S}})$ where $I(\cdot, \cdot)$ is the Shannon mutual information and $H(\cdot)$ is the entropy.

Lemma 1 reveals that the mutual information between local features and labels can express the predictive power of the local features X_p held by participant p. Thus the utility of participant p can be measured by MI between X_p and the labels Y.

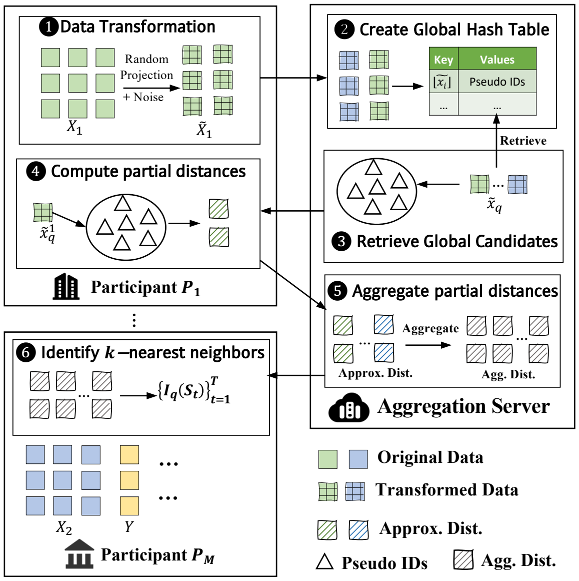


Figure 2: The system architecture and workflow of PS-MI.

Intuition. In this work, we utilize the mutual information as the utility metric for three main reasons. *First*, from an information-theoretic perspective, mutual information quantifies the reduction in uncertainty about the target variable in predicting Y attributed to $X_{\mathcal{S}}$. *Second*, as demonstrated in Lemma 1, mutual information is the reduction of minimum expected loss in predicting Y when the model is trained over the coalition S using cross-entropy loss. *Third*, mutual information is a model-agnostic utility metric, relying only on data distribution and applicable across diverse models. We provide a detailed discussion in our technical report [11].

The Shapley value of participant p can be thus expressed as

$$s(p) = \frac{1}{P} \sum_{\mathcal{S} \subseteq \mathcal{P} \backslash p} \binom{P-1}{|\mathcal{S}|}^{-1} [I(\mathcal{S} \cup \{p\}; Y) - I(\mathcal{S}; Y)]. \quad (2)$$

Comparison to Previous Results. Previous works [22, 24, 31] have employed MI to quantify participant contributions in VFL. However, these methods overlook the marginal contributions of participants and fail to ensure fairness among them. Han et al. [18] proposed using mutual information as a utility measure, but this approach requires every participant to access labels, which poses significant privacy risks and is impractical in VFL. Critically, none of these works offer theoretical justification for using MI as a measure of utility; they rely solely on empirical evidence. Instead, our work theoretically formalizes the VFDV problem and establishes that MI is the reduction of the minimum expected loss in predicting labels for the classification task when the loss function is cross-entropy.

KNN-based MI Estimator. Now, the next question becomes *how to estimate the mutual information for each coalition* $\mathcal{S} \subseteq \mathcal{P}$. Among the various methods to estimate MI, k-nearest neighbors (KNN) MI estimators are widely used due to their superior theoretical and practical performance [30, 44]. The basic idea is to estimate the local log-density around each data sample by computing the volume of the ball that encloses its k-nearest neighbors. Formally, consider the dataset $\mathcal{D} = \{(x_i, y_i)\}, i = 1, 2, \ldots, N$ where x_i denotes the feature vector of sample i and y_i is the label. For each sample $i \in \mathcal{D}$, the process contains three major steps:

❶ identify the k-nearest neighbors of the sample i among N_i samples that share the same label with y_i;

❷ compute the maximal distance $\overline{d_i}$ between i and its k neighbors;

❸ count the number of neighbors m_i in the full dataset $\mathcal{D}$ that lie within distance $\overline{d_i}$ to sample i.

Based on N_i and m_i, we compute $I_i = \psi(N) - \psi(N_i) + \psi(k) - \psi(m_i)$ where $\psi(x) = \frac{d}{dx} ln(\Gamma(x)) \sim lnx - \frac{1}{2x}$ is the digamma function. Given the query set $Q \subseteq \mathcal{D}$, we estimate the mutual information by averaging I_i over the query set Q $I(X; Y) = \frac{1}{|Q|} \sum_{i \in Q} I_i$.

3.2 Utility Evaluation: Random Projection

KNN-based MI estimator needs to identify the k-nearest neighbors for $|Q|$ query samples. This process requires participants to calculate partial distances $d^p = [\|x_i^p - x_q^p\|_2^2, i \in [N]]$ between their local features and each query sample $q \in Q$, then send these distances to an aggregation server. However, these partial distances could potentially leak the feature privacy of participants. To address this issue, existing works [22, 24] utilize homomorphic encryption (HE) to encrypt the partial distances. HE provides a strong privacy guarantee, however, it is time-consuming for encryption and mathematical operation on encrypted data. To address this problem, we resort to differential privacy, a more efficient data protection technique. Differential privacy protects data by adding noise, which is easy to implement and resource-efficient. However, this protection inherently introduces a privacy-utility trade-off, as the added perturbations degrade estimation accuracy.

To address this issue, we propose a novel differentially private mutual information estimator (DP-MI) with a utility guarantee. The main idea of DP-MI is to project the local features of participants into a different feature space. This preserves the distance characteristics of the original feature space essential for mutual information (MI) estimation and enables secure data sharing [60]. Participants and the server then jointly estimate MI for coalitions based on these transformed features. The DP-MI procedure consists of two main steps: data transformation and mutual information estimation, as outlined in Algorithm 1. The data transformation in DP-MI consists of two steps: *random projection* and *noise perturbation*.

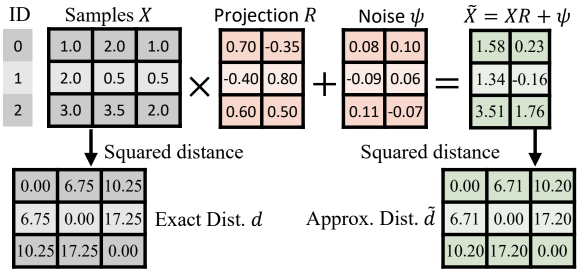
❶ *Random Projection*: each participant p generates a Gaussian random projection matrix $R_p \in \mathbb{R}^{F_p \times r_p}$ where each entry is chosen independently drawn from $\mathcal{N}(0, 1)$, and projects its local features $X_p \in \mathbb{R}^{N \times F_p}$ into new feature space as $X_p R_p \in \mathbb{R}^{N \times r_p}$.

❷ *Noise Perturbation*: each participant p generates Gaussian noise matrix $\psi_p \in \mathbb{R}^{N \times r_p}$, where each entry is independently sampled from $\mathcal{N}(0, \sigma^2)$. The noise matrix is then added to the transformed features it to the projected features $\widetilde{X_p} = X_p R_p + \psi_p$.

After data transformation, each participant p computes $\widetilde{d^p} = [\|\widetilde{x}_i^p - \widetilde{x}_q^p\|_2^2, i \in [N]] - 2r\sigma^2$ as partial distances and sends $\widetilde{d_p}$ instead of d_p during mutual information estimation.

