Efficient Data Valuation Approximation in Federated Learning: A Sampling-based Approach

Shuyue Wei[1], Yongxin Tong[1], Zimu Zhou[2], Tianran He[1], Yi Xu[1]
[1] State Key Laboratory of Complex & Critical Software Environment Lab, School of Computer Science and Engineering,
Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, China
[2] City University of Hong Kong, Hong Kong, China
[1]{weishuyue, yxtong, hetianran, xuy}@buaa.edu.cn, [2]zimuzhou@cityu.edu.hk

***Abstract*— Federated learning (FL) has emerged as a prominent distributed learning paradigm to utilize datasets across multiple data providers. In FL, cross-silo data providers often hesitate to share their high-quality dataset unless their data value can be fairly assessed. Shapley value (SV) has been advocated as the standard metric for data valuation in FL due to its desirable properties. However, the computational overhead of SV is prohibitive in practice, as it inherently requires training and evaluating an FL model across an exponential number of dataset combinations. Furthermore, existing solutions fail to achieve high accuracy and efficiency, making practical use of SV still out of reach, because they ignore choosing suitable computation scheme for approximation framework and overlook the property of utility function in FL. We first propose a unified stratified-sampling framework for two widely-used schemes. Then, we analyze and choose the more promising scheme under the FL linear regression assumption. After that, we identify a phenomenon termed key combinations, where only limited dataset combinations have a high-impact on final data value. Building on these insights, we propose a practical approximation algorithm, *IPSS*, which strategically selects high-impact dataset combinations rather than evaluating all possible combinations, thus substantially reducing time cost with minor approximation error. Furthermore, we conduct extensive evaluations on the FL benchmark datasets to demonstrate that our proposed algorithm outperforms a series of representative baselines in terms of efficiency and effectiveness.**

I. Introduction

In recent years, *federated learning* (FL) has gained increasing attention in both academia and industry, as it provides a novel solution to utilize datasets across multiple data-rich entities without directly accessing raw data [1]–[5]. In FL, cross-silo data providers may be reluctant to share high-quality datasets unless the value of their datasets are fairly measured, ensuring they receive appropriate compensation [6]–[8]. Therefore, data valuation is a fundamental problem in FL, as it is the prerequisite for motivating multiple data providers to contribute, thereby serving as a crucial component for the sustainability of the FL ecosystem.

The *Shapley value* (SV), a classical concept for measuring the player's contribution in cooperation, has been considered as the standard data valuation metric for FL in prior work [6]–[13], as it uniquely satisfies several basic fairness properties [14] (*e.g.*, *no-free-riders*, *symmetric fairness* and *linear additivity*). However, the SV-based data valuation is widely-acknowledged computationally prohibitive due to its intrinsic combinatorial nature [15], [16], *i.e.*, we have to train and evaluate FL models across an exponential number of combinations of datasets, as the toy example in Fig. 1(a).

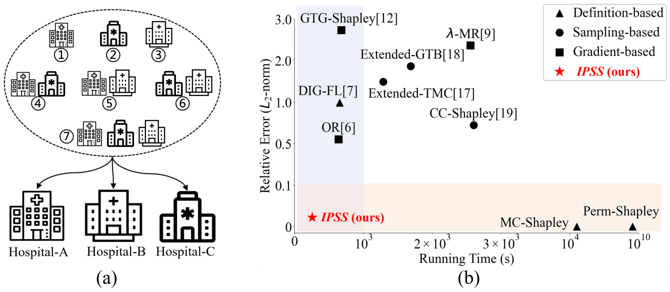


Fig. 1: (a):Three hospitals collaborate to train the FL model and aim to identify each hospital's data value. The SV-based data valuation requires training and evaluating FL models across all possible hospital combinations (①∼⑦), *i.e.*, it needs to tackle seven FL processes. As client number increases, the number of required combinations grows exponentially. (b):Evaluations on the FL benchmark dataset FEMNIST with ten FL clients indicate that the existing solutions fail to achieve both high effectiveness and efficiency simultaneously.

The SV-based data valuation has attracted extensive research interest from both the database and the data mining communities, where researchers prioritize efficiency as a central issue and devise a line of approximation algorithms [6]–[12]. Existing solutions can primarily be divided into two categories. *(i) The first category is the gradient-based approximation* [6], [9], [12], which utilizes gradients in the training process to construct the FL models, that are required to be evaluated in data valuation, thereby avoiding extra FL training processes. Though these solutions provide notable computational efficiency, their lack of accuracy guarantees diminishes their practicality (as in Fig. 1(b)). *(ii) The second category is the sampling-based approximation* [17]–[19], which only trains and evaluates FL models under a few number of dataset combinations chosen from all possible combinations to estimate the data value. Recently, more studies [15]–[19] have advocated for sampling-based approaches as they provide a flexible trade-off between accuracy and efficiency. However, prior sampling-based approximations still fail to achieve both high efficiency and accuracy simultaneously in FL, as shown in Fig. 1(b), primarily due to the following two limitations.

- *Limitation 1*: *Ignoring selecting the suitable computation*

2375-026X/25/$31.00 ©2025 IEEE
DOI 10.1109/ICDE65448.2025.00219

scheme of Shapley value. There are two commonly used equivalent expressions of the Shapley value, including the marginal-contribution-based (*MC-SV*) and the complementary-contribution-based (*CC-SV*), each provides a computation scheme by its definition. Choosing a more suitable scheme for the approximation framework is often overlooked. As shown in Sec. III-B, taking the *MC-SV* for our sampling framework can yield a lower variance in the approximation.

- *Limitation 2*: *Ignoring utilizing intrinsic properties of utility function in federated learning*. In SV-based data valuation within FL, we usually set the utility function to model accuracy, which is different from that in traditional game theory. For example, the utility function in *weighted majority game* exhibits a binary jump, making it #$\mathcal{P}$-hard [20] in this scenario. In contrast, the utility in FL (*e.g.*, model accuracy) typically exhibits a monotonic property as more clients join, providing it with unique features for data valuation.

To address these limitations, we propose an efficient and effective sampling-based approximation algorithm, *Importance-Pruned Stratified Sampling (IPSS)*. Specifically, we first propose a unified stratified sampling framework, which can seamlessly integrate both the *MC-SV*-based and the *CC-SV*-based computation schemes. Then, we compare two computation schemes consistently in this framework under the assumption of FL linear regression model and then choose the *MC-SV* for further investigation. We also identify a phenomenon in data valuation for FL through observations and empirical studies on utility function, referred to as *key combinations*, which shows that different dataset combinations in FL have varying impacts on the final data value. Finally, we propose an approximation algorithm, *IPSS*, which prunes data combinations with minimal impacts from all possible combinations, significantly reducing time costs while maintaining accuracy. Main contributions of this work are summarized as follows.

- We propose a unified stratified sampling based framework that can integrate with both the *MC-SV*-based and the *CC-SV*-based computation schemes and then compare them to select the most promising one for further study.
- We identify a phenomenon called *key combinations*, *i.e.*, data valuation in FL can be approximated accurately by utilizing only a small group of dataset combinations instead of all possible exponential combinations.
- We devise an efficient and effective algorithm, *IPSS*, tailored for the *MC-SV*-based data valuation in FL and then analyze its approximation error and time complexity.
- We conduct extensive experiments and compare our proposed *IPSS* algorithm with a series of baselines to validate its superiority in time cost and approximation error on both synthetic and benchmark datasets in FL.

In the rest of this paper, we first introduce the basic concepts of the data valuation in FL in Sec. II. Then, we propose a sampling-based framework and compare computation schemes in Sec. III. In Sec. IV, we identify the key combinations phenomenon and introduce the proposed *IPSS* algorithm. Finally, we present experimental evaluations in Sec. V, review the related work in Sec. VI, and conclude this work in Sec. VII.

II. Problem Statement

In this section, we first present basic concepts of the *federated learning* (FL) and the *data valuation* problem in the context of FL. Then, we introduce two equivalent computation schemes for the *Shapley value* (SV) based data valuation.

A. Preliminary and Problem Definition

Definition 1 (***Federated learning, FL***). *FL is a distributed learning paradigm that enables multiple data providers to utilize massive training data for the data-driven tasks without directly accessing raw data. [1]–[4]. In FL, there are n data owners (a.k.a. FL clients), each with a dataset $\mathcal{D}_i$, and a coordinator (a.k.a. FL server) and they aim to jointly train a learning model $M_N(\mathcal{A})$ across datasets from all clients through a FL algorithm $\mathcal{A}$, where $N = \{1, \ldots, n\}$.*

We take the most widely-used FL algorithm, *FedAvg* [5], as an example to illustrate the main process of FL. A FL algorithm $\mathcal{A}$ operates iteratively at the FL server and clients as follows. *(i) Acts at server*: In the first iteration, the FL server initializes and distributes the global model to all clients. Otherwise, the FL server obtains a new global model by aggregating all local models in previous iteration from the clients. *(ii) Acts at clients*: Take the client i as an example. The client i trains the received model on its local dataset $\mathcal{D}_i$ and then uploads an updated local model to the FL server. The FL algorithm executes above two steps alternately until the required converge criterion or training round is reached.

Definition 2 (***Data valuation for FL***). *Given n datasets $\mathcal{D}_N = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$ and a FL algorithm $\mathcal{A}$, the federation trains model $M_S(\mathcal{A})$ (or simply M_S) under a subset of clients $S \subseteq N$ and evaluates its utility as $U(M_S)$ on the test dataset $\mathcal{T}$, where utility function $U(\cdot)$ is defined as model performance (e.g., accuracy). Then, data valuation problem aims to quantify contribution of each dataset $\mathcal{D}_i$ as $\phi(\mathcal{A}, \mathcal{D}_N, \mathcal{T}, \mathcal{D}_i)$ (ϕ_i for short), satisfying the following three basic properties,*

- *(i) null-player (or no-free-riders)*: If a dataset $\mathcal{D}_j$ is irrelevant to FL model $M_S(\mathcal{A})$ on test dataset $\mathcal{T}$ for any dataset combination $\mathcal{D}_S$, ϕ_j should be zero. Formally,

$$\forall S \subseteq N, U(M_S) = U(M_{S\cup\{j\}}) \Rightarrow \phi_j = 0, \qquad (1)$$

- *(ii) symmetric-fairness:* If two datasets $\mathcal{D}_i$ and $\mathcal{D}_j$ have the same effect on FL model $M_S(\mathcal{A})$ on the test dataset $\mathcal{T}$, their value in FL should be the same as well. Formally,

$$\forall S \subseteq N \backslash \{i, j\}, U(M_{S\cup\{i\}}) = U(M_{S\cup\{j\}}) \Rightarrow \phi_i = \phi_j, \qquad (2)$$

- *(iii) linear-additivity:* The data value for FL is linear with respect to two disjoint test dataset $\mathcal{T}_1$ and $\mathcal{T}_2$. Formally,

$$\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset \Rightarrow \forall i \in N, \phi_i(\mathcal{T}_1 \cup \mathcal{T}_2) = \phi_i(\mathcal{T}_1) + \phi_i(\mathcal{T}_2), \qquad (3)$$

where $\phi_i(\mathcal{T}_1)$, $\phi_i(\mathcal{T}_2)$ and $\phi_i(\mathcal{T}_1 \cup \mathcal{T}_2)$ are assigned value to the dataset $\mathcal{D}_i$ using same datasets $\mathcal{D}_N$ and algorithm $\mathcal{A}$.

