41st IEEE International Conference
on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

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]Beihang University, [2]City University of Hong Kong.

Outline

- Background
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

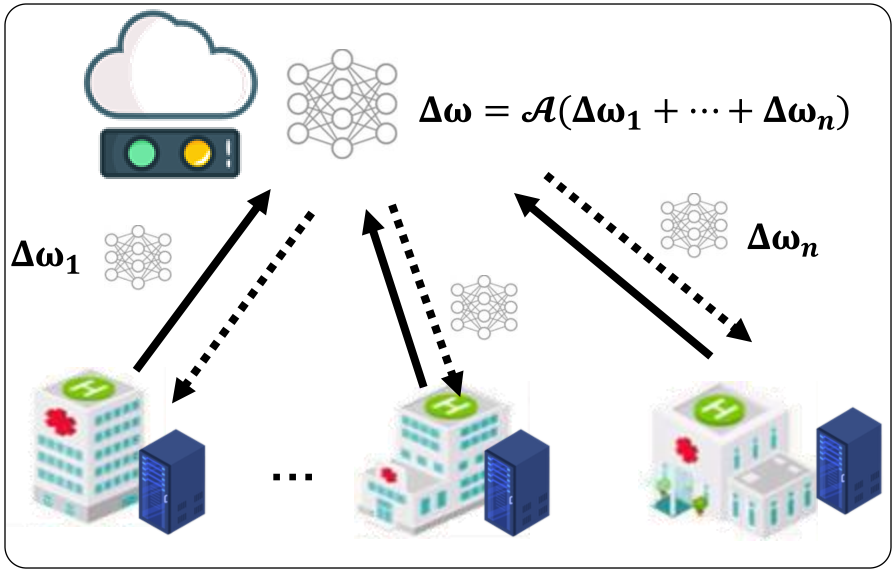
Outline

- Background
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Federated Learning

- **What is Federated Learning (FL) ?**

A distributed learning paradigm using datasets across data owners (*eg.* hospitals) **without accessing raw data** [1,2]



Major Features:

① Keeping the private data local

② Only sharing the parameters

③ Data quality are heterogenous

Data owners may be reluctant to share high-quality datasets unless the their data value are fairly measured

[1] Federated Machine Learning: Concept and Applications. ACM TIST 2019
[2] Advances and open problems in federated learning. Found. Trends Mach. Learn. 2021.

Shapley Value for Data Valuation

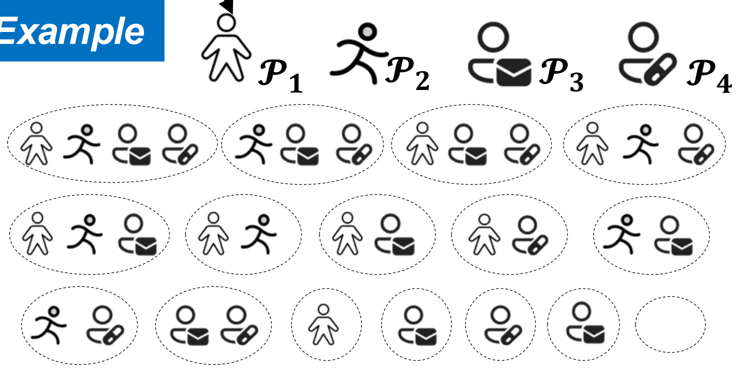
- ## Why Shapley value (SV) ?

The SV is a classical concept for <u>measuring contributions</u> in a cooperation, which holds several <u>essential fairness</u> properties

$$\phi_i = \frac{1}{n}\sum_{S\in N\setminus\{i\}} \frac{U(S\cup\{i\}) - U(S)}{\binom{|S|}{n-1}}$$

Assign contribution for players

Example



By assessing all 2^n combinations of players

The Essential Fairness Properties of SV

(i) Null Player: *Player $\mathcal{P}_i$ without impact on utility has zero contribution in the cooperation, i.e., $\phi_i = 0$*

(ii) Symmetric Fairness: *If two players $\mathcal{P}_i, \mathcal{P}_j$ can be alternatives for each other in the game, they will be assigned with the same contribution, i.e., $\phi_i = \phi_j$*

(iii) Group Rationality: *The sum of contributions of all players in the game is exactly equal to the utility with all players, i.e., $\sum_{i\in N}\phi_i = U(N)$*