Algorithm 1: DP-MI

```
Input: Coalition $\mathcal{S}$; Privacy parameters $\epsilon, \delta$; Projection dimension $r_p$
Output: Mutual information $I(X_\mathcal{S}, Y)$
Operation: Data Transformation
Random Projection. Each participant $p \in \mathcal{S}$ generates a Gaussian projection
  matrix $R \in \mathbb{R}^{F_p \times r_p}$ and projects its local features $X_p$ into $X_p R_p$ ;
Noise Perturbation. Each participant $p \in \mathcal{S}$ generates a Gaussian random
  noise matrix $\psi \in \mathbb{R}^{N \times r_p}$. The noise matrix is then added to the
  transformed features it to the projected features $\widetilde{X_p} = X_p R_p + \psi_p$;
Operation: Mutual Information Estimation
for $q = 1$ to $|Q|$ do
    Participant $p \in \mathcal{S}$:
    $\widetilde{d^p} = [\|\widetilde{x}_q^p - \widetilde{x}_i^p\|_2^2 - 2r_p\sigma^2, i \in \mathcal{D}]$;
    // Compute the partial distances $\widetilde{d^p}$
    (Leader) Compute $I_q = \psi(N) - \psi(N_q) + \psi(k) - \psi(m_q)$;
    Server:
    $\widetilde{d} = \sum_{p \in \mathcal{S}} \widetilde{d^p}$;
    // Aggregate partial distances $\widetilde{d^p}$ into the complete
       distances $\widetilde{d}$ and send $\widetilde{d}$ to leader
(Leader) Compute $I(X_\mathcal{S}; Y) = \frac{1}{|Q|} \sum_{i \in Q} I_i$ for coalition $\mathcal{S}$ ;
```



$$\begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{bmatrix} 1.0 & 2.0 & 1.0 \\ 2.0 & 0.5 & 0.5 \\ 3.0 & 3.5 & 2.0 \end{bmatrix} \times \begin{bmatrix} 0.70 & -0.35 \\ -0.40 & 0.80 \\ 0.60 & 0.50 \end{bmatrix} + \begin{bmatrix} 0.08 & 0.10 \\ -0.09 & 0.06 \\ 0.11 & -0.07 \end{bmatrix} = \begin{bmatrix} 1.58 & 0.23 \\ 1.34 & -0.16 \\ 3.51 & 1.76 \end{bmatrix}$$

$$d = \begin{bmatrix} 0.00 & 6.75 & 10.25 \\ 6.75 & 0.00 & 17.25 \\ 10.25 & 17.25 & 0.00 \end{bmatrix} \qquad \tilde{d} = \begin{bmatrix} 0.00 & 6.71 & 10.20 \\ 6.71 & 0.00 & 17.20 \\ 10.20 & 17.20 & 0.00 \end{bmatrix}$$

Figure 3: An example of data transformation in DP-MI.

Example 1. Figure 3 shows an example of data transformation. After projection and noise addition, the squared distance between sample 0 and 1 changes slightly from 6.75 to 6.71. Notably, the nearest neighbor of all samples remains the same.

In the following, we explain how DP-MI preserves the utility while guaranteeing data privacy when estimating MI for coalitions.

Utility Guarantee. The utility guarantees depend on the random projection matrix R_p and noises ψ. The squared Euclidean distance in transformed feature space is unbiased.

Theorem 1. Let $R \in \mathbb{R}^{F \times r}$ be a Gaussian random projection matrix and $\psi \in \mathbb{R}^{N \times r}$ be a Gaussian noise matrix. For any $x_i, x_j \in X \in \mathbb{R}^{N \times F}$ and $\widetilde{x}_i, \widetilde{x}_j$ denote their corresponding elements in $\widetilde{X} = XR + \psi$. Then, $\|\widetilde{x}_i - \widetilde{x}_j\|_2^2 - 2r\sigma^2$ is unbiased estimator of $\|x_i - x_j\|_2^2$.

Proof. Due to the limited space, please see our technical report [11] for detailed proof. The same to the following theorems. □

Theorem 1 states that the squared Euclidean distance between two vectors in expectation can be preserved after the data transformation of differential privacy in our DP-MI.

Privacy Guarantee. DP-MI enhances privacy by adding Gaussian noise to the projected feature vectors XR. Now we investigate the minimum amount of noise ψ that must be added to XR to ensure that DP-MI satisfies (ϵ, δ) – differential privacy.

Corollary 1. Given the vectors X and a random projection matrix R, $XR + \psi$ satisfies (ϵ, δ) − differential privacy if $\delta < \frac{1}{2}$ the noises ψ are sampled from $\mathcal{N}(0, \sigma^2)$ with $\sigma \geq \frac{2}{\epsilon}\sqrt{\ln(\frac{1}{2\delta}) + \epsilon}$.

Corollary 1 establishes the minimum noises required to achieve (ϵ, δ) − differential privacy. This result is derived as a by-product of the proof for the well-known distributed Gaussian mechanism [9].

3.3 Approximate Valuation: Stratified Sampling

Above we provide an efficient vertical federated mutual information estimator DP-MI. Taking DP-MI as a basic operation, computing the exact Shapley value using Equation 2 still requires executing $2^P - 1$ times DP-MI for all possible coalitions. It will bring expensive computation and communication costs in VFL. The Monte Carlo (MC) sampling is commonly used to compute the approximate SV [18, 23]. The core idea behind MC-based methods is to use the *sample mean* to approximate the SV. More formally, let $\mathcal{T} = [\pi_1, \pi_2, \ldots, \pi_T]$ be T permutations and each permutation randomly sampled from $\Pi(\mathcal{P})$ with a probability of $1/P!$. The approximate SV of participant p is $\hat{s}(p) = \frac{1}{T}\sum_{\pi \in \mathcal{T}}[v(\mathcal{S}_\pi \cup \{p\}) - v(\mathcal{S}_\pi)]$. $\hat{s}_p$ is the average of utility difference ϕ_p over T sampled permutations. The estimation error $|\hat{s}(p) - s(p)|$ can be bounded by applying Hoeffding's inequality [40]. The detailed process of the Monte Carlo method is provided in our technical report [11].

MC-based methods are based on simple random sampling, which treats all coalitions equally, ignoring population bias. This can lead to high estimation variance, especially with uneven subgroup sizes [23, 63]. Stratified sampling can give a smaller variance than simple random sampling [29, 63]. Specifically, stratified sampling partitions coalitions into P disjoint stratum $\mathcal{G} = \{\mathcal{G}^1, \ldots, \mathcal{G}^P\}$ based on coalition size, each of which contains N_j coalitions. Neyman allocation [42] is the optimal allocation that allocates samples to strata and minimizes the sample variance of the estimator.

$$\min Var[\hat{s}_p] = \sum_{j=1}^{P} \frac{N_j \sigma_j^2}{N^2} \quad \text{s.t.} \sum_{j=1}^{P} N_j = N \tag{3}$$

where σ_j^2 is the variance of the coalitions utilities in stratum j.

Stratum Size. Neyman approach allocates more samples to larger or more variable strata, with optimal allocation depending on stratum variance. Existing works typically sample coalition utilities to estimate variance [29, 63], which is impractical for VFL due to high computation costs. Fortunately, we can easily know the utility range of each stratum. Based on the Popoviciu's inequality [46], i.e. the variance $\sigma^2 \leq \frac{r^2}{4}$, we approximate variance using the utility range r_j of each stratum j which yields a practical sample allocation. Given a total sample size T, the optimal size T_j^* of stratum $\mathcal{G}^j$ is $T_j^* = T\frac{N_j r_j^2}{\sum_{j=1}^{P} N_j r_j^2}$. The variance σ^2 only determines proportional allocation. Thus the range can serve the purpose well [39].

Sample Strategy. After determining the size of each stratum, how should we sample effectively? A naive approach is to apply the Monte Carlo method for each stratum. However, in this approach, one sample of marginal contributions $I(\mathcal{S} \cup \{p\}) - I(\mathcal{S})$ can only be used to update the SV for one participant p, although coalition $\mathcal{S}$ may contain many other participants. We observe that the coalitions utilities can be reused for efficient computation, as illustrated below.

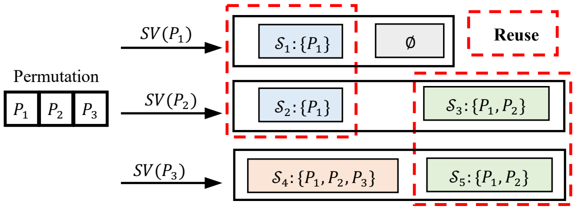


Figure 4: A running example of coalition utility reuse.

Example 2. Consider a sampled permutation $[P_1, P_2, P_3]$, the utilities of $v(\{P_1, P_2\})$ and $v(\{P_1\})$ are computed for estimating $SV(P_2)$ for P_2. $v(\{P_1, P_2\})$ and $v(\{P_1\})$ can be reused for estimating $SV(P_3)$ for P_3 (as shown by the red dotted box in Figure 4).