Remarks. Above three properties are all essential in FL. Firstly, the *null player* is instrumental in identifying free riders who do not contribute to the FL model. The *symmetric fairness* provides a fundamental fairness in FL, *i.e.*, if two datasets are interchangeable, they should have the same data value. Finally, the *linear additivity* ensures that introducing new test data does not alter existing data value, *i.e.*, original data value remains reusable, simplifying the integration of new test data.

B. The Shapley Value based Data Valuation Schemes

If we consider each FL client as a player and model performance as utility function in a collaborative game, then the *Shapley value* (SV), a classical concept to fairly measure the player's contribution, naturally inherits its properties and ensures three desirable properties in Def. 2 [6]. Therefore, the *Shapley value* (SV) has been widely adopted as a standard data valuation metric in FL [6]–[9], [11]–[13], [15]–[18]. Furthermore, there are two commonly used equivalent SV expression, the marginal contribution based (*MC-SV*) and the complementary contribution based (*CC-SV*). Each provides a computation scheme for data valuation by its definition.

Definition 3 (***MC-SV based computation scheme***). *Given n datasets $\mathcal{D}_N = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, a learning algorithm $\mathcal{A}$, test dataset $\mathcal{T}$ and the utility function $U(\cdot)$ in FL, MC-SV computes the data value for each dataset $\phi(\mathcal{A}, \mathcal{T}, \mathcal{D}_N, \mathcal{D}_i)$ as follows,*

$$\phi(\mathcal{A}, \mathcal{D}_N, \mathcal{T}, \mathcal{D}_i) = \sum_{S \subseteq N \setminus \{i\}} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{n \cdot \binom{n-1}{|S|}}, \tag{4}$$

where M_S denotes the FL model trained on dataset combination $\cup_{i \in S} \mathcal{D}_i$ and $|S|$ represents the number of datasets involved in S. The term $\binom{n-1}{|S|} = \frac{(n-1)!}{|S|!(n-1-|S|)!}$ is the combinatorial number. This computation scheme is referred to as the marginal contribution based SV (MC-SV for short), because it is based on the marginal contribution of each FL client i.

Definition 4 (***CC-SV based computation scheme***). *Given datasets $\mathcal{D}_N$, learning algorithm $\mathcal{A}$, test datasets $\mathcal{T}$ and the utility function $U(\cdot)$ in FL, CC-SV computes the dataset $\mathcal{D}_i$'s data value $\phi(\mathcal{A}, \mathcal{T}, \mathcal{D}_N, \mathcal{D}_i)$ in FL as follows,*

$$\phi(\mathcal{A}, \mathcal{T}, \mathcal{D}_N, \mathcal{D}_i) = \sum_{S \subseteq N \setminus \{i\}} \frac{U(M_{S \cup \{i\}}) - U(M_{N \setminus (S \cup \{i\})})}{n \cdot \binom{n-1}{|S|}}, \tag{5}$$

where $U(M_{S \cup \{i\}}) - U(M_{N \setminus (S \cup \{i\})})$ is called the complementary contribution [19] of clients $S \cup \{i\}$ and we use CC-SV as the abbreviation for this computation scheme of Shapley value.

Table II summarizes the major symbols throughout this work.

TABLE I: SV-based data valuation for FL with three clients.

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$U(M_S)$	0.10	0.50	0.70	0.60	0.80	0.90	0.90	0.96

Example 1. *Considering a FL scenario with three clients $N = \{1, 2, 3\}$. The utilities of FL models for each possible client combination are detailed in Table I. We take FL client 1 as an example whose data value is denoted as ϕ_1. In this example, we employ the MC-SV-based computation scheme. The data value ϕ_1 is determined by averaging the marginal contribution of FL client 1 when added to all combinations without it.*

TABLE II: Summary of the major symbol notions.

Notations	Descriptions
N, S	the set of all FL clients and a subset of all clients
$\phi_i, \hat{\phi}_i$	FL client i's data value and its approximated value
$\mathcal{D}_S, \mathcal{D}_i$	datasets of client combination S and of FL client i
$\mathcal{A}, \mathcal{T}$	the training algorithm and the test dataset in FL
M_S	FL model trained over datasets held by clients S
$U(\cdot)$	the utility function in SV-based data valuation
MC-SV	the SV scheme based on marginal contribution
CC-SV	the SV scheme based on complementary contribution
$\mathcal{S}_k$	dataset combinations with datasets of k clients
$\lvert S \rvert$	the number of FL clients involved in combination S
γ	total sampling rounds in approximation algorithm
τ	the time cost for training and testing a FL model

- *For the empty combination $\emptyset$, the marginal contribution of adding the FL client 1 is $U(\{1\}) - U(\emptyset) = 0.40$.*
- *For combinations including one other FL client, the marginal contributions of adding the client 1 are $U(\{1, 2\}) - U(\{2\}) = 0.10$ and $U(\{1, 3\}) - U(\{3\}) = 0.30$.*
- *For the combination with other two FL clients, the marginal contribution is $U(\{1, 2, 3\}) - U(\{2, 3\}) = 0.06$.*

Finally, these contributions are averaged to compute ϕ_1 as $(0.40/1 + (0.10 + 0.30)/2 + 0.06/1)/3 = 0.22$. Similarly, data value of clients 2 and 3 are $\phi_2 \approx 0.32$ and $\phi_3 = 0.32$.

C. Approximations for SV-based Data Valuation in FL

Though Shapley value has been widely adopted as a standard metric [6]–[13], it is acknowledged that *the high computational overhead of Shapley value prohibits its practical use, as it requires evaluating for all possible dataset combinations.* Specifically, using both *MC-SV* and *CC-SV* based schemes needs to train and assess $\mathcal{O}(2^n)$ FL models, which is infeasible in practice. Thus, it is imperative to devise practical approximation methods for the SV-based data valuation in FL that meet the following two critical two requirements.

- *R1: It is necessary to efficiently estimate SV-based data value for FL dataset.* In commercial applications, FL clients may often prioritize their computational resources for model training and deployment. Excessive time cost spent on data valuation in FL should be avoided to ensure the economic interests in real-world scenarios.
- *R2: It is crucial to approximate SV-based data value for FL clients with tolerable errors.* The Shapley value is favored for its desirable fairness properties (*e.g.*, *no free riders and symmetric fairness*). However, substantial approximation errors can undermine these fairness properties, jeopardizing its applicability for data providers.

Roadmap for Approximations. To address the two requirements (*efficiency* and *effectiveness*), existing literature [6]–[12] primarily explore two types of approximation methods: *gradient-based approximation* and *sampling-based approximation*. *(i)* The *gradient-based* approaches [6], [9], [12] utilize the gradients during the FL training process to construct the FL models under various dataset combinations, which avoids training FL models from scratch, thereby reducing the computational time significantly. However, these gradient-based methods usually lack theoretical underpinnings, which can result in higher approximation errors. *(ii)* The *sampling-based* solutions [11], [17], [21] strategically select subsets from entire potential combinations, providing a flexible trade-off between accuracy and efficiency, making it more suited to meet the efficiency and effectiveness requirements. Therefore, this paper advocates for the *sampling-based* approximations and we propose a novel sampling-based approach for practical SV-based data valuation in FL, which is detailed in the following sections (in Sec. III and Sec. IV).

III. Stratified Sampling based Framework

In this section, we first introduce a unified stratified sampling framework that integrates both computation schemes outlined in Sec. II-B. Then, we analyze and choose the *MC-SV*-based computation scheme as the more appropriate choice for the proposed stratified sampling-based framework.

A. The Unified Sampling Framework for SV-based Schemes

By definition, both the *MC-SV* and *CC-SV* possess an inherent hierarchical structure based on the size of dataset combinations (as shown in Fig. 2), where each FL client's data value is calculated by the average marginal (or complementary) contributions across combinations of various sizes. Thus, it is natural to treat dataset combinations of the same size as strata and employ the stratified sampling for approximation. To this end, we devise a unified stratified sampling framework to support both *MC-SV*-based and *CC-SV*-based computation schemes, which is illustrated in Alg. 1. Then we can compare these two schemes in a consistent framework.

Main Idea. Let $\mathcal{S}_k$ denotes all dataset combinations with datasets from k clients and we can use the Monte Carlo method to approximate a stratified-SV $\hat{\phi}_{i,k}$ for dataset combination $\mathcal{S}_k$ with k datasets (*i.e.*, each stratum) and the estimated SV $\hat{\phi}_i$ is the average across all strata. Specifically, Alg. 1 takes as input sampling rounds m_k for the k_{th} stratification ($\gamma = \sum_{k=1}^{n} m_k$), along with n datasets $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and a utility function to measure the performance of trained FL model. In lines 1-8, the framework first randomly samples dataset combinations used for each FL client and tests the utility of FL models for each stratum. Then, the framework calculates the stratified-SV for each combination size and it can be easily integrated with both the *MC-SV*-based and the *CC-SV*-based data valuation computation schemes. For *MC-SV*, two combinations S and $\overline{S}$ are paired when $\overline{S} = S \backslash \{i\}$, whereas in *CC-SV*, S are paired with $\overline{S} = N \backslash S$. Finally, the framework approximates and returns the SV by averaging marginal (or complementary) contributions within each stratum.

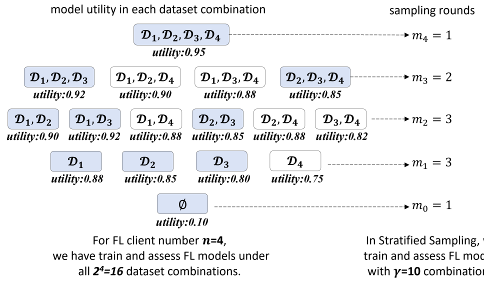

model utility in each dataset combination
sampling rounds
$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$
utility:0.95
$m_4 = 1$
$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$
utility:0.92
$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$
utility:0.90
$\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$
utility:0.85
$m_3 = 2$
$\mathcal{D}_1, \mathcal{D}_2$
utility:0.90
$\mathcal{D}_1, \mathcal{D}_3$
utility:0.92
$\mathcal{D}_1, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_2, \mathcal{D}_3$
utility:0.85
$\mathcal{D}_2, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_3, \mathcal{D}_4$
utility:0.82
$m_2 = 3$
$\mathcal{D}_1$
utility:0.88
$\mathcal{D}_2$
utility:0.85
$\mathcal{D}_3$
utility:0.80
$\mathcal{D}_4$
utility:0.75
$m_1 = 3$
∅
utility:0.10
$m_0 = 1$
For FL client number n=4, we have train and assess FL models under all 2^4=16 dataset combinations.
In Stratified Sampling, we train and assess FL models with γ=10 combinations.

Fig. 2: Example for the unified stratified sampling framework: Both *MC-SV* and *CC-SV* rely on this hierarchical structure, which is naturally suitable for stratified sampling. There are four FL clients and model utility is below each dataset combination. For instance, the utility of FL model under dataset combination $\{\mathcal{D}_1, \mathcal{D}_3\}$ is 0.92.