The Shapley Value has been widely considered as the standard metric in both DB and AI community [3,4]

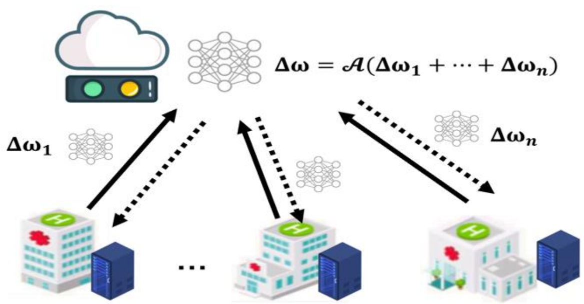
[3] The Shapley value in database management. SIGMOD Rec. 2023
[4] The Shapley value in machine learning. IJCAI 2022.

Shapley Value meets Federated Learning

- **Challenges of SV-based Data valuation solution**

If we calculate the SV-based data valuation directly in FL, the computational cost can be prohibited due to following reasons:

Data Valuation for Federated Learning



View ***FL client*** *as* ***player***
and ***loss/accuracy*** *as* ***utility***

$$\phi_i = \frac{1}{n} \sum_{\mathcal{D}_S \in N \setminus \{i\}} \frac{U(\mathcal{D}_{S \cup \{i\}}) - U(\mathcal{D}_S)}{\binom{|S|}{n-1}}$$

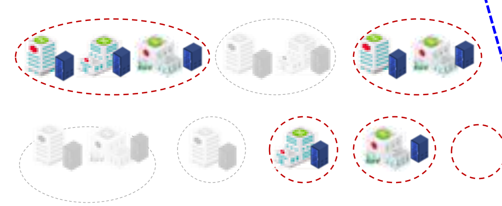
It is necessary to **train 2^n FL models**
under all possible combinations of FL clients

Existing approximation algorithms

Computation Complexity:

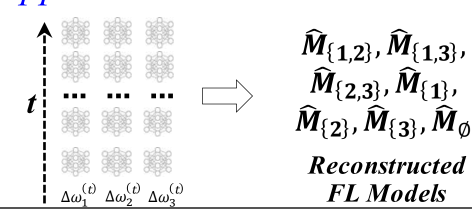
$$\mathcal{O}(2^n \cdot \tau)$$

Approx. idea 1: Sampling



choose limited combinations

Approx. idea 2: Gradient-Reconstruction



avoid extra model training

Objective: how to effectively approximate the SV-based data valuation for the FL scenario ?

Shapley Value meets Federated Learning

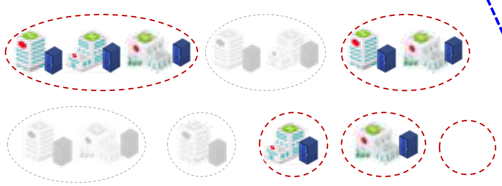
- **Limitation of existing solutions:**

Existing approximation algorithms

Computation Complexity:

$$\mathcal{O}(2^n \cdot \tau)$$

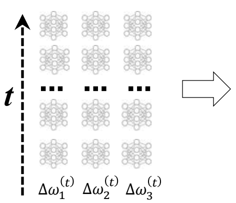
Approx. idea 1: Sampling



choose limited combinations

SIGMOD24, ICDE 22, AISTATS, ICML 19, etc

Approx. idea 2: Gradient-Reconstruction



$\widehat{M}_{\{1,2\}}, \widehat{M}_{\{1,3\}}, \widehat{M}_{\{2,3\}}, \widehat{M}_{\{1\}}, \widehat{M}_{\{2\}}, \widehat{M}_{\{3\}}, \widehat{M}_{\emptyset}$

Reconstructed FL Models

avoid extra model training

ACM TIST24, FLPI 20, Big Data19, etc

Limitation 1:

Ignoring *selecting the suitable* ***computation scheme*** *of Shapley Value.*

$$\sum_{S \subseteq N \setminus \{i\}} \frac{U(M_{S\cup\{i\}}) - U(M_S)}{n \cdot \binom{n-1}{|S|}}$$

Marginal Contribution Based

or

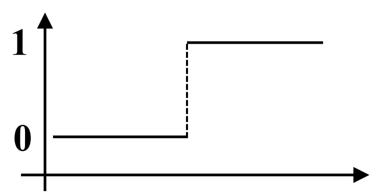
$$\sum_{S \subseteq N \setminus \{i\}} \frac{U(M_{S\cup\{i\}}) - U(M_{N\setminus(S\cup\{i\})})}{n \cdot \binom{n-1}{|S|}}$$