In order to estimate the Shapley value of multiple participants simultaneously, we can treat the Shapley value as the difference of two utility expectations and reuse utilities accordingly. That is,

$$s(p) = \mathbb{E}_{\pi \in \Pi(\mathcal{P})}[v(\mathcal{S}_\pi \cup \{p\})] - \mathbb{E}_{\pi \in \Pi(\mathcal{P})}[v(\mathcal{S}_\pi)] \tag{4}$$

Note that permutation π in $v(\mathcal{S}_\pi \cup \{p\})$ is not necessary the same as π in $v(\mathcal{S}_\pi)$. The SV can be rewritten over the stratum

$$\hat{s}(p) = \frac{1}{P}\sum_{j=1}^{P}\left[\mathbb{E}_{\mathcal{S}_\pi \in \mathcal{G}^j} v(\mathcal{S}_\pi \cup \{p\}) - \mathbb{E}_{\mathcal{S}_\pi \in \mathcal{G}^j} v(\mathcal{S}_\pi)\right] \tag{5}$$

where $\mathcal{G}^j = \{\mathcal{S} | \mathcal{S} \subseteq \mathcal{P}, |\mathcal{S}| = j\}$. In this way, the coalition utilities can be reused since all evaluations of $v(\mathcal{S})$ are used for estimating the Shapley value $s(p)$ for every $p \in \mathcal{P}$.

Theorem 2. The estimated Shapley value $\hat{s}(p)$ is an unbiased estimator of $s(p)$, i.e. $\mathbb{E}(\hat{s}(p)) = s(p)$.

Based on Hoeffding's inequality, we can obtain a bound on the error of approximate Shapley value $\hat{s}(p)$ as follows.

Lemma 2. Given the range r_j of each stratum $\mathcal{G}^j$, an error bound $\epsilon > 0$, then $Pr(\hat{s}(p) - s(p) \leq \epsilon) = 1 - 2\exp(\frac{2\epsilon^2}{\sum_{j=2}^{P-1}\frac{r_j^2}{P^2 T_j}})$.

3.4 Overall Workflow

Below we present a naive implementation of our PS-MI, as outlined in Algorithm 2. The server maintains a stratified sampler that generates a set of T coalitions to be evaluated (line 1). Each participant p applies a one-time data transformation based on the Gaussian random projection matrix R and noise matrix ψ to obtain $\widetilde{X_p}$ (line 2-3). For each $t \in [T]$, participants and the server collaboratively execute DP-MI based on the already-transformed features $\widetilde{X_p}$ in Algorithm 1 to estimate the mutual information between the features of coalition $\mathcal{S}_t$ and the labels (line 4-5). After computing mutual information for all coalitions, the leader computes the Shapley value $\hat{s}(p)$ using Equation 5 for each participant $p \in \mathcal{P}$ (line 6).

Complexity Analysis. Let α be the cost to calculate a partial distance and β be the cost to sum two distances. For each coalition, the computation and communication complexity is $O(N_Q N)$. Thus

Algorithm 2: Naive Implementation of **PS-MI**

```
Input: Query sets of each participant $Q_1, \ldots, Q_P$
Output: SVs of participants $[s_1, s_2, \ldots, s_P]$
(Server) Generate $T$ coalitions using strategy in Section 3.3;
Operation: Data Transformation
Random Projection: Each participant $p \in \mathcal{S}$ generates a Gaussian projection
  matrix $R \in \mathbb{R}^{F_p \times r_p}$ and projects $X_p$ into $X_p R_p$ ;
Noise Perturbation: Each participant $p \in \mathcal{S}$ generates a Gaussian random
  noise matrix $\psi \in \mathbb{R}^{N \times r_p}$. The noise matrix is then added to the
  transformed features it to the projected features $\widetilde{X}_p = X_p R_p + \psi_p$;
for $t = 1$ to $T$ do
    Invoke the mutual information estimation procedure for the $t$-th
      coalition as defined in Algorithm 1;
(Leader) Compute $\hat{s}(p)$ using Equation 5.
```

the overall computation complexity is $O(TN_QNP)$ and the total communication complexity is $O(TN_QN)$.

4 IMPLEMENTATION OPTIMIZATIONS

The inefficiency of the naive workflow arises from the following: ❶ computing partial distances for all samples for a query, ❷ repeatedly enumerating the same query samples in Q across different coalitions, ❸ evaluating each coalition over the entire query set, even when it has already been fully evaluated. Obviously, there exist considerable redundant computations. To overcome these challenges, we propose a series of efficiency optimization strategies:

❶ To reduce the number of samples needed for distance computation, we leverage locality sensitive hashing (LSH) to obtain a much smaller candidate set for a query in Section 4.1.

❷ To avoid model retraining for T coalitions, we introduce a batch optimization strategy to calculate the utility of different coalitions for one query in one communication round in Section 4.2.

❸ To reduce the number of query samples N_Q, we design an adaptive termination mechanism, which can dynamically discard the fully evaluated coalitions during evaluation in Section 4.3.

4.1 Candidate Pruning with LSH

DP-MI requires identifying the nearest samples for a query to estimate the mutual information. We leverage locality sensitive hashing (LSH) to prune the candidates needed for distance computation. In LSH, every data sample is converted into codes in each hash table by using a hash function h. The hash function is designed to preserve the relative distance between different data samples. In other words, similar data samples have the same hashed value with high probability. We choose the Euclidean space under the l_2 norm, in this case, a commonly used hash function is: $h(x) = \lfloor \frac{a \cdot x + b}{r} \rfloor$. where a is a F-dimensional vector each of whose entries is chosen from the standard Gaussian distribution $\mathcal{N}(0, 1)$ and $b \in \mathbb{R}$ is uniformly chosen from the range $[0, r]$. The overall idea is that if two samples are "close" together in the Euclidean space and if we project them onto some other vector drawn from Gaussian distribution, then they should remain "close" to each other. We observe that LSH can naturally combine with our DP-MI, requiring only a little additional computation. In DP-MI, each participant $p \in \mathcal{P}$ generates the Gaussian projection matrix $R \in \mathbb{R}^{F_p \times r}$ to transform original features into $\widetilde{X}_p = X_p R_p$. This transformation aligns with the hash function used in Euclidean LSH. When aggregated, the global projection becomes $\widetilde{X} = XR$ where $R^{F \times r}$ and each entry of R is sampled from $\mathcal{N}(0, 1)$. This effectively applies r independent hash functions via one matrix multiplication. As a result, hash values can be directly obtained from $\widetilde{X}$ with negligible overhead.

Implementation Details. The overall process is as follows. ❶ *Local hashing*. Each participant computes local hash values h_p for all samples and sends them to the server. ❷ *Hash aggregation*. The server computes the global hash code as $h(x_i) = \sum_{p \in \mathcal{P}} h_p(x_i)$. ❸ *Hash table construction*. The server builds a hash table H, where hash codes are keys and pseudo identities (IDs) are values. ❹ *Query processing*. For a query q, the server retrieves a candidate set C from bucket $h(q)$, significantly narrowing the search space. To protect the identities, we shuffle the data and use pseudo IDs during transmission. Specifically, each participant shuffles data using a shared seed and assigns pseudo IDs based on shuffled indices. The mapping to original IDs is stored locally and used only when needed.

4.2 Batched Candidate Checking

To solve the second inefficiency problem of the naive workflow, we study *can we batch repeated computation and communication to avoid redundancy?* By reforming the definition of Shapley value, we can accurately calculate the SV without any repeated computation.

The naive workflow directly interprets the definition of SV and chooses a coalition-wise mechanism that handles all possible coalitions iteratively. It is the fundamental reason for the inefficiency problem because different coalitions handle the same query samples individually. To address this, we adopt a sample-wise approach by reformulating the Shapley value as a sum over query samples.

$$\hat{s}(p) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|Q|} \sum_{q \in Q} I_q \left(\mathcal{S}_t \cup \{p\} \right) - I_q(\mathcal{S}_t)) \tag{6}$$

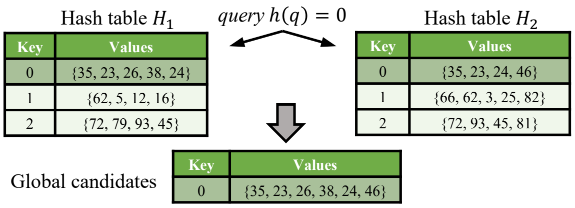
$$= \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{T} \sum_{t=1}^{T} I_q \left(\mathcal{S}_t \cup \{p\} \right) - I_q(\mathcal{S}_t)), \tag{7}$$

where $I_q = \psi(N) - \psi(N_q) + \psi(k) - \psi(m_q)$. Here the utility is first calculated over a single query sample instead of all query samples. To avoid redundancy, we need to answer the following question:

> *Can we obtain the utility of different coalitions for one instance in one round of communication?*

As we will show, after the union of candidates sets over different coalitions, we can run KNN in VFL only once for each query sample and calculate all necessary utility values.