Algorithm 1: Stratified Sampling Approximation

```
   Input: The $n$ data providers with datasets $\mathcal{D}_N$, a test
          dataset $\mathcal{T}$ and a utility function $U(\cdot)$. Sampling
          rounds for each stratum $m_k$ ($\gamma = \sum_k m_k$).
   Output: Data value of all datasets $\hat{\phi}_1, \cdots, \hat{\phi}_n$.
1  for $k \leftarrow 1$ to $n$ do
2      Initialize $\mathcal{S}_k$ as all combinations with $k$ datasets;
3      $\boldsymbol{S}_k \leftarrow \{S_{k,1}, \ldots, S_{k,m_k}\}$ w.r.t. $S_{k,\cdot} \sim \mathcal{S}_k$;
4      $\boldsymbol{S}_{k,i} \leftarrow \{S | S \in \mathcal{S}_k$ and $i \in S\}$;
5      for $S \in \boldsymbol{S}_k$ do
6          Train and evaluate FL model $M_S$ and then we
             can obtain the model's utility $U(M_S)$ on $\mathcal{T}$;
7      end
8  end
9  for $i \leftarrow 1$ to $n$ do
10     for $S \in \boldsymbol{S}_{k,i}$ do
11         if the paired combination $\overline{S}$ is sampled then
12             $m_{i,k} \leftarrow m_{i,k} + 1$;
13             $\hat{\phi}_{i,k} \leftarrow \hat{\phi}_{i,k} + U(M_S) - U(M_{\overline{S}})$;
14         end
15     end
16 end
17 $\hat{\phi}_i \leftarrow \frac{1}{n} \sum_{k=1}^{n} \frac{\hat{\phi}_{i,k}}{m_{i,k}}$ (where $i = 1, 2, \ldots, n$);
18 return $\hat{\phi}_1, \cdots, \hat{\phi}_n$
```

Example 2. *We illustrate the stratified sampling framework by the example in Fig. 2. We set the total sampling round* $\gamma = 10$ *and sampled dataset combinations are marked in light blue.* *Case 1 (using MC-SV): Set the adopted computation scheme in Alg. 1 to MC-SV, we calculate the average marginal contribution for each stratum in this case. We take the FL client* 1*'s data value as an instance which is* 0.26 *by its definition. For dataset combination* $\mathcal{S}_1$*, we calculate marginal contribution of FL client* 1 *as* $\hat{\phi}_{1,1} = U(M_{\{1\}}) - U(M_{\emptyset}) = 0.78$. *For combinations with two datasets* $\mathcal{S}_2$*, we can calculate the average marginal contribution in this stratum as* $\hat{\phi}_{1,2} = (U(M_{\{1,2\}}) - U(M_{\{2\}}) + U(M_{\{1,3\}}) - U(M_{\{3\}}))/2 = 0.085$. *Similarly, for* $\mathcal{S}_3$ *and* $\mathcal{S}_4$*, we have* $\hat{\phi}_{1,3} = 0.07$ *and* $\hat{\phi}_{1,4} = 0.10$. *Finally, the data value of FL client* 1 *can be approximated by* $\hat{\phi}_1 = (0.78 + 0.085 + 0.07 + 0.10)/4 \approx 0.2588$.

Case 2 (using CC-SV): We take the average complementary contribution for each stratum in this case. For dataset combination with one dataset $\mathcal{S}_1$*, the complementary contribution of FL client* 1 *is* $\hat{\phi}_{1,1} = U(M_{\{1\}}) - U(M_{\{2,3,4\}}) = 0.03$. *For dataset combinations with two datasets* $\mathcal{S}_2$*, the complementary contribution is calculated as* $\hat{\phi}_{1,2} = 0$*, since no paired combination*

$N \backslash S_2$ ($S_2 \in \mathcal{S}_2$) is sampled. Similarly, we have $\hat{\phi}_{1,3} = 0$ and $\hat{\phi}_{1,4} = 0.85$ for $\mathcal{S}_3$ and $\mathcal{S}_4$. Finally, the FL client 1's data value is approximated by $\hat{\phi}_1 = (0.03 + 0 + 0 + 0.85)/4 = 0.22$.

B. Choosing Computation Scheme for Sampling Framework

As mentioned above, the Alg. 1 supports both the *MC-SV*-based and the *CC-SV*-based computation schemes, allowing us to analyze and compare them within a consistent sampling based framework. Since *MC-SV*-based and *CC-SV*-based schemes have the same time complexity of $\mathcal{O}(2^n \tau)$ based on the definition, where τ denotes time cost of training and assessing a FL model. We further compare their performance in expectation and variance using a consistent sampling strategy.

Theorem 1. *The Alg. 1 provides an unbiased estimation of SV in expectation when using both the MC-SV or the CC-SV.*

Proof. We first analyze the expectation of stratified-SV $\hat{\phi}_{i,k}$,

$$\mathbb{E}[\frac{\hat{\phi}_{i,k}}{m_{i,k}}] = \frac{\mathbb{E}_{S \sim \mathcal{S}_{k,i}}[\hat{\phi}_{i,k}]}{m_{i,k}} = \frac{\mathbb{E}_{S \sim \mathcal{S}_{k,i}}[\sum_{t=1}^{m_{i,k}} U(M_S) - U(M_{\overline{S}})]}{m_{i,k}}$$
$$= \mathbb{E}_{S \sim \mathcal{S}_{k,i}}[U(M_S) - U(M_{\overline{S}})] = \sum_{S \subseteq N \backslash \{i\}} \frac{U(M_{S \cup \{i\}}) - U(M_{\overline{S \cup \{i\}}})}{\binom{n-1}{|S|}} \tag{6}$$

Then, the expectation of the SV based on Alg. 1 is,

$$\mathbb{E}[\hat{\phi}_i] = \mathbb{E}[\frac{1}{n} \cdot \sum_{k=1}^{n} \hat{\phi}_{i,k}] = \frac{1}{n} \sum_{S \subseteq N \backslash \{i\}} \frac{U(M_{S \cup \{i\}}) - U(M_{\overline{S \cup \{i\}}})}{\binom{n-1}{|S|}} \tag{7}$$

Thus, by the definition of *MC-SV* and *CC-SV*, Eq. (7) equals the FL client i's data value ϕ_i, completing our proof. □

Theorem 2. *Assume that the data from all providers are all drawn from the same distribution and let $|\mathcal{D}_i|$ denote the size of dataset held by the FL client i. Then for any sampling strategy for CC-SV based scheme, using the MC-SV based scheme can yield a lower variance in Alg. 1 in FL linear regression.*

Proof. Based on the theoretical analysis in [22], the variance of error in a linear regression model applied to a dataset $\mathcal{D}$ with t training samples, can be described as follows,

$$\mathbb{V}ar[U(M_{\mathcal{D}})] = \mathbb{V}ar[\sum_{j=1}^{t} e_j] = \sum_{j=1}^{t} \mathbb{V}ar[|\hat{f}(x_j) - y_j|] = t^2 \sigma^2 \tag{8}$$

where $e_j = |\hat{f}(x_j) - y_j|$ is the mean absolute error on each training sample (x_j, y_j) and σ^2 is the variance of intrinsic random noise in the dataset. If we take negative mean average error as utility and randomly sample a dataset combination S from $N \backslash \{i\}$ to approximate *MC-SV*, its variance $\mathbb{V}ar[\hat{\phi}_i^{MC}]$ is,

$$\mathbb{V}ar[\hat{\phi}_i^{MC}] = \mathbb{V}ar[\frac{1}{n} \sum_{k=1}^{n} \sum_{S \sim (N \backslash \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{m_{i,k}}]$$
$$= \sum_{k=1}^{n} \sum_{S} \frac{\mathbb{V}ar[U(M_{S \cup \{i\}}) - U(M_S)]}{n^2 \cdot m_{i,k}^2}$$
$$= \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} \cdot \mathbb{V}ar[- \sum_{j \in \mathcal{D}_{S \cup \{i\}}} e_j + \sum_{j \in \mathcal{D}_S} e_j] \tag{9}$$
$$= \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} \cdot \mathbb{V}ar[\sum_{j \in \mathcal{D}_i} e_j] = \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} |\mathcal{D}_i|^2 \sigma^2$$

Similarly, the variance of the *CC-SV* can be calculated as,

$$\mathbb{V}ar[\hat{\phi}_i^{CC}] = \mathbb{V}ar[\frac{1}{n} \sum_{k=1}^{n} \sum_{S \sim (N \backslash \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_{N \backslash (S \cup \{i\})})}{m_{i,k}}]$$
$$= \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} \cdot (\mathbb{V}ar[\sum_{j \in \mathcal{D}_{S \cup \{i\}}} e_j] + \mathbb{V}ar[\sum_{j \in \mathcal{D}_{N \backslash (S \cup \{i\})}} e_j])$$
$$= \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} ((|\mathcal{D}_S| + |D_i|)^2 + (|\mathcal{D}_N| - |\mathcal{D}_S| - |\mathcal{D}_i|)^2) \sigma^2 \tag{10}$$

If they take the same sampling strategy in approximation, we can compare their variances by (10) - (9) as follows,

$$\mathbb{V}ar[\hat{\phi}_i^{CC}] - \mathbb{V}ar[\hat{\phi}_i^{MC}] \geq \sum_{k=1}^{n} \sum_{S} \frac{1}{n^2 m_{i,k}^2} |\mathcal{D}_S|^2 \sigma^2 > 0 \tag{11}$$

Therefore, the variance of *MC-SV* is lower than that of *CC-SV* when using Alg. 1 and we finish the proof of Theorem 2. □

Takeaways. Based on above theoretical analysis, we have two main results: *(i) when implemented within Alg. 1, MC-SV-based and CC-SV-based computation schemes can both provide unbiased estimations for data valuation. (ii) for each stratified sampling strategy of CC-SV, there exists a corresponding strategy of MC-SV that yields lower variance in the context of FL linear model.* It is important to note that Alg. 1 operates as a stratified sampling framework without imposing specific assumptions on the number of sampling rounds m_i for each stratum. Thus, our analysis provides broadly applicable evidence when comparing the two schemes and justifying our choice of the *MC-SV* based computation scheme.

IV. Importance-Pruned Stratified Sampling

In this section, we explore the *MC-SV*-scheme within our stratified sampling framework to devise a practical approximation algorithm. Initially, we further observe the *MC-SV*-based computation scheme and identify a phenomenon, called *key combinations*, *i.e., it is sufficient to focus on only a selected subset of all 2^n dataset combinations to approximate the SV in FL.* Armed with this knowledge, we can selectively prune the less significant dataset combinations to optimize efficiency and accuracy for the *MC-SV*-based data valuation in FL.

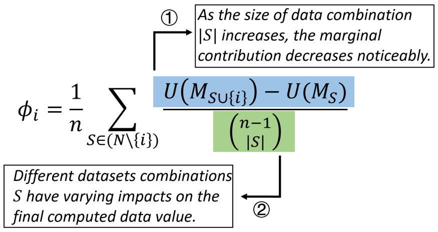

①
As the size of data combination |S| increases, the marginal contribution decreases noticeably.
$\phi_i = \frac{1}{n} \sum_{S \in (N \backslash \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}$
Different datasets combinations S have varying impacts on the final computed data value.
②

Fig. 3: Observations when using the *MC-SV*-based scheme.