Complementary Contribution Based

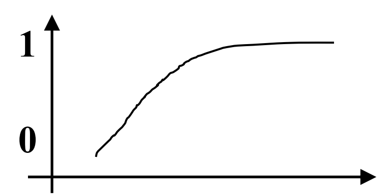
Limitation 2:

Ignoring *utilizing intrinsic* ***properties of utility function*** *in federated learning.*

e.g., majority game



federated learning



Can we design more effective approximation algorithms ?

Outline

- Background
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

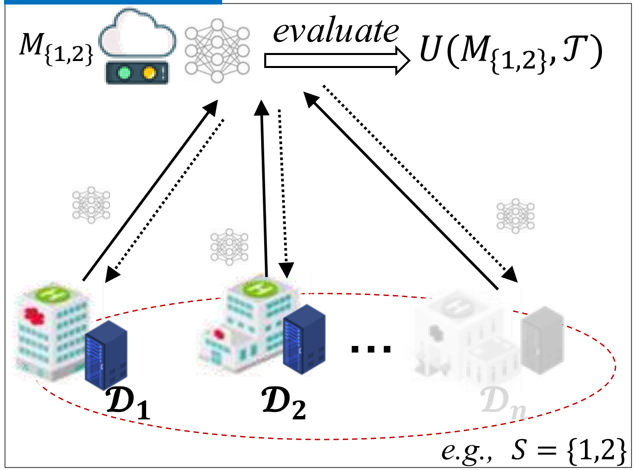
Problem Statement

- **Data Valuation for FL**

Given n FL clients with datasets $\mathcal{D}_N = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, and a FL algorithm $\mathcal{A}$, the federation trains model $M_S(\mathcal{A})$ under a subset of clients $S \subseteq N$, and evaluates it utility $U(M_S)$ on test dataset $\mathcal{T}$.

Then, data valuation problem is to qualify contribution of dataset $\mathcal{D}_i$ as $\boldsymbol{\phi(\mathcal{A}, \mathcal{D}_N, \mathcal{T}, \mathcal{D}_i)}$ (ϕ_i for short) with following properties:

Example



Three desirable properties

① *Null-Player*

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

② *Symmetric-Fairness*

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

③ *Linear-Additivity*

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

Shapley Value based data valuation naturally inherits its ***fairness properties*** and ensures above desirable properties

Problem Statement

- **The SV-based Data Valuation Schemes**

There are two commonly used equivalent Shapley value expression and each provides a computation scheme for the data valuation.

1. Marginal Contribution SV based Computation Scheme (MC-SV)

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

2. Complementary Contribution SV based Computation Scheme (CC-SV) [5]

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

Example with 3 FL 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

Take ϕ_1 as an example

1) For $|S| = 0$, compute $U(\{1\}) - U(\emptyset) = 0.40$
2) For $|S| = 1$, compute $U(\{1,2\}) - U(\{2\}) = 0.10, U(\{1,3\}) - U(\{3\}) = 0.30$
3) For $|S| = 2$, compute $U(\{1,2,3\}) - U(\{2,3\}) = 0.06$

Then, based on MC-SV, $\phi_1 = (0.4 \div 1 + (0.1 + 0.3) \div 2 + 0.06 \div 1) \div 3 = 0.22$

As both the MC-SV and CC-SV based scheme requires $O(2^n)$ FL models, ***efficient and accurate*** approximation algorithm is expected

[5] Efficient Sampling Approaches to Shapley Value Approximation. SIGMOD 2023.