Batched Hashing and Querying. To reduce the samples involved in identifying the k-nearest neighbors, we adopt locality-sensitive hashing (LSH). It narrows down the candidate set by searching samples in the same bucket as the query. However, candidate sets vary across coalitions. To this end, we propose to retrieve all possible candidates at once and batch KNN tasks of all coalitions in a single execution. Specifically, each participant $p \in \mathcal{P}$ computes the local hash codes $h_p(X_p)$ for its local features X_p and sends them to the server. The server builds T hash tables $\mathcal{H} = (H_1, \ldots, H_T)$, each for coalition $\mathcal{S}_t$, $t \in [T]$. For a query q, we retrieve candidates C_t from the corresponding bucket in each H_t. The union of all C_t forms the global candidate set C. Each participant computes local distances



Key	Values
0	{35, 23, 26, 38, 24}
1	{62, 5, 12, 16}
2	{72, 79, 93, 45}

Key	Values
0	{35, 23, 24, 46}
1	{66, 62, 3, 25, 82}
2	{72, 93, 45, 81}

Key	Values
0	{35, 23, 26, 38, 24, 46}

Figure 5: An example of batch optimization mechanism.

between q and all samples in C and sends them to the server. The server performs the additive aggregation for each $\mathcal{S}_t$. The global candidate set allows one-pass utility evaluation for all coalitions with the shared distance computations, which effectively reduces the computation and communication overhead.

Example 3. As shown in Figure 5, consider a query q with hash code $h(q) = 0$. For two coalitions $\mathcal{S}_1$ and $\mathcal{S}_2$, the candidate sets are $\{35, 23, 26, 38, 24\}$ and $\{35, 23, 24, 46\}$, respectively. Their union is $\{35, 23, 26, 38, 24, 46\}$, includes the k-nearest neighbors for both coalitions. Thus, local distances need to be computed only once, enabling shared utility evaluation for both $\mathcal{S}_1$ and $\mathcal{S}_2$.

4.3 Adaptive Coalition Termination

For each coalition, the naive workflow needs to calculate the mutual information over the entire query set $\mathcal{Q}$. However, some coalitions may already be sufficiently evaluated, yet their valuation continues. That means that the utility of these coalitions will change only marginally. This leads to significant redundant computation and communication costs. To address this issue, we propose an adaptive valuation mechanism that dynamically excludes fully evaluated coalitions, enabling a more efficient Shapley value estimation.

Theorem 3. Given the query set $\mathcal{Q}$, an error bound ϵ, if $\sigma^2 \leq \frac{|\mathcal{Q}|\epsilon^2}{-\ln(\delta)} - \frac{1}{3}r\epsilon$, then $\hat{I} - I \leq \epsilon$ holds with probability at least $1 - \delta$.

The proof of Theorem 3 is provided in the techinical report [11]. Specifically, we compute the unbiased estimation of $\sigma^2_{\mathcal{S}}$ for each coalition $\mathcal{S}$ using Bessel's correction once a predetermined threshold is reached. If $\sigma^2_{\mathcal{S}}$ meets the condition outlined in Theorem 3, the coalition $\mathcal{S}$ is excluded from the further evaluations.

Example 4. Figure 6 shows an example of the adaptive mechanism applied to a query set of size 6 and four coalitions. Coalition $\mathcal{S}_2$ is fully evaluated by query q_3 and is therefore excluded from further evaluations (i.e., q_4, q_5). The same applies to coalitions $\mathcal{S}_3$ and $\mathcal{S}_4$.

4.4 Optimized Execution Workflow

Below we describe the process of our proposed PS-MI with optimization strategies in detail. We provide the pseudo-code in our technical report [11]. The server maintains a stratified sampler that selects a set of T coalitions to be evaluated. The core task is to identify the k-nearest neighbors for the query dataset corresponding to each coalition $\mathcal{S}_t, t \in [T]$. As shown in Figure 2, the procedure of *k-nearest neighbor search*, which includes the following steps:

❶ *Data transformation.* Each participant $p \in \mathcal{P}$ applies Gaussian random projection and adds noise to the local data to obtain $\widetilde{X_p}$.

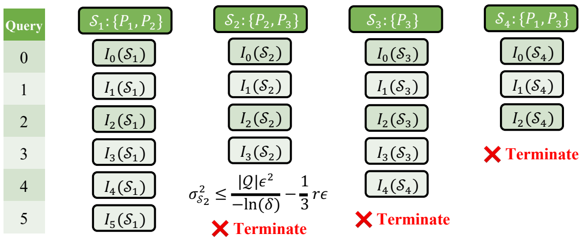


Figure 6: A running example of the adaptive mechanism.

❷ *Create the hash table.* The server builds a global hash table, where the keys are $\lfloor \widetilde{x_i} \rfloor$ for each $\widetilde{x_i} \in \widetilde{X}$, and the values are pseudo IDs.

❸ *Retrieve the global candidates.* For each query $q \in \mathcal{Q}$, each participant p retrieves the LSH candidates C from the server and computes the distances $\widetilde{d^p}$ between q and each candidate.

❹ *Compute partial distances.* Each participant p sorts $\widetilde{d^p}$ in ascending order to obtain a sorted list $L_p = [(i, \widetilde{d^p}(q, i), i \in C]$. The server and participants then apply the Fagin algorithm [12] to find the top-k candidates C_F from all sub-rankings [22, 24].

❺ *Aggregate partial distances.* Each participant sends their partial distances to the server, which aggregates them for each coalition: $\widetilde{d} = \{\widetilde{d_t} = \sum_{p \in \mathcal{S}_t} \widetilde{d^p} \mid t \in [T]\}$.

❻ *Identify k-nearest neighbors.* For each coalition $\mathcal{S}_t$, the leader computes $I_q(\mathcal{S}_t)$ as defined in Section 3.1.

After processing all query samples or all coalitions are fully evaluated, the workflow will be terminated. Then the leader computes the estimated Shapley value $\hat{s}(p)$ for each $p \in \mathcal{P}$.

Complexity Analysis Let N_C be the average number of the candidates and $\overline{N_Q}$ be the average query samples during the evaluation. For each participant, the complexity of retrieving the candidates from the global hash table is $O(\overline{N_Q}N_C)$. Thus the overall computation complexity is $O(\overline{N_Q}N_C P + TN_C)$, significantly lower than $O(TN_Q NP)$ of the naive implementation of PS-MI.

As we will show in the experiments in Section 5.4, $\overline{N_Q}N_C$ is much smaller than the involved samples of naive workflow $TN_Q N$.

4.5 Security Analysis

Below we analyze the security guarantee of our PS-MI. We consider the semi-honest model [16, 24, 58], a commonly used threat model used in FL [17, 32, 59]. That is, every party follows the protocol but it tries to infer other parties' private data based on the messages received. Some existing works [22, 24] assume that the server does not collude with other parties, but this assumption may not hold in practical VFL scenarios. Other related works [31, 67] rely solely on a naive split learning framework, which is vulnerable to emerging attacks [25, 38, 50, 61] in VFL. In contrast, our PS-MI can effectively ensure the security of features, labels, and identities.

- Local features on each participant are fully protected since they are never shared with other parties. In PS-MI, each participant only sends the partial distances $\widetilde{d^p}$ computed over the transformed features. Even if some attackers attempt to reconstruct

Table 3: Statistics of the experiment datasets.

Datasets	Bank	Adult	Web	Heart	SUSY
Instances	10,000	32,561	64,700	253,661	1,000,000
Features	11	123	300	21	18

local features from the transmitted distances, they can access only the privacy-preserving features, as discussed in Section 3.2.

- Label is protected against any malicious party since the labels are maintained by the leader and never shared with others. However, if the leader colludes with others, the labels are naturally leaked.
- Identities are guaranteed against any malicious party due to the pseudo-identities, as discussed in Section 4.1.

5 EXPERIMENTS

We outline our experimental settings in Section 5.1, compare PS-MI with state-of-the-art baselines in Section 5.2, evaluate the designs of PS-MI in Section 5.3, and perform an ablation study in Section 5.4.