A. Identifying the Key Combinations Phenomenon

Observations. It is essential to utilize the inherent properties of the *MC-SV* for effective and efficient data valuation in FL. To this end, we recall and examine the *MC-SV*-based computation scheme as depicted in Fig. 3. This analysis reveals that different dataset combinations S have varying impacts on the data value ϕ_i primarily in two aspects:

- *(i)* The marginal utility of FL model decreases with the addition of more datasets. Once there are already sufficient datasets for training in FL, adding a new dataset can only improves limited utility, *i.e.*, when $\mathcal{D}_S$ are large, the marginal utility $U(M_{S\cup\{i\}}) - U(M_S)$ is usually small.
- *(ii)* For certain dataset combination S, if its size $|S|$ is close to $(n-1)/2$, its impact on the final result tends to be limited as well, because its coefficient $1/\binom{n-1}{|S|}$ in *MC-SV* is much smaller compared to others.

The above observations *(i)* and *(ii)* indicate that the impact on the estimated *MC-SV* diminishes when the size of the dataset subset $|S|$ approaches or exceeds $(n-1)/2$. This leads us to conjecture that *the contributions of datasets in FL are predominantly influenced by a select few combinations of datasets $S \subseteq N$, particularly those involving smaller FL clients*. To test and validate these observations, we further develop a simple algorithm called K-Greedy (in Alg. 2), which adopts the *MC-SV*-based scheme. Alg. 2 only focuses on combinations with no more than K datasets, intentionally disregarding impacts of combinations with more FL clients.

Empirical Setups. To validate our conjecture above, we embark on an empirical investigation of the K-Greedy algorithm. Our experiments take the FL benchmark dataset FEMNIST [23] and employ the widely-used convolutional neural network as the FL model. Without loss of generality, we partition the FEMNIST dataset into 10 distinct data providers (*i.e.*, FL clients), each holds digits contributed by different writers.

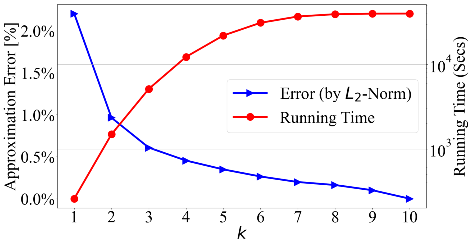


Fig. 4: Results under combinations with size no more than K.

Key Combinations Phenomenon. The empirical results are shown in Fig. 4. To quantify the approximation error, we adopt the relative error metric, defined as $\frac{\|\phi-\hat{\phi}\|_2}{\|\phi\|_2}$, where ϕ represents the data value calculated by *MC-SV*, and $\hat{\phi}$ denotes the approximated data value in FL. The empirical results show that for dataset combinations of size $K \leq 2$, the relative error is less than 1%, which suggests that using dataset combinations involving no more than 2 FL clients allows for a highly accurate approximation of the *MC-SV*. Besides, the relative error decreases rapidly as K increases from 1 to 3 and the rate of decrease becomes more gradual as K becomes larger, which implies that dataset combinations with a larger number of clients have less impact on the final *MC-SV* in FL, which aligns with our conjecture. To summarize, we characterize this observed phenomenon as the *key combinations*, *where a limited number of dataset combinations, typically involving a few clients, play a pivotal role when we take the MC-SV-based computation scheme for data valuation in FL.*

Algorithm 2: K-Greedy

```
Input: The $n$ datasets $\mathcal{D}_N$, a test dataset $\mathcal{T}$, a utility
       function $U(\cdot)$ in FL, and a constant number $K$.
Output: The estimated data value $\hat{\phi}_1, \ldots, \hat{\phi}_n$
// Evaluate the utility of combination of datasets.
for $S \subseteq N$ and $|S| \leq K$ do
    Train the model $M$ and evaluate its utility $U(M_S)$;
end
// Approximate the MC-SV for each data providers.
for $i \in N$ do
    $\hat{\phi}_i \leftarrow \sum_{S\subseteq(N\setminus\{i\}),|S|<K} \frac{U(M_{S\cup\{i\}})-U(M_S)}{n\cdot\binom{n}{|S|}}$ ;
end
return $\hat{\phi}_1, \cdots, \hat{\phi}_n$
```

Algorithm 3: Importance-Pruned Stratified Sampling

```
Input: The $n$ datasets $\{D_1, \cdots, D_n\}$, a test dataset $\mathcal{T}$,
       a utility function $U(\cdot)$ and sampling rounds $\gamma$
Output: Data value for $n$ datasets $\hat{\phi}_1, \cdots, \hat{\phi}_n$
$k^* \leftarrow \max\{k \in \mathbb{N} | \sum_{j=0}^{k} \binom{n}{j} \leq \gamma\}$;
for $k \leq k^*$ do
    Initialize $\mathcal{S}_k$ with all combinations with $k$ datasets;
    for $S \in \mathcal{S}_k$ do
        Train and evaluate the FL model $M_S$ on
         dataset combination $\mathcal{D}_S$ and test dataset $\mathcal{T}$;
    end
end
Sampling a set of dataset combinations $\mathcal{P}$ such that:
  (1) $|\mathcal{P}| \leq \gamma - \sum_{j=0}^{k^*} \binom{n}{j}$ ;
  (2) $\forall S \in \mathcal{P}, |S| = k^* + 1$;
  (3) $\forall i, j \in N, C_i = C_j$ where $C_k = \sum_{S\in\mathcal{P}} \mathbb{I}[k \in S]$ ;
for $S \in \mathcal{P}$ do
    Train and evaluate the FL model $M_S$ on dataset
     combination $\mathcal{D}_S$ and test dataset $\mathcal{T}$;
end
for $i \leftarrow 1$ to $n$ do
    $\hat{\phi}_i \leftarrow \frac{1}{n} \sum_{S\subseteq(N\setminus\{i\}),|S|<k^*} \frac{U(M_{S\cup\{i\}})-U(M_S)}{\binom{n-1}{|S|}}$
      $+\frac{1}{n} \sum_{S\subseteq(N\setminus\{i\}),(S\cup\{i\})\in\mathcal{P}} \frac{U(M_{S\cup\{i\}})-U(M_S)}{\binom{n-1}{|S|}}$;
end
return $\hat{\phi}_1, \cdots, \hat{\phi}_n$
```

B. Importance-Pruning for Acceleration

IPSS Algorithm. Building upon the above empirical evidence, we further refine the proposed stratified sampling framework as described in Sec. III. We introduce a novel approximation algorithm, importance-pruned stratified sampling (*IPSS*), tailored for *MC-SV*-based data valuation in FL. Given the total sampling rounds γ, the *IPSS* prunes dataset combinations involving a large number of FL clients, focusing only on those combinations that have a high-impact on final results. The *IPSS* is detailed in Alg. 3. Given n datasets of FL clients, the utility function $U(\cdot)$ and the sampling rounds γ, the algorithm is design to efficiently approximate contributions of these datasets in FL. The *IPSS* algorithm operates in two phases. *(i) Initially, the algorithm evaluates the utility of FL model trained on various dataset combinations.* In lines 1-7, the *IPSS* calculates the maximum size of used dataset combinations k^*, and then we train and evaluate the FL model on dataset combinations whose sizes do not exceed k^*. In lines 8-11, for remaining sampling rounds, the *IPSS* samples dataset combinations of size $k^* + 1$ and ensures equal sampling frequency for each dataset, thereby providing a fair approximation error across FL clients. In lines 12-14, we train and evaluate the utility of FL models under these dataset combinations containing k^*+1 datasets. *(ii) The Alg. 3 approximates data value based on MC-SV.* In lines 15-17, the algorithm takes the evaluated dataset combinations as proxies for all combinations and estimates the data value by the *MC-SV*-based computation scheme, reducing the computational overhead. Finally, the *IPSS* returns data value for each dataset.

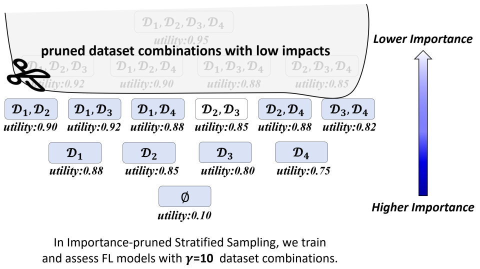

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$
utility:0.95
pruned dataset combinations with low impacts
$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$
utility:0.92
$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$
utility:0.90
$\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$
utility:0.85
$\mathcal{D}_1, \mathcal{D}_2$
utility:0.90
$\mathcal{D}_1, \mathcal{D}_3$
utility:0.92
$\mathcal{D}_1, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_2, \mathcal{D}_3$
utility:0.85
$\mathcal{D}_2, \mathcal{D}_4$
utility:0.88
$\mathcal{D}_3, \mathcal{D}_4$
utility:0.82
$\mathcal{D}_1$
utility:0.88
$\mathcal{D}_2$
utility:0.85
$\mathcal{D}_3$
utility:0.80
$\mathcal{D}_4$
utility:0.75
$\emptyset$
utility:0.10
Lower Importance
Higher Importance
In Importance-pruned Stratified Sampling, we train and assess FL models with γ=10 dataset combinations.

Fig. 5: Example of Alg. 3 with the same setup as Fig. 2

Example 3. *As in Fig. 5, we further illustrate the Alg. 3 back to the settings in Example 2 with four FL clients and sampling rounds $\gamma = 10$. We also take client 1 as the representative. Initially, we computed the maximum size for combinations which are fully evaluated in IPSS algorithm,* $k^* = \max\{k \in \mathbb{N} | \sum_{j=0}^{k} \binom{4}{j} \leq 10\} = 1$. *Then, we train and evaluate FL models with combinations involving no more than one client,* i.e., $M_\emptyset, M_{\{\mathcal{D}_1\}}, M_{\{\mathcal{D}_2\}}, M_{\{\mathcal{D}_3\}}, M_{\{\mathcal{D}_4\}}$. *We can still sample up to* $\gamma - \sum_{j=0}^{k^*} \binom{4}{j} = 5$ *dataset combinations. Satisfying constrains (1)-(3) in Alg. 3, we further sample and evaluate FL models under* $\{\mathcal{D}_1, \mathcal{D}_2\}$, $\{\mathcal{D}_1, \mathcal{D}_3\}$, $\{\mathcal{D}_1, \mathcal{D}_4\}$, $\{\mathcal{D}_2, \mathcal{D}_4\}$ *and* $\{\mathcal{D}_3, \mathcal{D}_4\}$. *We take the MC-SV to compute average marginal utilities of FL client 1 using all assessed combinations. Finally, we have $\hat{\phi}_1 = 0.22$. Similarly, the estimated data value of FL client 2, 3 and 4 are 0.20, 0.1842 and 0.1667, respectively.*

C. Theoretical Analysis of the IPSS Algorithm

Theoretical Evidence. For simplicity, we continue to use the FL linear regression model and theoretically analyze the approximation error and time complexity of the *IPSS* algorithm.