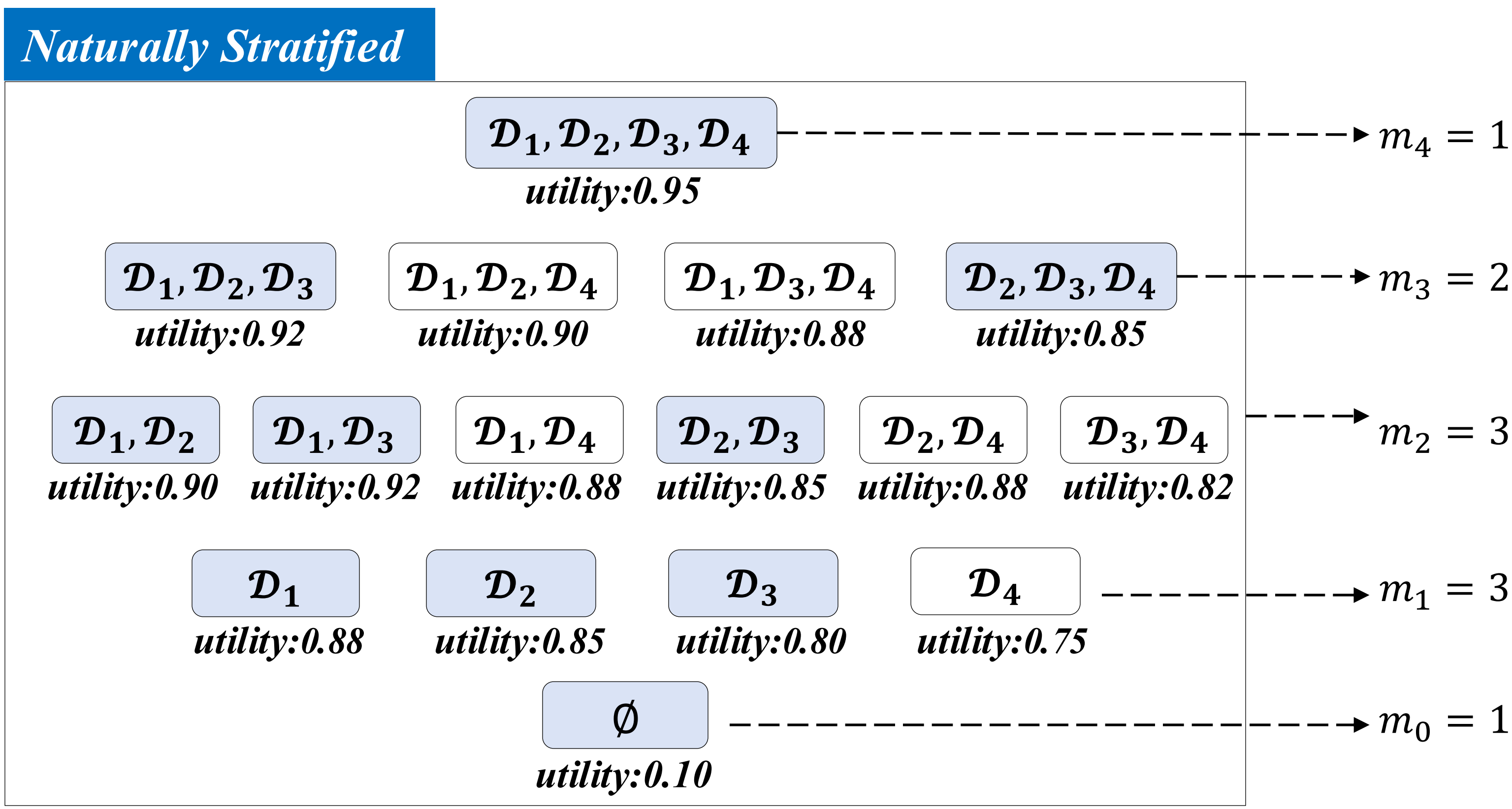
Outline

- Background
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Our solution

- **Overview:**

① We propose a ***unified stratified sampling*** framework to support both the MC-SV and CC-SV and then find MC-SV is more appropriate for the proposed approximating framework through theoretical analysis.