5.1 Experiment Settings

Dataset. We conduct extensive experiments on various real-world datasets, as detailed in Table 4. The evaluated datasets are collected from existing works [24, 54] and online repositories [8, 27]. Each dataset is randomly partitioned into a training set (70%) and a test set (30%). We randomly split each dataset into four vertical partitions and put each partition on a physical machine.

Baselines We compare our PS-MI with the following baselines, each of which is the contribution estimation method in VFL. ❶ VFMI-SV uses MI as the utility and computes the Shapley valueby evaluating all possible coalitions with the KNN-MI estimator. It applies homomorphic encryption for data ❷ VF-PS [24] estimates MI for participant groups using the KNN-MI estimator and assigns contributions based on group averages. ❸ VF-CE [31] applies a scalar-level attention mechanism within a mutual information neural network to measure the contributions. ❹ DIG-FL [54] uses logistic regression as the proxy and approximates Shapley values using the Hessian matrix during training. ❺ VFDV-IM [67] proposes to utilize the historical training logs to accelerate one single evaluation. It utilizes test accuracy as the utility function and requires evaluating all possible coalitions. Note that VF-CE and VFDV-IM adopt the naive split learning framework. As discussed in Section 4.5, they can not defend against the emerging attacks on VFL s models.

Metrics. We evaluate our PS-MI using three key metrics: *estimation precision*, *end-to-end running cost*, and *downstream task performance*. The estimation precision is quantified by the Pearson correlation coefficient (PCC) between the Shapley value estimated by the baseline and the actual Shapley value. The end-to-end running cost is quantified by the time (seconds) that the algorithm runs. The downstream task performance is quantified by the test accuracy of downstream machine learning models.

Implementation. For communication between parties, we implement RPC communication with proto3 and gRPC. TENSEAL [2] is a homomorphic encryption library built on top of Microsoft SEAL.

Table 4: Correlation between the estimated and the actual Shapley values. The actual Shapley value is computed by performing $2^P - 1$ retraining (P is the number of participants), using the same utility function as the corresponding baseline. The first and second highest correlations are highlighted in bold and underlined respectively. Note that \ denotes it cannot finish within a reasonable time limit (48h).

Method	Bank	Adult	Web	Heart	SUSY
VFMI-SV	0.9865	**0.9956**	**0.9876**	\	\
VF-PS	0.6922	0.4395	0.5355	0.7421	0.8450
VF-CE	0.4508	0.9732	0.4860	0.8864	0.9182
DIG-FL	0.8894	0.9519	0.6617	0.4255	\
VFDV-IM	0.9352	0.9922	0.4416	0.9188	0.9211
PS-MI	**0.9882**	0.9922	0.9700	**0.9233**	**0.9304**

We adopt Adam [28] as the optimization algorithm for the logistic regression model and the multi-layer perception model. We set the batch size to 32 and terminate each task after 100 epochs. In our PS-MI, the query set comprises 30% of each dataset. We set the number of nearest samples k to 5, the projection dimension size to 5, the threshold for adaptive coalition termination to 30% of query set. All experiments are conducted on Amazon AWS, with each party deployed on separate g4dn.xlarge EC2 GPU instances.

5.2 Main Results

In this section, we evaluate our PS-MI from three perspectives: estimation precision, efficiency, and effectiveness.

Estimation Accuracy. We first study the following question: *How does the correlation between the contributions estimated by the baseline methods and the actual Shapley value compare?* Table 4 reports the Pearson correlation coefficient (PCC) between participants' contributions estimated by baselines and the actual Shapley value. Among all methods, VFMI-SV achieves a strong correlation with the ground-truth. This is expected, as it evaluates all possible coalitions. However, it brings high computation, making it impractical for larger datasets (e.g., it falis on Heart and SUSY within a reasonable time). In comparison, PS-MI reaches similar or better accuracy. For example, PCC between PS-MI estimates and actual values reaches >0.97 on Bank, Adult, and Web and >0.92 on Heart, SUSY. Notably, *the predicted rankings by PS-MI are identical to the ground-truth rankings.* Other Shapley value-based methods, such as DIG-FL and VFDV-IM, prioritize efficiency but suffer from accuracy degradation. Individual-based methods VF-PS and VF-CE focus only on individual contribution. They ignore marginal utility from coalitions, and lack theoretical guarantees for fair valuation.

Efficiency. We then turn to another question: *can our proposed method outperform baselines regarding the end-to-end running cost?* Figure 7 reports the running time of our PS-MI and the baselines. We can find our PS-MI significantly faster than all baselines.

First, our PS-MI is significantly faster than the Shapley value-based methods VFMI-SV, DIG-FL, and VFDV-IM. Unsurprisingly, VFMI-SV is the slowest across all datasets. The poor performance of VFMI-SV is attributed to two reasons: repeated $2^P - 1$ evaluation model retraining and the high cost of homomorphic encryption

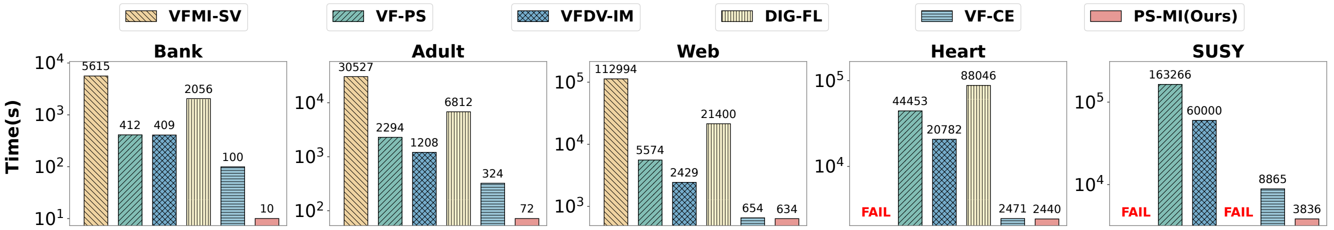


Figure 7: PS-MI vs. the baselines regarding running time. FAIL denotes that it cannot finish within a reasonable time limit (48h).

Table 5: Test accuracy on different downstream tasks. We select the top 50% participant based on the estimations of the baselines and involve them train three models (KNN, LR, MLP). The highest test accuracy among the baselines is highlighted in bold.

Method	KNN			LR			MLP		
	Bank	Adult	Heart	Bank	Adult	Heart	Bank	Adult	Heart
VF-PS	0.7752	0.7991	0.8861	0.78	0.8379	**0.9839**	0.819	0.8422	**0.9065**
VF-CE	0.7752	0.7991	**0.889**	0.78	0.8379	**0.9839**	0.8218	0.8422	0.9064
DIG-FL	0.7945	0.7968	**0.889**	0.7988	0.7546	0.9791	0.801	0.7569	0.9064
VFDV-IM	**0.8200**	0.7968	0.8861	**0.8093**	0.8379	0.9834	**0.8563**	0.8422	**0.9065**
PS-MI	**0.8200**	**0.8011**	**0.889**	**0.8093**	**0.8427**	**0.9839**	**0.8563**	**0.8473**	0.9064

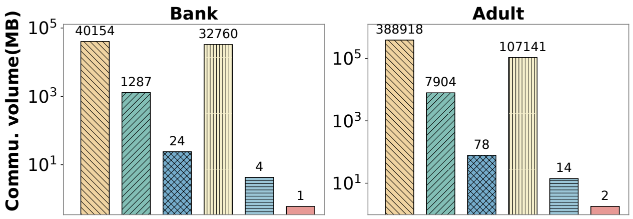


Figure 8: PS-MI vs. baselines regarding communication cost.

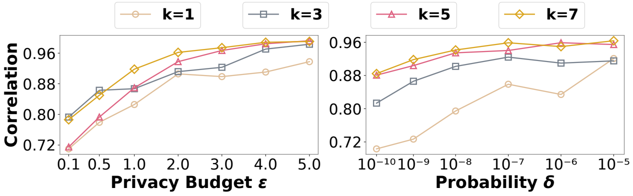


Figure 9: Effect of DP parameters ϵ and δ. k represents the number of nearest samples identified in DP-MI.