Lemma 1. *Given n datasets each with t training samples, if we take negative mean square error (MSE) as the utility function, the estimated data value of client i is $\mathbb{E}[\hat{\phi}_i] = \frac{1}{n}(m_0 - \frac{\mu_e|x|}{nt-|x|-1})$, where $|x|$ is input feature dimensions, μ_e is expectation of random noise and m_0 is MSE of the initialized model.*

Proof. In the proof we follow the analysis framework for FL by *Donahue* and *Kleinberg* [24] where all $|\mathcal{D}|$ data items are drawn from the standard *Gaussian distribution* $\mathcal{N}(0, I)$ and the expected MSE of the linear regression model is as,

$$\mathbb{E}[mse(|\mathcal{D}|)] = \mu_e|x|/(|\mathcal{D}| - |x| - 1), \tag{12}$$

where μ_e is the expectation of random noise over data, $|x|$ is the dimension of input features, and $|\mathcal{D}|$ is the size of used data. Similarly, for a FL linear regression model with $|\mathcal{D}_S| = t|S|$ data items, its expected MSE can be writen as,

$$\mathbb{E}[U(M_S)] = \mathbb{E}[mse(|\mathcal{D}_S|)] = \mu_e|x|/(t|S| - |x| - 1), \tag{13}$$

If we take the negative MSE as the utility function $U(\cdot)$ in data valuation for FL, we can calculate the expectation of data value using *MC-SV* and above analysis model [24] as below,

$$\begin{aligned}\mathbb{E}[\hat{\phi}_i] &= \mathbb{E}[\frac{1}{n} \sum_{S \subseteq (N \setminus \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}] \\ &= \mathbb{E}[\frac{1}{n} \sum_{S \subseteq (N \setminus \{i\})} \frac{-mse((|S|+1)t) + mse(|S|t)}{\binom{n-1}{|S|}}] \\ &= \frac{1}{n} \sum_{k=0}^{n-1} (-\mathbb{E}[mse((k+1)t)] + \mathbb{E}[mse(kt)])\end{aligned} \tag{14}$$

As $mse(0)$ is not defined in [24], we let m_0 denotes the MSE of the initialized linear model. So the $\mathbb{E}[\hat{\phi}_i]$ is ,

$$\mathbb{E}[\hat{\phi}_i] = \frac{1}{n}(m_0 - \mathbb{E}[mse(nt)]) = \frac{1}{n}(m_0 - \frac{\mu_e \cdot |x|}{nt - |x| - 1}) \tag{15}$$

Finally, we have the $\mathbb{E}[\hat{\phi}_i]$ and then completed our proof. □

Theorem 3. *Given the sampling rounds γ and taking same assumption as Lemma 1, the approximation error bound of Alg. 3 is $\mathcal{O}(\frac{n-k^*}{k^* nt})$, where $k^* = \arg\max_k\{\sum_{i=0}^{k} \binom{n}{j} \leq \gamma\}$.*

Proof. Alg. 3 takes all dataset combinations with no more than k^* clients, where $k^* = \arg\max_k\{\sum_{i=0}^{k} \binom{n}{j} \leq \gamma\}$. Similar to Lemma 1, we calculate FL client i's expected contribution as,

$$\begin{aligned}\mathbb{E}[\hat{\phi}_i^{k^*}] &= \frac{1}{n} \sum_{k=0}^{k^*-1} (-\mathbb{E}[mse((k+1)t)] + \mathbb{E}[mse(kt)]) \\ &= \frac{1}{n}(m_0 - \frac{\mu_e|x|}{k^*t - |x| - 1})\end{aligned} \tag{16}$$

Together with Lemma 1, the ratio of $\mathbb{E}[\hat{\phi}_i^{k^*}]$ and $\mathbb{E}[\phi_i]$ is,

$$\frac{\mathbb{E}[\hat{\phi}_i^{k^*}]}{\mathbb{E}[\phi_i]} = \frac{\frac{1}{n}(m_0 - \frac{\mu_e \cdot |x|}{k^*t-|x|-1})}{\frac{1}{n}(m_0 - \frac{\mu_e \cdot |x|}{nt-|x|-1})} = 1 - \frac{\frac{\mu_e \cdot |x|}{k^*t-|x|-1} - \frac{\mu_e \cdot |x|}{nt-|x|-1}}{m_0 - \frac{\mu_e \cdot |x|}{nt-|x|-1}} \tag{17}$$

As a model trained by $|x|+2$ training samples can outperform the initialized model, so the $mse(|x|+2)$ is less than the MSE of the initialized model m_0. We have following inequations,

$$\frac{\mathbb{E}[\hat{\phi}_i^{k^*}]}{\mathbb{E}[\phi_i]} \geq 1 - \frac{\frac{1}{k^*t-|x|-1} - \frac{1}{nt-|x|-1}}{\frac{1}{|x|+2-|x|-1} - \frac{1}{nt-|x|-1}} = 1 - \frac{(n-k^*)t}{(k^*t-|x|-1)(nt-|x|-2)} \tag{18}$$

Note that the input feature dimension $|x|$ is an constant number and we can complete the proof of Theorem 3 as follows,

$$\frac{|\mathbb{E}[\hat{\phi}_i^{k^*}] - \mathbb{E}[\phi_i]|}{\mathbb{E}[\phi_i]} \leq \frac{(n-k^*)t}{(k^*t-|x|-1)(nt-|x|-2)} = \mathcal{O}\left(\frac{n-k^*}{k^*nt}\right) \tag{19}$$

□

Approximation Error Analysis. Theorem 3 suggests that even with a small k^*, the relative error between $\mathbb{E}[\phi_i^{k^*}]$ and $\mathbb{E}[\phi_i]$ can remain minimal, which is particularly relevant in typical FL scenarios, where the number of training samples in a dataset substantially exceeds the dimension of input features. Take MNIST [25], a most widely used benchmark dataset, as an example. It contains over 60,000 training images, each represented by 784 dimensional features, implying that $|x| \ll nt$. If each client holds the same number of training samples, the relative error of Alg. 3 is $\mathcal{O}(\frac{n}{k^*\mathcal{D}})$. In this case, accurately approximating the data value only need to evaluate a select group of dataset combinations, consistent with the *key combinations phenomenon* observed earlier in Sec. IV-A.

Time Complexity Analysis. The time cost of Alg. 3 mainly relies on FL training and assessing processes on various dataset combinations. Assuming time cost to train and evaluate the FL model is denoted as τ. The time complexity of Alg. 3 can be analyzed as follows. For lines 1-14, the complexity is $\mathcal{O}(\tau\gamma)$, as Alg. 3 utilizes no more than γ dataset combinations. For lines 15-17, Alg. 3 computes marginal contributions upto $\sum_{j=0}^{k^*-1}\binom{n}{j+1}\binom{n}{j}+\binom{n}{k^*}(\gamma-\sum_{j=0}^{k^*}\binom{n}{j})$ times, leading to a time complexity of $\mathcal{O}(\gamma\binom{n}{k^*})$. We usually only take a small group of combinations in data valuation so k^* is a small integer. As time to train and evaluate a FL model τ is usually much larger than $\binom{n}{k^*}$, the time complexity of the *IPSS* is $\mathcal{O}(\tau\gamma)$.

V. Experimental Evaluations

This section presents evaluations of our proposed methods.

A. Experimental Setup

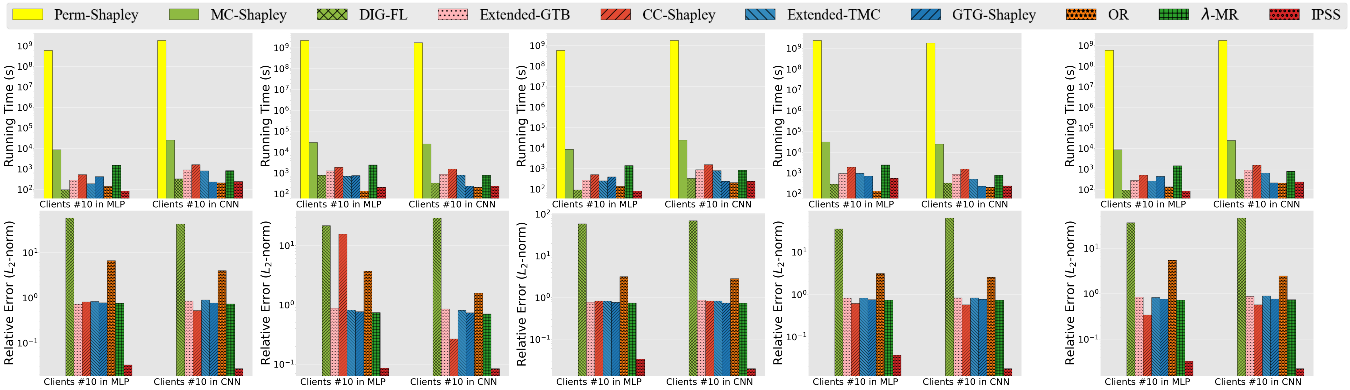
Datasets. We evaluate baseline algorithms for SV-based data valuation in FL over both synthetic and real-world datasets. *(i) Synthetic Dataset*. We take the MNIST [25], a widely-used datasets with 60,000+ training samples and 10,000+ testing samples to create the synthetic datasets in FL. Following the experimental setup in [6], [12], we split the MNIST [25] into partitions and create customized training dataset tailored for FL, where datasets of each FL client varies in size, distribution and quality (*i.e.*, noise). We highlight experimental features in each FL setting. *(a) same-size-same-distribution*: we split training dataset into partitions with same size and label distribution. *(b) same-size-different-distribution*: we partition the training samples and set some label are majorly belongs to certain client. *(c) different-size-same-distribution*: we randomly split training samples into partitions with their ratios of data size $1:2:\cdots:n$, where n is the client number. *(d) same-size-noisy-label*: we change $0\% \sim 20\%$ of labels in the partitioned dataset into one of other labels with equal probability. *(e) same-size-noisy-feature*: we generate *Gaussian* noise $\mathcal{N}(0,1)$ and scale them by multiplying $0.00 \sim 0.20$ as the noise added on training samples.
(ii) Real-world Dataset. We also conduct experiments on three real-world datasets, FEMNIST [23], Adult [26] and Sent-140 [23]. FEMNIST [23] is a benchmark dataset for FL and is included within TensorFlow Federated [27]. It contains 805,000+ samples from 3,500+ users, allowing it to be partitioned into datasets for clients in FL by the user-ids. The Adult [26] is a tabular dataset commonly used in vertical FL [7], [28], [29] and it contains 48,800+ training samples and 14 features (*e.g.*, income, occupation, and native-country). Without loss of generality, we can partition the training samples in Adult to several datasets for FL clients according to user's occupation.

Compared Algorithms. We compare our *IPSS* algorithm (in Sec. IV) with a series of existing baselines (in three categories). The first category (*Perm-Shapley*, *MC-Shapley* and *DIG-FL*) calculates data value directly by definition, while the second category (*Exteneded-TMC*, *Extended GTB* and *CC-Shapley*) uses sampling-based methods. The last category (*OR*, *λ-MR*, *GTG-Shapley*) approximates the data value through gradients collected in FL training process.