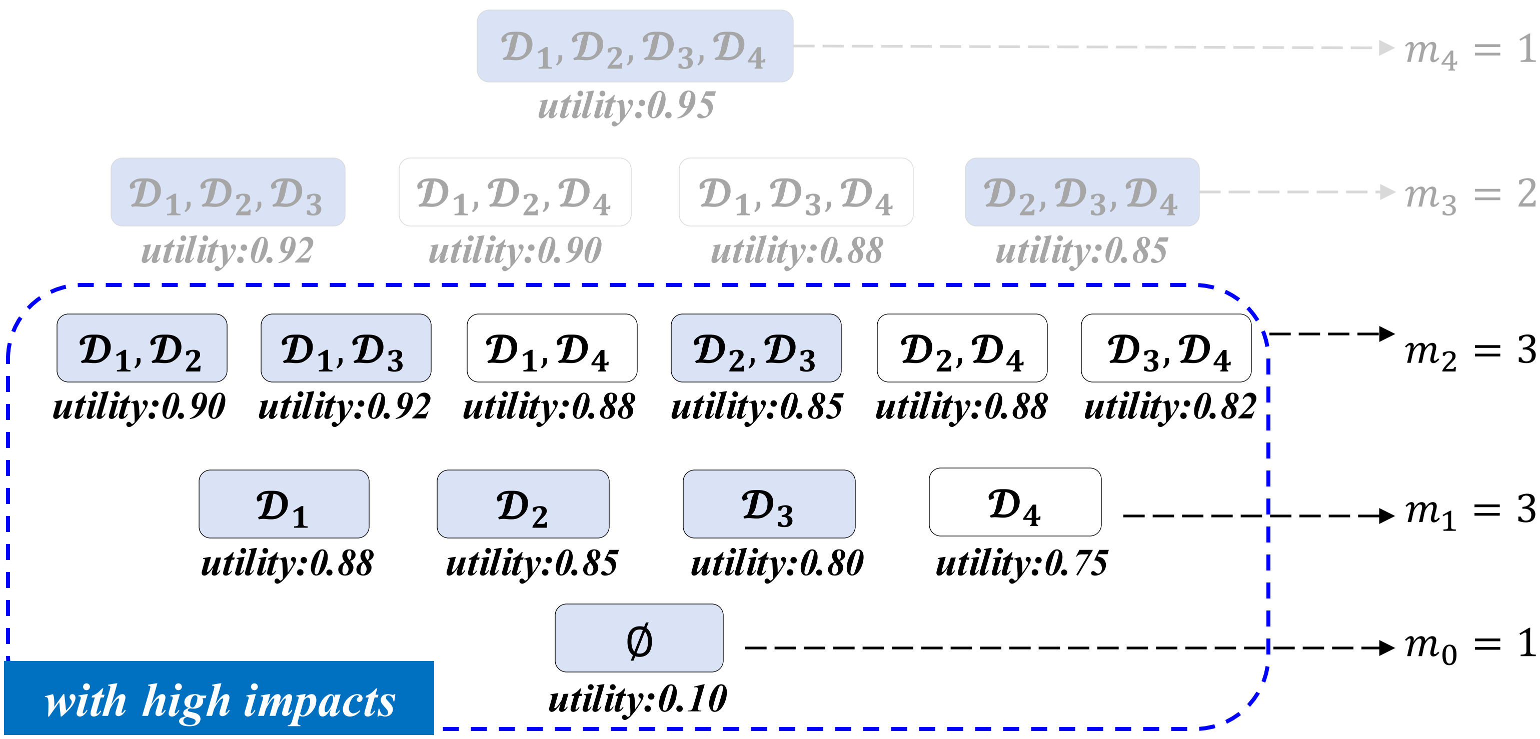


Our solution

- **Overview:**

① We propose a ***unified stratified sampling*** framework to support both the MC-SV and CC-SV and then find MC-SV is more appropriate for the proposed approximating framework through theoretical analysis.

② We observe a key insight for MC-SV, i.e., ***only limited dataset combinations highly affect the final data values*** under loss/acc utility.



Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2:** Train and Evaluate the FL Models → **Step 3:** Compute Data Value of FL Clients

Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2: Train and Evaluate the FL Models** → **Step 3: Compute Data Value of FL Clients**

Set Sampling Round $\gamma = 10$

All dataset combinations

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$ $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$ $\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$

$\mathcal{D}_1, \mathcal{D}_2$ $\mathcal{D}_1, \mathcal{D}_3$ $\mathcal{D}_1, \mathcal{D}_4$ $\mathcal{D}_2, \mathcal{D}_3$ $\mathcal{D}_2, \mathcal{D}_4$ $\mathcal{D}_3, \mathcal{D}_4$

$\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_4$

$\emptyset$

Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2: Train and Evaluate the FL Models** → **Step 3: Compute Data Value of FL Clients**

Set Sampling Round $\gamma = 10$

All dataset combinations

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$ $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$ $\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$