(HE) operations. In particular, the speedup of our proposed PS-MI over VFMI-SV is more than 500× on Bank and more than 100× on other datasets. While DIG-FL requires only a single LR model training over all participants, it remains the second slowest due to its reliance on HE for protecting forward outputs. VFDV-IM is also much slower than PS-MI because it necessitates evaluating all possible coalitions. For instance, on Bank, PS-MI is 41× and 45× faster than VFDV-IM and DIG-FL, respectively.

Second, compared to individual-based methods that fail to ensure fairness among data owners, our PS-MI still outperforms both VF-PS and VF-CE. VF-PS is much slower than our PS-MI and VF-CE since it requires measuring the utilities for multiple coalitions and employs HE to protect the transmitted data. For instance, PS-MI achieves a 41× speedup over VF-PS on Bank and 32× on Adult. Meanwhile, VF-CE is slightly slower than PS-MI across some datasets, as it requires only a single training iteration over the entire participant coalition and employs a naive split-learning framework without incorporating additional privacy protection mechanisms.

Figure 8 shows PS-MI also incurs the lowest communication cost, aligned with its runtime efficiency. For instance, it reduces the cost from 4.0×10^4 to 1 on Bank compared to VFMI-SV. We report the full results in our technical report [11].

Effectiveness. To validate the effectiveness, we run the downstream classification tasks using three ML models (KNN, LR, MLP) on three datasets (Bank, Adult, Heart). Specifically, we select the top 50% participants based on the SV estimated by the baselines and involve them train the downstream ML models. Table 5 reports the test accuracy of three downstream ML models. The results show that our proposed PS-MI can achieve high performance in all models across all datasets. For example, on Bank with KNN, our PS-MI achieves 4.5% higher than VF-PS and VF-CE. Notably, VFDV-IM also performs well. This is because it exhaustively evaluates all possible coalitions, which contributes to its high accuracy. However, this comes at a high computational cost—its total evaluation time far exceeds that of training the downstream models. Overall, PS-MI offers both accuracy and efficiency. Its model-agnostic utility ensures robust and unbiased participant selection across architectures.

5.3 Micro Results

Below, we evaluate the designs in our proposed PS-MI.

DP Parameters. Below we evaluate the performance of PS-MI under different privacy settings. Specifically, we vary the differential privacy parameters: the privacy budget ϵ and the probability δ.

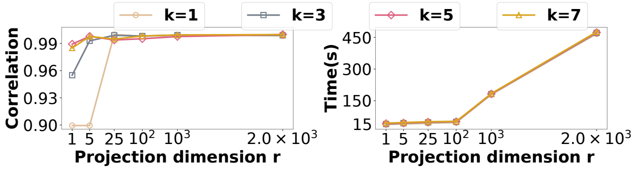


Figure 10: Effect of projection dimension size r. k represents the number of nearest samples identified in DP-MI.

Figure 9 reports the PCC between the estimated and the actual Shapley value. We can observe a key trade-off: as ϵ and δ increase, estimation accuracy improves. This improvement comes at the cost of weaker privacy guarantees, as less noise is added. Notably, PS-MI remains robust even under relatively strong privacy settings. When ϵ ranges from 2 to 4 and δ ranges from 10^{-8} to 10^{-5}, the PCC still exceeds 0.90. This robustness originates from our differentially private mutual information estimator. As proven in Section 3.2, the squared Euclidean distance in the projected feature space serves as an unbiased estimator of distances in the original space.

Projection Dimension Size. Figure 10 illustrates the trade-off between projection dimension r, estimation accuracy, and computational efficiency in PS-MI. We observe that the Pearson correlation coefficient (PCC) plateaus beyond $r > 5$. This aligns with the Johnson-Lindenstrauss lemma [26], where low-dimensional embeddings often preserve pairwise distances sufficiently. Conversely, runtime scales quadratically with r. Critically, PS-MI achieves near-optimal PCC at $r = 5$, with only 27% running time required for $r = 1000$. This highlights the practical advantage of low-to-moderate r in balancing privacy-utility efficiency.

Scalability Evaluation. Below we study the scalability of the baselines. Figure 11 reports the running time of Shapley-based methods with varying numbers of participants (4/8/12/16/20) on Adult and Web. To reduce running time, we apply stratified sampling, with all methods evaluating $2^{P/2}$ coalitions. The results show that PS-MI consistently maintains the fastest execution time. DIG-FL grows slowly in runtime due to Hessian approximation, which requires only one evaluation but yields lower accuracy. Moreover, Hessian approximation depends on sample size, making it unsuitable for large datasets (e.g., it fails on SUSY in Figure 7). In contrast, PS-MI remains efficient even with larger datasets and more participants.

Properties of the Shapley value. Below we simulate two commonly encountered real-world behaviors: data replication and low-quality data. These correspond to two fairness properties of Shapley value: *symmetry* and *zero element*. Table 6 reports the results on Bank and Adult, each partitioned into four participants $\{P_i\}_{i=1}^4$. ❶ *Symmetry: The same contribution brings the same payoff.* We duplicate the data in P_4 to create a new partition P_4' and apply PS-MI to estimate Shapley values for $\{P_i\}_{i=1}^4 \cup \{P_4'\}$. We observe that P_4 and P_4' receive nearly identical Shapley values. For example, on Adult, the values are 0.108 and 0.109 respectively, differing by only 10^{-3}. This confirms that PS-MI approximately preserves the symmetry property. ❷ *Zero Element: No contribution, no payoff.* We simulate low-quality data by assigning meaningless input (e.g., all set to 0) to P_4'. Ideally, this participant should have little to no contribution.

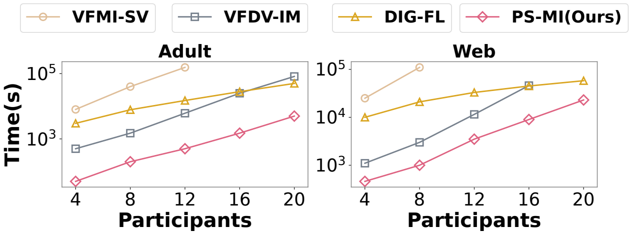


Figure 11: Scalability evaluation. The y-axis is the running time, and the x-axis is the number of participants.

Table 6: Properties of the Shapley value estimated by PS-MI.

Properties	Dataset	P_1	P_2	P_3	P_4	P_4'
Symmetry	Bank	0.333	0.060	-0.035	0.320	0.322
	Adult	0.360	0.289	0.134	0.108	0.109
Zero Element	Bank	0.210	0.358	-0.027	0.776	0.005
	Adult	0.304	0.354	0.123	0.250	-0.003

Table 7: Comparison of sampling strategies on Bank Dataset ($T = 2^{P-1}$). Best-performing method is highlighted in bold.

Method	PCC	MAE	Time(s)
Monte Carlo	0.9522	0.97	122.4
Ours	**0.9836**	**0.75**	**36.1**

As shown in Table 6, the Shapley value assigned to P_4' drops close to zero: 0.005 on Bank and -0.003 on Adult. This shows that PS-MI down-weights uninformative data, satisfying the zero element property. In summary, PS-MI treats identical data sources fairly and effectively filters out unhelpful participants.

5.4 Ablation Study

Below we conduct an ablation study to investigate the impact of sampling strategy, involved samples, and time cost in PS-MI.

Sampling Methods We compare our proposed sampling strategy with the Monte Carlo sampling (MC) method. Table 7 reports PCC and the mean average error (MAE) between the estimated and the actual Shapley value under different sampling strategies. The results show that our proposed method is much faster than the MC method and reaches high estimation accuracy. As discussed in Section 3.3, MC requires evaluating many coalitions, and each marginal contribution only updates one participant's Shapley value. In contrast, our method reuses coalition utilities across participants, enabling more efficient and accurate estimation with fewer evaluations.

Involved Samples. Figure 12 reports the number of samples involved in estimating the Shapley value ($N_C\overline{N_Q}$ in Section 4.4). The results show that our optimized implementation can reduce the involved samples by orders of magnitude, e.g., from 2.03×10^{11} to 8.05×10^8 on the Heart dataset. The naive workflow requires computing all samples' distances for a query and enumerating the

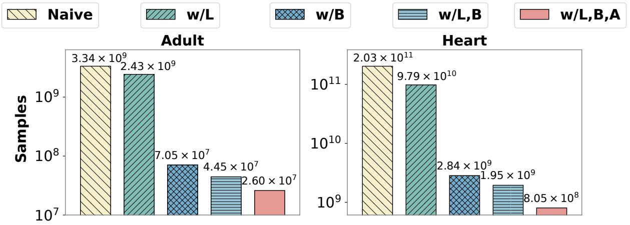


Figure 12: Ablation study on involved samples. (i) candidate pruning with LSH (L); (ii) batched Querying and Hashing (B); (iii) adaptive Termination (A).