- *Perm-Shapley*. It directly calculates data value of clients in FL according to the definition of the *Permutation-based Shapley value* (*Perm-SV*), which trains and evaluates FL models based on permutations of all datasets.
- *MC-Shapley*. Similarly, it directly calculates the data value through the *MC-SV* based computation scheme.
- *DIG-FL*. It efficiently approximates the data value in FL [7], which only needs to evaluate $\mathcal{O}(n)$ numbers of dataset combinations under certain assumptions [7].
- *Extended-TMC*. It is an extension of widely-adopted data valuation scheme for general machine learning [17]. We extend and compare the Truncated Monte Carlo algorithm of [17] to FL scenario. It randomly generates a permutation π of all $n!$ permutations and trains and evaluates the FL models based on the permutation. Then, the algorithm approximates the *Perm-SV* according to,

$$\phi_i = \mathbb{E}_{\pi\sim\Pi}[U(M_{\pi[\mathrm{p}(i)]\cup\{i\}}) - U(M_{\pi[\mathrm{p}(i)]})]. \tag{20}$$

- *Extended-GTB*. It is also an extension of a representative data valuation scheme [18] and we extend the Group-Testing-Based SV estimation to FL scenario as follows. The *GTB* can estimate the contributions of each client in FL by solving a feasibility problem through the randomly selected subsets $S \subseteq N$. Finally, we incrementally relax the constraints until there is a feasible solution.
- *OR*. It directly takes gradients within the FL process with all clients the same as gradients under other combinations [6]. OR can approximate the FL model by these gradients



(a) same-size-same-distr. *(b) same-size-diff.-distr.* *(c) diff.-size-same-distr.* *(d) same-size-noisy-label* *(e) same-size-noisy-feature*

Fig. 6: Experimental results on the synthetic datasets with five different setups varying in size, distribution and quality.

without extra training, however, there is no theoretical guarantee for OR in approximation errors.

- *λ-MR*. It takes the *MC-SV*-based scheme and estimates data value in each training round of FL and aggregate them as the final results [9]. The λ-MR avoids the additional training of the FL models as well.
- *CC-Shapley*. It is one of the state-of-the-art sampling methods to approximate the Shapley value [19], which estimates data value using the *CC-SV*-based schemes.
- *GTG-Shapley*. Similar to λ-MR, it also approximates the data value using gradients [12]. It adopts the *Perm-SV* and uses Monte Carlo sampling approach to reduce the number of model reconstructions over rounds.

Evaluation Metrics. We employ following two metrics to assess the performance of compared algorithms. *(i) Calculation Time*: it measures the running time required to calculate the data value, including the time to train and evaluate FL models. *(ii) Approximation Error*: it represents the effectiveness of approximation algorithms by the relative error in l_2-norm:

$$l_2(\hat{\phi},\phi) = \|\hat{\phi}-\phi\|_2/\|\phi\|_2 = \sqrt{\sum_{i=1}^{n}(\phi_i-\hat{\phi}_i)^2}\Big/\sqrt{\sum_{i=1}^{n}{\phi_i}^2} \qquad (21)$$

where $\phi = (\phi_1, \phi_2, \ldots, \phi_n)$ denotes the data value of n FL clients and $\hat{\phi} = (\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_n)$ is the approximation results.

Implementations. All algorithms were implemented in Python with TensorFlow 2.4 [30] and TensorFlow Federated 0.18 [27]. To simulate multiple data providers in FL, we adopt the multi-processing techniques and the gRPC protocol. The experimental setup was executed on a machine equipped with an NVIDIA GeForce RTX 3090 GPU, an AMD Ryzen 7950X CPU @ 3.0GHz, and 128GB of main memory. Our experiments incorporated multi-layer perceptron (MLP), convolutional neural network (CNN) and XGBoost (XGB) models, which are all extensively used in data science community. When the number of FL clients is three, six, and ten, all sampling-based approximation approaches are configured with the same number of sampling rounds, *i.e.*, 5, 8, and 32, respectively (as in Table III). The open-sourced code is available at “https://github.com/t0ush1/Shapley-Data-Valuation”.

$n = 3 \rightarrow \gamma = 5$	$n = 6 \rightarrow \gamma = 8$	$n = 10 \rightarrow \gamma = 32$

TABLE III: The adopted sampling rounds (γ) for client number (n).

B. Performance on Synthetic Datasets

We showcase the experimental results under varying dataset sizes, distributions, and noise levels. This series of experiments presents the time cost and approximation error for the compared algorithms using both MLP and CNN models. In each of the following five training setups, we use ten clients in FL. *(a) same-size-same-distr.*. Fig. 6(a) plots time cost and error of compared algorithms. For the time cost, *OR* and *IPSS* have the lowest time cost in both MLP model and CNN model. The time cost of *MC-Shapley* is 62.2× and 104.1× as *OR* and *IPSS*, respectively. For approxi error, *IPSS* is the lowest and close to zero, *i.e.*, the estimated data value via *IPSS* is almost the same as the exact one. *(b) same-size-diff.-distr.*. From Fig. 6(b), *OR* is still the fastest in both MLP and CNN model and *IPSS* is the second fastest. *IPSS* is 3.4∼9.0× faster than *GTG-Shapley* and *CC-Shapley*, respectively. For approximation error, *IPSS* outperforms other algorithms. *OR* performs poor in accuracy. Overall, *IPSS* outperforms the others in this setting. *(c) diff.-size-same-distr.*. As shown in Fig. 6(c), *OR* and *IPSS* exhibit a lower time cost compared to other baseline algorithms. For estimation error, *IPSS* also approximates the exact SV well and outperforms the other approximation algorithms significantly. The *λ-MR* ranks the second in accuracy for both MLP and CNN model.*(d) same-size-noisy-label*. From Fig. 6(d), the relative error of *λ-MR* and *IPSS* is stable and *IPSS* still has the lowest error. The relative error of *Extended-TMC* and *Extended-GTB* is 22.3× and 22.5× of *IPSS*, respectively. *(e) same-size-noisy-feature*. *λ-MR* and *CC-Shapley* have the highest time cost for MLP and CNN models among all compared algorithms. As shown in Fig. 6(e), the error of *CC-Shapley* and *λ-MR* is 10.3∼26.0× and 22.1∼33.6× greater than that of *IPSS*, respectively.

C. Results on Real-world Dataset

We also validate our approximation algorithm on two real dataset, which can be naturally partitioned several datasets for clients in FL and we detail evaluations on each below.

	n	Metrics	*Perm-Shap.*	*MC-Shap.*	*DIG-FL*	*Ext-TMC*	*Ext-GTB*	*CC-Shap.*	*GTG-Shap.*	*OR*	*λ-MR*	*IPSS*
MLP	3	Time(s)	3729	842	584	568	807	1021	47	**12**	29	258
		Error(l_2)	-	-	5.01	0.79	0.59	0.35	0.90	2.46	0.88	**0.06**
	6	Time(s)	9.1×10^6	6496	1077	843	1120	2020	161	**89**	228	329
		Error(l_2)	-	-	0.70	0.96	0.90	1.93	0.89	3.13	0.87	**0.49**
	10	Time(s)	6.8×10^9	95985	1695	3061	4129	5988	1086	1414	3764	**568**
		Error(l_2)	-	-	0.77	0.82	0.85	1.16	0.85	3.09	0.83	**0.02**
CNN	3	Time(s)	1629	372	230	231	352	413	26	**7**	22	142
		Error(l_2)	-	-	95.14	0.81	0.60	0.02	0.87	0.46	0.73	**0.01**
	6	Time(s)	3.6×10^5	2783	407	352	484	667	108	**47**	154	211
		Error(l_2)	-	-	78.25	0.91	0.70	0.40	0.76	0.35	0.73	**0.02**
	10	Time(s)	2.8×10^9	40134	655	1220	1612	2553	680	641	2504	**257**
		Error(l_2)	-	-	98.42	0.83	0.87	2.60	0.75	0.76	0.71	**0.02**

TABLE IV: We mark the "best performance" as **green**. "-" denotes the solution can exactly computes the *SV*-based data values.

Results on FEMNIST. Table IV shows the experimental results on FEMNIST [23] datasets across various numbers of FL clients and we take both MLP and CNN as the FL model. *In MLP model*. Taking MLP as the FL model, we have the following observations: *(i)* In scenarios with ten FL clients, our *IPSS* algorithm achieves the lowest time cost, reducing computing overhead by 99% compared to *MC-Shapley* and performing 2.98× and 1.91× faster than *DIG-FL* and *GTG-Shapley*, respectively. *(ii)* In terms of the relative error, *IPSS* significantly outperforms other algorithms across all numbers of clients. Notably, the error of *IPSS* is 38.5× and 42.5× lower than *Extended-TMC* and *GTG-Shapley* with 10 clients. *In CNN model*. The results in CNN model exhibit similarities to that observed in MLP model. *(i)* For the efficiency, *OR* is superior when the number of clients is 3 and 6, while *IPSS* is the fastest when there are larger number of clients. *(ii)* Regarding approximation error, *IPSS* consistently shows the lowest error over various clients, which is one order of magnitude smaller than other approximation algorithms. *(iii)* However, the relative error of *DIG-FL* is notably higher in the CNN model. *In summary, IPSS excels in efficiency with more FL clients and consistently exhibits lower error compared to baselines across various numbers of FL clients.*

	n	Metrics	*Perm-Shap.*	*MC-Shap.*	*DIG-FL*	*Ext-TMC*	*Ext-GTB*	*CC-Shap.*	*GTG-Shap.*	*OR*	*λ-MR*	*IPSS*
MLP	3	Time(s)	720	164	94	95	138	199	59	**13**	48	69
		Error(l_2)	-	-	1.02	1.46	1.89	0.09	5.30	1.00	2.93	**0.05**
	6	Time (s)	3.3×10^5	2820	252	220	306	530	271	**74**	347	146
		Error(l_2)	-	-	1.12	2.30	2.02	0.18	3.65	1.00	3.21	**0.13**
	10	Time(s)	2.1×10^9	28983	454	732	1152	1850	1428	1127	5575	**206**
		Error(l_2)	-	-	1.23	2.19	1.97	0.09	3.95	0.99	3.83	**0.08**
XGB	3	Time(s)	29.2	6.5	4.7	4.6	8.5	8.2	\	\	\	**1.8**
		Error(l_2)	-	-	0.95	1.38	0.45	0.27				**0.04**
	6	Time(s)	13308	96	19	14	22	25	\	\	\	**3**
		Error(l_2)	-	-	0.98	2.16	1.77	0.13				**0.07**
	10	Time(s)	1.7×10^8	2256	50	81	111	151	\	\	\	**5**
		Error(l_2)	-	-	0.98	1.41	1.59	0.13				**0.12**

TABLE V: We mark the "best performance" as **green**. "\" denotes gradient-based approximation is not applicable to the XGB model.

Results on Adult. We also take a tabular dataset and adopt the MLP and XGB model as the FL model to compare the effectiveness and efficiency of the baselines. Table V presents the experimental results over different client numbers.

In MLP model. The experimental results on Adult are similar to those on FEMNIST, when using MLP as the FL model. *(i)* For time cost, *IPSS* is still the most efficient when there are 10 clients and it is 2.2× faster than *DIG*, the second most efficient algorithm. *(ii)* For approximation error, *IPSS* exhibits the lowest error over all client numbers, achieving an average improvement of 43× over *GTG-Shap* and 34× over *λ-MR*. *In XGB model.* As gradient-based algorithms (*GTG-Shapley*, *OR* and *λ-MR*) are not applicable to XGBoost, we evaluate the definition-based and sampling-based approaches for calculating the SV in this setup. The experimental observations are as follows. *(i)* When varying numbers of clients, *IPSS* consistently shows its superior in efficiency. When there are 10 clients in FL, it is 10∼30× faster than other compared algorithms. *(ii)* Similarly, *IPSS* achieves the lowest approximation error, reducing the error by 25.7× and 16.6× compared to *Extended-TMC* and *Extended-GTB*, respectively. *In this setup, the proposed IPSS performs the best in efficiency as the client number increases and achieves the highest accuracy as well.*

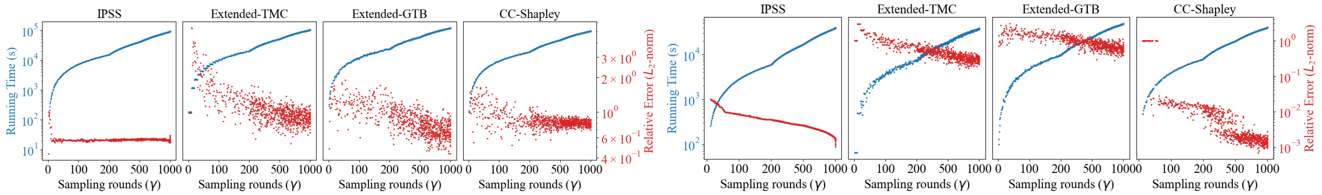
D. In-depth Analysis of Compared Algorithms

Next, we conduct more interpretation experiments on the FL benchmark dataset FEMNIST [23] to validate the efficiency and effectiveness of the compared approximation algorithms.

1) Impacts of varying the sampling rounds: As the total sampling round is crucial for sampling-based solutions (*i.e.*, *IPSS*, *Extended-TMC*, *Extended-GTB* and *CC-Shapley*), we study the impacts of varying total sampling rounds γ with ten FL clients on FEMNIST [23]. From Fig. 7, we have following observations. *(i)* As γ grows, *IPSS* has more stable and lower error compared with other baselines. Specifically, the variance in error of *CC-Shapley* is 7.7× and 50.9× higher than that of *IPSS* on MLP and CNN, respectively. *(ii) IPSS* fast achieves low approximation errors (*i.e.*, below 10^{-2}) with $\gamma < 100$, whereas *CC-Shapley* reaches the same error level stably only when $\gamma > 200$. In summary, *IPSS* achieves lower error more quickly and stably than compared algorithms.

2) Pareto curves for time-error trade-off: We run the sampling-based algorithms 100 times with each γ and plot the *Pareto curves* to show the trade-off between efficiency and effectiveness. The experimental results using FEMNIST [23] across three, six, and ten FL clients are shown in Fig. 8 (a)∼(f). We have the following observations: *(i) IPSS* achieves *Pareto optimality* on FEMNIST across various numbers of FL clients. *(ii)* Though *OR* runs fast in above end-to-end experiments with three to six clients (in Table IV), *IPSS* can achieve comparable performance to *OR* with a suitable sampling round γ.

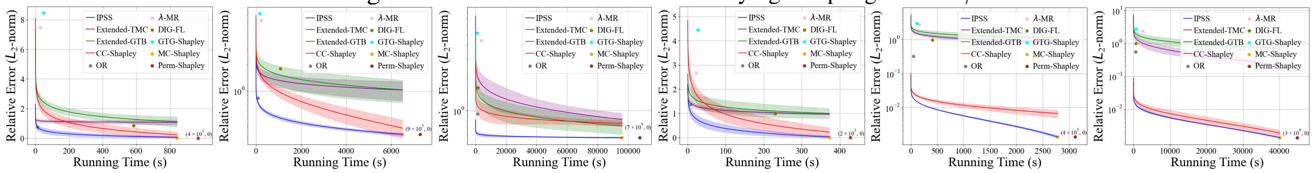
3) Scalability test for larger FL clients: We conduct experiments with up to 100 FL clients, a large-scale scenario for cross-silo FL [1]–[3], where more than 10^{30} dataset combinations must be assessed by SV definition, making it infeasible to compute the ground-truth within limited time. Thus, we set 5% of FL clients with empty datasets and 5% of FL clients having same datasets as others and take the extent to which algorithms satisfy required properties (*i.e.*, *no-free-rider* and *symmetric-fairness*) as proxies for approximation error. We set the sampling round γ for sampling-based algorithms to $n \log n$. As in Fig. 9: *(i)* For running time, *IPSS* outperforms *Extended-TMC*, *Extended-GTB* and *CC-Shapley* with both 20 and 100 FL clients. *(ii)* As client number increases from 20 to 100, the running time of our *IPSS* increases only by 2.4×.



(a) Results of using the MLP as FL model.

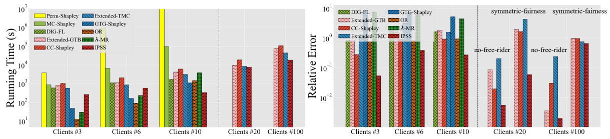
(b) Results of using the CNN as FL model.

Fig. 7: Results on FEMNIST when varying sampling rounds γ.



(a) Client#3+MLP *(b) Client#6+MLP* *(c) Client#10+MLP* *(d) Client#3+CNN* *(e) Client#6+CNN* *(f) Client#10+CNN*

Fig. 8: Pareto curves for trade-off in efficiency and effectiveness.

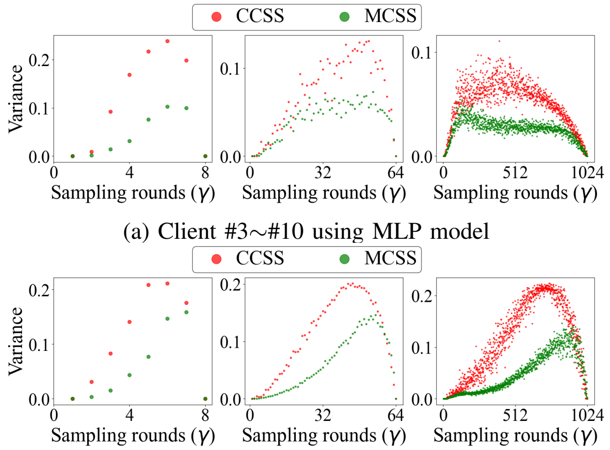


(a) Time cost. *(b) Approximation error.*

Fig. 9: Varying client number on FEMNIST using MLP model.

(iii) For error based on *no-free-riders* and *symmetric fairness*, *IPSS* achieves the lowest error among compared algorithms.

4) Comparing variance of MC-SV and CC-SV: We run Alg. 1 100 times using *MC-SV* and *CC-SV*, respectively, to calculate their variance. The experimental results are shown in Fig. 10. *(i)* As γ increases, the variance of *MC-SV* and *CC-SV* initially rises and then decreases as almost all possible combinations are sampled, introducing nearly exact data values with close to zero variance. *(ii)* Using both MLP and CNN models, *MC-SV* exhibits lower variance than *CC-SV*, with FL clients number from three to ten, consistent with the theoretical analysis of Theorem 2 in Sec. III-B and justifying the selection of *MC-SV* for our stratified sampling based approximation.



(a) Client #3∼#10 using MLP model

(b) Client #3∼#10 using CNN model

Fig. 10: Analysis of variance for *MC-SV* and *CC-SV*.

E. Summary of Experimental Results

Our major experimental findings are summarized as follows:

Efficiency. Among approximation algorithms evaluated, *Extended-GTB* and *CC-Shapley* incur the highest time cost in the most experimental setup. Our *IPSS* algorithm emerges as the most efficient one in most setups, especially within larger number of clients. The time cost of *λ-MR* increases exponentially with number of FL clients, limiting its scalability.

Effectiveness. Leveraging insights in Sec. IV-A, the proposed*IPSS* consistently achieves the lowest estimation errors across nearly all setups. Though *DIG-FL* and *OR* are more efficient than most compared baselines, they also exhibit a higher approximation error in most experimental setups.

VI. Related Work

Our work is mainly related to two lines of research topics: the *federated learning* and the *Shapley value based data valuation*. We review the representative work in the following.

A. Federated Learning

In recent years, data regulations (*e.g.*, GDPR [31] and CCPA [32]) have imposed strict requirements on data privacy, posing challenges for privacy-preserving data analysis in both academia and industry. *Federated learning (FL)*, enabling multiple data providers to collaboratively train models without sharing their raw data, has emerged as a new paradigm to tackle the data privacy issues. Based on the type of clients (*a.k.a.* data providers), FL can be divided into two settings: *cross-device* and *cross-silo*. We introduce each setting below.

1) Cross-Device FL: In this setting, the typical FL clients are a large number mobile or IoT devices [1], where both the computation and the communication is often the bottleneck. Therefore, how to reduce the communication cost is a crucial issue in cross-silo FL. In the seminal work [5], McMahan *et al.* propose the most widely adopted FL algorithm, FedAVG, to solve the well-known non-IID problem and reduce communication costs by aggregating model parameters rather than gradients. Then, a series of subsequent works have proposed FL algorithms, such as FedProx [33], Scaffold [34], etc. In addition to the non-IID issue, how to tackle device heterogeneity has recently received increasing attention in cross-silo FL as well [35]–[38]. For example, authors in [35], [36] propose an open-source platforms for real-world cross-device FL, called FS-REAL, which supports advanced FL features such as communication optimization and asynchronous concurrency. Liu *et al.* [39] propose the InclusiveFL, an FL framework that adjusts the size of models before assigning them to clients with different computing capabilities.

2) Cross-Silo FL: Yang *et al.* [2] enrich the concept of FL and introduce the cross-silo FL, a scenario usually involving a small number of clients, such as institutions or companies with abundant computational and communication resources. The non-IID issue is also the central challenge in cross-silo FL. Representatively, Huang *et al.* [40] adopt the neural networks as the FL model and propose the FedAMP algorithm to solve the non-IID problem and Li *et al.* [41] conduct a comprehensive experimental study to compare the performance of various FL algorithms in cross-silo FL. Besides, tree-based models have been widely studied in cross-silo FL by prior work [7], [28], [29], [42], [43], especially when clients hold the partitioned tabular datasets. In this paper, we focus on the cross-silo FL setting and adopt both the neural networks and tree-based models as the FL model in our evaluations.

B. Shapley Value Based Data Valuation

The Shapley value [44] has been widely adopted in data valuation [7], [11], [12], [15]–[17], [19], [45] and some variants are proposed for various scenarios or requirements [8], [10], [13], [21], [46], [47]. Data valuation can be divided into two categories: *within a dataset* and *across datasets*.

1) Data Valuation within a Dataset: It aims to fairly measure the importance or contribution for each sample (*i.e.*, data point). In 2019, Ghorbani *et al.* [17] first introduce the Shapley value in data valuation and define the *Data Shapley* value to qualify the influence of a sample in a dataset. To reduce the computational overhead, they further propose two approximation algorithms, *Truncated Monte Carlo* and *Gradient Shapley*, where the former can be extended to the FL framework [6]. Since computing SV is usually time-consuming, prior work primarily focus on how to design effective and efficient approches to SV based data valuation. Jia *et al.* [11] propose an exact and efficient algorithm to calculate valuation of samples for *k*NN classification in $\mathcal{O}(n \log n)$ time complexity, where n is the dataset size. They also leverage the sparsity of SV for a singel sample in a dataset to enable efficient approximation [18]. However, the sparsity of SV is inexistent in cross-silo FL, so we only extend their another sampling-based algorithm, Group Testing Based SV, as a baseline in our paper. Zhang *et al.* [19] propose a novel equavilent expression of SV based on complementary contribution, upon which they design a sampling-based approximation algorithm for SV that is also applicable for data valuation. We compare their proposed equavilent SV expression (referred as *CC-SV* in this paper) with another two commonly used SV expression and adopt the approaches in [19] as one of our baselines as well.