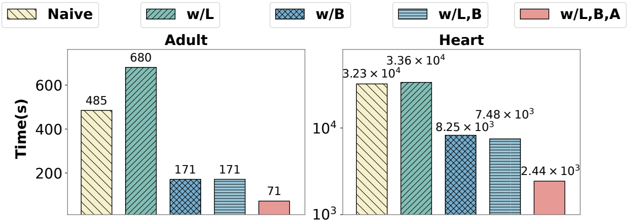


Figure 13: Ablation study on time cost. (i) candidate pruning with LSH (L); (ii) batched Querying and Hashing (B); (iii) Adaptive Termination (A).

entire query set for different coalitions. In contrast, our proposed optimization strategies can avoid repeated computation.

Time Cost. As shown in Figure 13, PS-MI also achieves notable runtime improvements, such as a 13× speedup on Heart compared to the naive workflow. In line with the involved samples, the gains stem from three key optimizations: ❶ the LSH-based candidate pruning reduces the number of distance computations, ❷ the batched candidate checking allows all coalitions to be evaluated in one pass through the query set, ❸ the adaptive termination mechanism dynamically skips unnecessary queries. Notably, naively integrating LSH into the naive workflow increases runtime due to repeated hash table construction per coalition. In contrast, PS-MI amortizes indexing overhead via the batched candidate checking, ensuring LSH accelerates rather than impedes performance.

6 RELATED WORK

Vertical Federated Learning (VFL). VFL enables distributed participants to jointly train ML models over the partitioned features, which has attracted much interests in many real-world cross-enterprise collaborations [1, 37, 41, 43, 56]. Existing methods fall into three categories based on how private features are handled. ❶ Cryptographic works [14, 16, 58] employ the secret sharing scheme and homomorphic encryption to provide desirable privacy guarantees. However, their design involves sophisticated protocols and time-consuming computation primitives to achieve zero knowledge disclosure. ❷ Differential privacy-based works [36, 59, 65] usually randomly perturbs the intermediate data (e.g., features, gradients) via noises. DP inevitably yields a trade-off between privacy guarantee and utility guarantee. ❸ Split learning-based works [3, 51] use different local bottom models for the participants to process private features, and then aggregate the forward activations to make predictions (i.e., via the top model). However, they suffer several data leakage problems and fail to convey provable security guarantees [15, 16, 50]. Some works apply cryptographic methods [19, 32] and differential privacy to protect the intermediate results.

Data Valuation in Vertical Federated Learning. Due to the great potential for applications, various efforts have been devoted to developing concepts of data value in VFL. Jiang et al. [24] propose VF-PS, using mutual information (MI) to evaluate participant importance while adopting homomorphic encryption for privacy protection. Li et al.[31] introduce a mutual information estimator based on split learning with scalar-level attention as a proxy for contribution. Although efficient, both methods overlook marginal contributions and fairness, and often sacrifice accuracy due to simplified heuristics. To improve fairness, Wang et al. [53] and Han et al. [18] employ the Shapley value (SV) to capture participant contributions. However, their methods require retraining for each coalition and rely on access to feature or label distributions, incurring high computational overhead and privacy risks. Wang et al.[54] propose DIG-FL to approximate SV during training, avoiding repeated retraining. But DIG-FL still suffers from accuracy degradation and runtime overhead due to heavy use of homomorphic encryption. Zhou et al.[67] improved efficiency by reusing historical training logs, but their method still evaluates all coalitions and lacks privacy preservation. Existing methods optimize accuracy, efficiency, or privacy individually, but none achieve all three. To our knowledge, our work is the first to achieve these three goals simultaneously in vertical federated data valuation.

7 CONCLUSION AND FUTURE WORK

In this work, we propose a vertical federated data valuation framework, which achieves estimation accuracy, execution efficiency, and data privacy at the same time. We theoretically formalize utility using mutual information and design a novel differential privacy mutual information estimator (DP-MI) for accurate and private estimation. For practical Shapley value computation, we further design stratified sampling, LSH-based pruning, batched candidate sharing, and adaptive termination for efficient execution. Experimental results validate the accuracy, efficiency, and effectiveness of PS-MI. One limitation of this study is that our PS-MI may overlook the importance of individual features within participants' data. Since our focus is on quantifying the value of groups of features (i.e., per-participant), we may miss fine-grained contributions at the feature level. An important direction for future work is to develop valuation strategies that can simultaneously measure the contributions at both the individual feature and participant levels.

ACKNOWLEDGMENTS

This work was sponsored by Key R&D Program of Hubei Province (2023BAB077), National Natural Science Foundation of China (62472-327) and the Fundamental Research Funds for the Central Universities (2042025kf0040). This work was supported by Sichuan Clinical Research Center for Imaging Medicine (YXYX2402). Xiaokai Zhou and Xiao Yan contributed equally to this work. Jiawei Jiang and Tieyun Qian are the corresponding authors.

REFERENCES

[1] Baidu. 2023. PaddleFL: Federated Deep Learning in PaddlePaddle. https://github.com/PaddlePaddle/PaddleFL. Accessed: 2024-12.

[2] Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. 2021. TenSEAL: a library for encrypted tensor operations using homomorphic encryption. *arXiv preprint arXiv:2104.03152* (2021).

[3] Timothy Castiglia, Yi Zhou, Shiqiang Wang, Swanand Kadhe, Nathalie Baracaldo, and Stacy Patterson. 2023. Less-vfl: communication-efficient feature selection for vertical federated learning. In *International Conference on Machine Learning (ICML)*. PMLR, 3757–3781.

[4] Amnon Catav, Boyang Fu, Yazeed Zoabi, Ahuva Libi Weiss Meilik, Noam Shomron, Jason Ernst, Sriram Sankararaman, and Ran Gilad-Bachrach. 2021. Marginal contribution feature importance - an axiomatic approach for explaining data. In *International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research)*, Marina Meila and Tong Zhang (Eds.), Vol. 139. PMLR, 1324–1335.

[5] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast private set intersection from homomorphic encryption. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1243–1255.

[6] Yiwei Chen, Kaiyu Li, Guoliang Li, and Yong Wang. 2024. Contributions estimation in federated learning:a comprehensive experimental evaluation. *Proceedings of the VLDB Endowment (VLDB)* 17, 8 (2024), 2077–2090.

[7] Ian C. Covert, Scott Lundberg, and Su-In Lee. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems (NIPS)* (2020).

[8] D. Dua and C. Graff. 2017. UCI machine learning repository. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu/ml.

[9] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-Eurocrypt 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*. 265–284.

[11] Xiaokai Zhou et.al. 2025. Supplementary. https://drive.google.com/file/d/1oWaU8utMGjc2qBwwccej26XlWosHNn7G/view?usp=sharing.

[12] Ronald Fagin. 1996. Combining fuzzy information from multiple systems. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*. 216–226.

[13] Fangcheng Fu, Xupeng Miao, Jiawei Jiang, Huanran Xue, and Bin Cui. 2022. Towards communication-efficient vertical federated learning training via cache-enabled local updates. *Proceedings of the VLDB Endowment (VLDB)* 15, 10 (2022).

[14] Fangcheng Fu, Yingxia Shao, Lele Yu, Jiawei Jiang, Huanran Xue, Yangyu Tao, and Bin Cui. 2021. VF2Boost: very fast vertical federated gradient boosting for cross-enterprise learning. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. Association for Computing Machinery, 563–576.

[15] Fangcheng Fu, Xuanyu Wang, Jiawei Jiang, Huanran Xue, and Bui Cui. 2024. ProjPert: projection-based perturbation for label protection in split learning based vertical federated learning. *IEEE Transactions on Knowledge and Data Engineering(TKDE)* (2024).

[16] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. 2022. Blindfl: vertical federated machine learning without peeking into your data. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. 1316–1330.

[17] Rui Fu, Yuncheng Wu, Quanqing Xu, and Meihui Zhang. 2023. FEAST: a communication-efficient federated feature selection framework for relational data. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, Vol. 1. 1–28.