2) Data Valuation Across Datasets: It aims to identify the contribution of each dataset (*i.e.*, the contribution estimation), which is consistent with the data valuation in FL [7], [13], [48], where how to design an efficient and effective approximation algorithms is the primary issue. The typical algorithms in this setup is the gradient construction based approximation, which utilizes gradients in FL to build federated models under various dataset combinations and avoids the need to train extra FL models for data valuation. Song *et al.* [6] first propose the gradient construction based approaches to measures the contribution of datasets in FL and they also propose another algorithm, λ-MR, to further reduce time cost by reconstructing FL model based on gradients in each training round [9]. In [12], the authors propose an efficient algorithm called *GTG-Shapley* to approximate the SV by combining the on the gradient construction with Monte Carlo sampling. We compare our proposed algorithm against OR, λ-MR and GTG-Shapley in experimental evaluations. Besides, Wang *et al.* [7] propose an efficient data valuation approaches to measure the contributions of clients in FL, which only needs linear number of evaluations under certain assumption and we take their approach as the baseline as well. Zheng *et al.* [45] study the secure data valuation for cross-silo FL and exploring ways to enhance the efficiency using an efficient two-server protocol. As security is outside the scope of this work, we do not take their approach as one of the baselines in our experiments.

VII. Conclusion

In this paper, we investigate the Shapley value based data valuation in FL and introduce an efficient and effective sampling-based approximation algorithm, *IPSS*. Specifically, we first propose a unified stratified sampling-based approximation framework that seamlessly integrates both *MC-SV*-based and *CC-SV*-based computation schemes. We also identify a crucial phenomenon called key combinations, where only limited dataset combinations highly impact final data value results in FL. Building upon our new findings, we propose a practical approximation algorithm, *IPSS*, which strategically selects high-impact dataset combinations rather than taking all possible dataset combinations in FL, thus significantly improving the efficiency with high approximation accuracy. Finally, we conduct extensive evaluations on real and synthetic datasets to validate that the proposed *IPSS* outperforms the representative baselines in both efficiency and effectiveness.

VIII. Acknowledgements

We thank the anonymous reviewers for their suggestions. This work was partially supported by National Science Foundation of China (NSFC) (Grant Nos. 62425202, U21A20516, 62336003), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities No.JK2024-03, the Didi Collaborative Research Program, and the State Key Laboratory of Complex & Critical Software Environment Lab. Zimu Zhou's research is supported by Chow Sang Sang Group Research Fund No. 9229139. Yongxin Tong is the corresponding author.

References

[1] P. Kairouz, H. B. McMahan, B. Avent *et al.*, “Advances and open problems in federated learning,” *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, p. 12, 2019.

[3] M. Ye, X. Fang, B. Du *et al.*, “Heterogeneous federated learning: State-of-the-art and research challenges,” *ACM Comput. Surv.*, vol. 56, no. 3, pp. 79:1–79:44, 2024.

[4] Q. Li, Z. Wen, Z. Wu, S. Hu *et al.*, “A survey on federated learning systems: Vision, hype and reality for data privacy and protection,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, 2023.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017, pp. 1273–1282.

[6] T. Song, Y. Tong, and S. Wei, “Profit allocation for federated learning,” in *BigData*. IEEE, 2019, pp. 2577–2586.

[7] J. Wang, L. Zhang, A. Li, X. You *et al.*, “Efficient participant contribution evaluation for horizontal and vertical federated learning,” in *ICDE*. IEEE, 2022, pp. 911–923.

[8] H. Xia, J. Liu, J. Lou, Z. Qin *et al.*, “Equitable data valuation meets the right to be forgotten in model markets,” *VLDB*, vol. 16, no. 11, pp. 3349–3362, 2023.

[9] S. Wei, Y. Tong, Z. Zhou, and T. Song, “Efficient and fair data valuation for horizontal federated learning,” *Federated Learning: Privacy and Incentive*, pp. 139–152, 2020.

[10] T. Wang, J. Rausch, C. Zhang, R. Jia *et al.*, “A principled approach to data valuation for federated learning,” in *Federated Learning Privacy and Incentive*. Springer, 2020, vol. 12500, pp. 153–167.

[11] R. Jia, D. Dao, B. Wang, F. Hubis *et al.*, “Efficient task-specific data valuation for nearest neighbor algorithms,” *VLDB*, vol. 12, no. 11, pp. 1610–1623, 2019.

[12] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, “Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning,” *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 60:1–60:21, 2022.

[13] Y. Chen, K. Li, G. Li, and Y. Wang., “Contributions estimation in federated learning: A comprehensive experimental evaluation,” *VLDB*, vol. 17, no. 8, pp. 2077–2090, 2024.

[14] R. Myerson, *Game theory*. Harvard University Press, 2013.

[15] L. E. Bertossi, B. Kimelfeld, E. Livshits *et al.*, “The shapley value in database management,” *SIGMOD Rec.*, vol. 52, no. 2, pp. 6–17, 2023.

[16] B. Rozemberczki, L. Watson, P. Bayer *et al.*, “The shapley value in machine learning,” in *IJCAI*. ijcai.org, 2022, pp. 5572–5579.

[17] A. Ghorbani and J. Zou, “Data shapley: Equitable valuation of data for machine learning,” in *ICML*, vol. 97, 2019, pp. 2242–2251.

[18] R. Jia, D. Dao, B. Wang, F. Hubis *et al.*, “Towards efficient data valuation based on the shapley value,” in *AISTATS*, vol. 89, 2019, pp. 1167–1176.

[19] J. Zhang, Q. Sun, J. Liu, L. Xiong *et al.*, “Efficient sampling approaches to shapley value approximation,” *SIGMOD*, vol. 1, no. 1, pp. 48:1–48:24, 2023.

[20] X. Deng and C. Papadimitriou, “On the complexity of cooperative solution concepts,” *Math. Oper. Res.*, vol. 19, no. 2, pp. 257–266, 1994.

[21] J. Wang and R. Jia, “Data banzhaf: A robust data valuation framework for machine learning,” in *AISTATS*, vol. 206, pp. 6388–6421.

[22] C. Rohilla, “The truth about linear regression,” *Online Manuscript*, 2015.

[23] S. Caldas, S. M. D. Karthik, P. Wu *et al.*, “Leaf: A benchmark for federated settings,” *arXiv*, vol. abs/1812.01097, 2018.

[24] K. Donahue and J. M. Kleinberg, “Model-sharing games: Analyzing federated learning under voluntary participation,” in *AAAI*. AAAI Press, 2021, pp. 5303–5311.

[25] Y. LeCun, C. Cortes, and C. J. Burges, “The MNIST Database,” in *http://yann.lecun.com/exdb/mnist/*.

[26] B. Barry and R. Kohavi, “Adult,” UCI Machine Learning Repository.

[27] “Tensorflow federated,” in *www.tensorflow.org/federated/federated_learning*.

[28] F. Fu, Y. Shao, L. Yu *et al.*, “Vf^{2}boost: Very fast vertical federated gradient boosting for cross-enterprise learning,” in *SIGMOD*. ACM, 2021, pp. 563–576.

[29] F. Fu, H. Xue *et al.*, “Blindfl: Vertical federated machine learning without peeking into your data,” in *SIGMOD*. ACM, 2022, pp. 1316–1330.

[30] M. Abadi, P. Barham, J. Chen, Z. Chen *et al.*, “Tensorflow: A system for large-scale machine learning,” in *OSDI*. USENIX, 2016, pp. 265–283.

[31] European Parliament and The Council of the European Union, “The general data protection regulation (GDPR),” in *https://eugdpr.org*, 2016.

[32] L. de la Torre, “A guide to the california consumer privacy act of 2018,” in *Available at SSRN 3275571*, 2018.

[33] T. Li, A. K. Sahu, M. Zaheer *et al.*, “Federated optimization in heterogeneous networks,” in *MLSys*. mlsys.org, 2020.

[34] S. P. Karimireddy, S. Kale, M. Mohri *et al.*, “SCAFFOLD: stochastic controlled averaging for federated learning,” in *ICML*, vol. 119. PMLR, 2020, pp. 5132–5143.

[35] D. Gao, D. Chen, Z. Li, Y. Xie, X. Pan *et al.*, “Fs-real: A real-world cross-device federated learning platform,” *VLDB*, vol. 16, no. 12, pp. 4046–4049, 2023.

[36] D. Chen, D. Gao, Y. Xie *et al.*, “Fs-real: Towards real-world cross-device federated learning,” in *KDD*. ACM, 2023, pp. 3829–3841.

[37] Z. Jiang, Y. Xu, H. Xu *et al.*, “Fedmp: Federated learning through adaptive model pruning in heterogeneous edge computing,” in *ICDE*. IEEE, 2022, pp. 767–779.

[38] M. Chen, Y. Xu, H. Xu, and L. Huang, “Enhancing decentralized federated learning for non-iid data on heterogeneous devices,” in *ICDE*. IEEE, 2023, pp. 2289–2302.

[39] R. Liu, F. Wu, C. Wu, Y. Wang *et al.*, “No one left behind: Inclusive federated learning over heterogeneous devices,” in *KDD*. ACM, 2022, pp. 3398–3406.

[40] Y. Huang, L. Chu, Z. Zhou *et al.*, “Personalized cross-silo federated learning on non-iid data,” in *AAAI*. AAAI Press, 2021, pp. 7865–7873.

[41] Q. Li, Y. Diao, Q. Chen, and B. He, “Federated learning on non-iid data silos: An experimental study,” in *ICDE*. IEEE, 2022, pp. 965–978.

[42] Y. Wu, S. Cai, X. Xiao, G. Chen *et al.*, “Privacy preserving vertical federated learning for tree-based models,” *VLDB*, vol. 13, no. 11, pp. 2090–2103, 2020.

[43] Q. Li, Z. Wen, and B. He, “Practical federated gradient boosting decision trees,” in *AAAI*. AAAI Press, 2020, pp. 4642–4649.

[44] L. Shapley *et al.*, “A value for n-person games,” *Annals of Mathematical Studies*, vol. 28, pp. 307–317, 1953.

[45] S. Zheng, Y. Cao, and M. Yoshikawa, “Secure shapley value for cross-silo federated learning,” *VLDB*, vol. 16, no. 7, pp. 1657–1670, 2023.

[46] X. Xu, L. Lyu, X. Ma *et al.*, “Gradient driven rewards to guarantee fairness in collaborative machine learning,” in *NeurIPS*, 2021, pp. 16 104–16 117.

[47] X. Xu, Z. Wu, C. S. Foo, and B. K. H. Low, “Validation free and replication robust volume-based data valuation,” in *NeurIPS*, 2021, pp. 10 837–10 848.

[48] Y. Wang, K. Li, Y. Luo, G. Li *et al.*, “Fast, robust and interpretable participant contribution estimation for federated learning,” in *ICDE*. IEEE, 2024.