[18] Xiao Han, Leye Wang, and Junjie Wu. 2021. Data valuation for vertical federated learning: an information-theoretic approach. *arXiv preprint arXiv:2112.08364* (2021).

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

[20] Yifei He, Runxiang Cheng, Gargi Balasubramaniam, Yao-Hung Hubert Tsai, and Han Zhao. 2024. Efficient modality selection in multimodal learning. *Journal of Machine Learning Research (JMLR)* 25, 47 (2024), 1–39.

[21] Jingwei Hu, Yongjun Zhao, Benjamin Hong Meng Tan, Khin Mi Mi Aung, and Huaxiong Wang. 2024. Enabling threshold functionality for private set intersection protocols in cloud computing. *IEEE Transactions on Information Forensics and Security (TIFS)* 19 (2024), 6184–6196.

[22] Jiahui Huang, Lan Zhang, Anran Li, Haoran Cheng, Jiexin Xu, and Hongmei Song. 2023. Adaptive and efficient participant selection in vertical federated learning. In *2023 19th International Conference on Mobility, Sensing and Networking (MSN)*. 455–462.

[23] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, and Dawn Song. 2019. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment (VLDB)* 12, 11 (2019), 1610–1623.

[24] Jiawei Jiang, Lukas Burkhalter, Fangcheng Fu, Bolin Ding, Bo Du, Anwar Hithnawi, Bo Li, and Ce Zhang. 2022. Vf-ps: how to select important participants in vertical federated learning, efficiently and securely? *Advances in Neural Information Processing Systems (NIPS)* 35 (2022), 2088–2101.

[25] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. 2021. Cafe: catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems (NIPS)* 34 (2021), 994–1006.

[26] William B. Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.* 26, 1 (1984), 189–206.

[27] Kaggle. 2024. Kaggle: your home for data science. Kaggle. Available online: https://www.kaggle.com/datasets.

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. 2025. Approximating the shapley value without marginal contributions. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Article 1477 (2025), 10 pages.

[30] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69, 6 (2004), 066138.

[31] Juan Li, Rui Deng, Tianzi Zang, Mingqi Kong, and Kun Zhu. 2024. Efficient and Secure Contribution Estimation in Vertical Federated Learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM) (CIKM '24)*. Association for Computing Machinery, 1205–1214.

[32] Songze Li, Duanyi Yao, and Jin Liu. 2023. Fedvs: straggler-resilient and privacy-preserving vertical federated learning for split models. In *International Conference on Machine Learning (ICML)*. PMLR, 20296–20311.

[33] Weida Li and Yaoliang Yu. 2024. Robust data valuation with weighted banzhaf values. In *Advances in Neural Information Processing Systems (NIPS)* (New Orleans, LA, USA). Curran Associates Inc., Article 2637, 35 pages.

[34] Xiaochen Li, Yuke Hu, Weiran Liu, Hanwen Feng, Li Peng, Yuan Hong, Kui Ren, and Zhan Qin. 2022. OpBoost: a vertical federated tree boosting framework based on order-preserving desensitization. *Proceedings of the VLDB Endowment (VLDB)* 16, 2 (oct 2022), 202–215.

[35] Zitao Li, Bolin Ding, Ce Zhang, Ninghui Li, and Jingren Zhou. 2021. Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment (VLDB)* 15, 4 (2021).

[36] Zitao Li, Tianhao Wang, and Ninghui Li. 2023. Differentially private vertical federated clustering. *Proceedings of the VLDB Endowment (VLDB)* 16, 6 (2023), 1277–1290.

[37] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. 2024. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 36, 7 (2024), 3615–3634.

[38] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2021. Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 181–192.

[39] Sasan Maleki. 2015. *Addressing the computational issues of the Shapley value with applications in the smart grid.* Ph.D. Dissertation. University of Southampton.

[40] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the estimation error of sampling-based Shapley value approximation. *arXiv preprint arXiv:1306.4265* (2013).

[41] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *2017 IEEE Symposium on Security and Privacy (SP)*. 19–38.

[42] Jerzy Neyman. 1992. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics: Methodology and distribution*. Springer, 123–150.

[43] OpenMined. 2023. PySyft: A library for answering questions using data you cannot see. https://github.com/OpenMined/PySyft. Accessed: 2024-12.

[44] Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. 2010. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. *Advances in Neural Information Processing Systems (NIPS)* 23 (2010).

[45] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2018. Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS)* 21, 2 (2018), 1–35.

[46] Tiberiu Popoviciu. 1935. Sur l'approximation des fonctions convexes d'ordre supérieur. *Mathematica (Cluj)* 10 (1935), 49–54.

[47] Srinivasan Raghuraman and Peter Rindal. 2022. Blazing fast PSI from improved OKVS and subfield VOLE. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 2505–2517.

[48] Alvin E Roth. 1988. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.

[49] Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, and Vijay Chidambaram. 2019. Understanding and benchmarking the impact of gdpr on

database systems. *Proceedings of the VLDB Endowment (VLDB)* (2019), 1064–1077.

[50] Takumi Suimon, Yuki Koizumi, Junji Takemasa, and Toru Hasegawa. 2024. A data reconstruction attack against vertical federated learning based on knowledge transfer. In *IEEE Conference on Computer Communications (INFOCOM)*. 1–6.

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

[52] Paul Voigt and Axel von dem Bussche. 2017. *The EU general data protection regulation (gdpr): a practical guide.* Springer Publishing Company, Incorporated.

[53] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. In *IEEE International Conference on Big Data (Big Data)*. 2597–2604.

[54] Junhao Wang, Lan Zhang, Anran Li, Xuanke You, and Haoran Cheng. 2022. Efficient participant contribution evaluation for horizontal and vertical federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 911–923.

[55] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. *A principled approach to data valuation for federated learning.* Springer International Publishing, 153–167.

[56] Webank. 2019. FATE: Federated AI Technology Enabler. https://github.com/FederatedAI/FATE. Accessed: 2024-12.

[57] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy preserving vertical federated learning for tree-based models. *Proceedings of the VLDB Endowment (VLDB)* 13, 12 (2020), 2090–2103.

[58] Yuncheng Wu, Naili Xing, Gang Chen, Tien Tuan Anh Dinh, Zhaojing Luo, Beng Chin Ooi, Xiaokui Xiao, and Meihui Zhang. 2023. Falcon: a privacy-preserving and interpretable vertical federated learning system. *Proceedings of the VLDB Endowment (VLDB)* 16, 10 (2023), 2471–2484.

[59] Zihang Xiang, Tianhao Wang, Wanyu Lin, and Di Wang. 2023. Practical differentially private and byzantine-resilient federated learning. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, Vol. 1. 1–26.

[60] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2017. DPPro: differentially private high-dimensional data release via random projection. *IEEE Transactions on Information Forensics and Security* 12, 12 (2017), 3081–3093.

[61] Xiangrui Xu, Wei Wang, Zheng Chen, Bin Wang, Chao Li, Li Duan, Zhen Han, and Yufei Han. 2024. Finding the piste: towards understanding privacy leaks in vertical federated learning systems. *IEEE Transactions on Dependable and Secure Computing(TDSC)* (2024).

[62] Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data valuation using reinforcement Llarning. In *International Conference on Machine Learning (ICML)*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. 10842–10851.

[63] Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Efficient sampling approaches to Shapley Value approximation. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, Vol. 1.

[64] Jiayao Zhang, Haocheng Xia, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Dynamic Shapley value computation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. 639–652.

[65] Fangyuan Zhao, Zitao Li, Xuebin Ren, Bolin Ding, Shusen Yang, and Yaliang Li. 2024. VertiMRF: differentially private vertical federated data synthesis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. Association for Computing Machinery, 4431–4442.

[66] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. 2023. Secure shapley value for cross-silo federated learning. *Proceedings of the VLDB Endowment (VLDB)* 16, 7 (2023).

[67] Xiaokai Zhou, Xiao Yan, Xinyan Li, Hao Huang, Quanqing Xu, Qinbo Zhang, Yen Jerome, Zhaohui Cai, and Jiawei Jiang. 2024. VFDV-IM: an efficient and securely vertical federated data valuation. In *International Conference on Database Systems for Advanced Applications (DASFAA)*. 409–424.