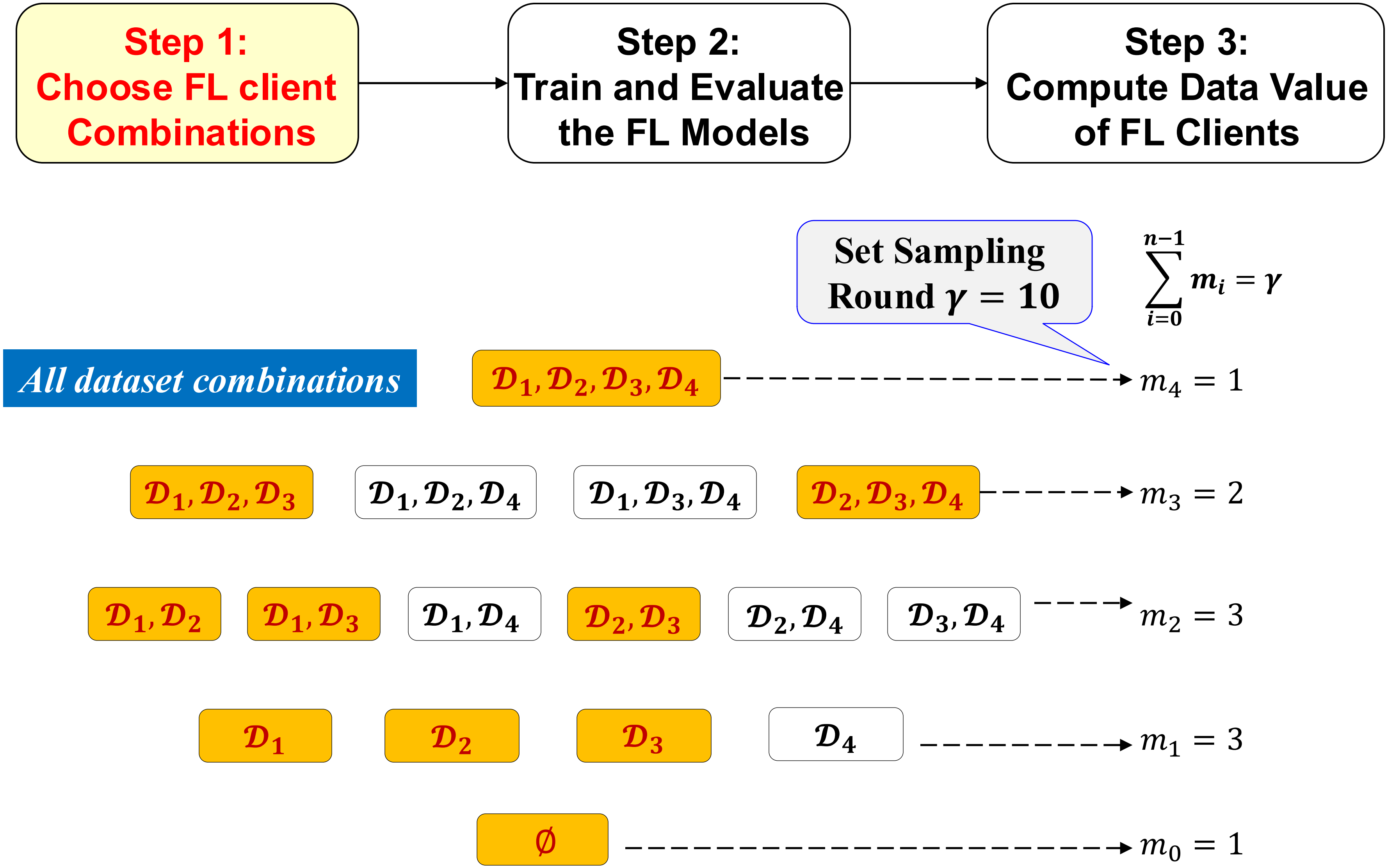
$\mathcal{D}_1, \mathcal{D}_2$ $\mathcal{D}_1, \mathcal{D}_3$ $\mathcal{D}_1, \mathcal{D}_4$ $\mathcal{D}_2, \mathcal{D}_3$ $\mathcal{D}_2, \mathcal{D}_4$ $\mathcal{D}_3, \mathcal{D}_4$

$\mathcal{D}_1$ $\mathcal{D}_2$ $\mathcal{D}_3$ $\mathcal{D}_4$ $\dashrightarrow m_1 = 3$

$\emptyset$ $\dashrightarrow m_0 = 1$

Our solution：Framework

- **Unified Stratified Sampling Framework**

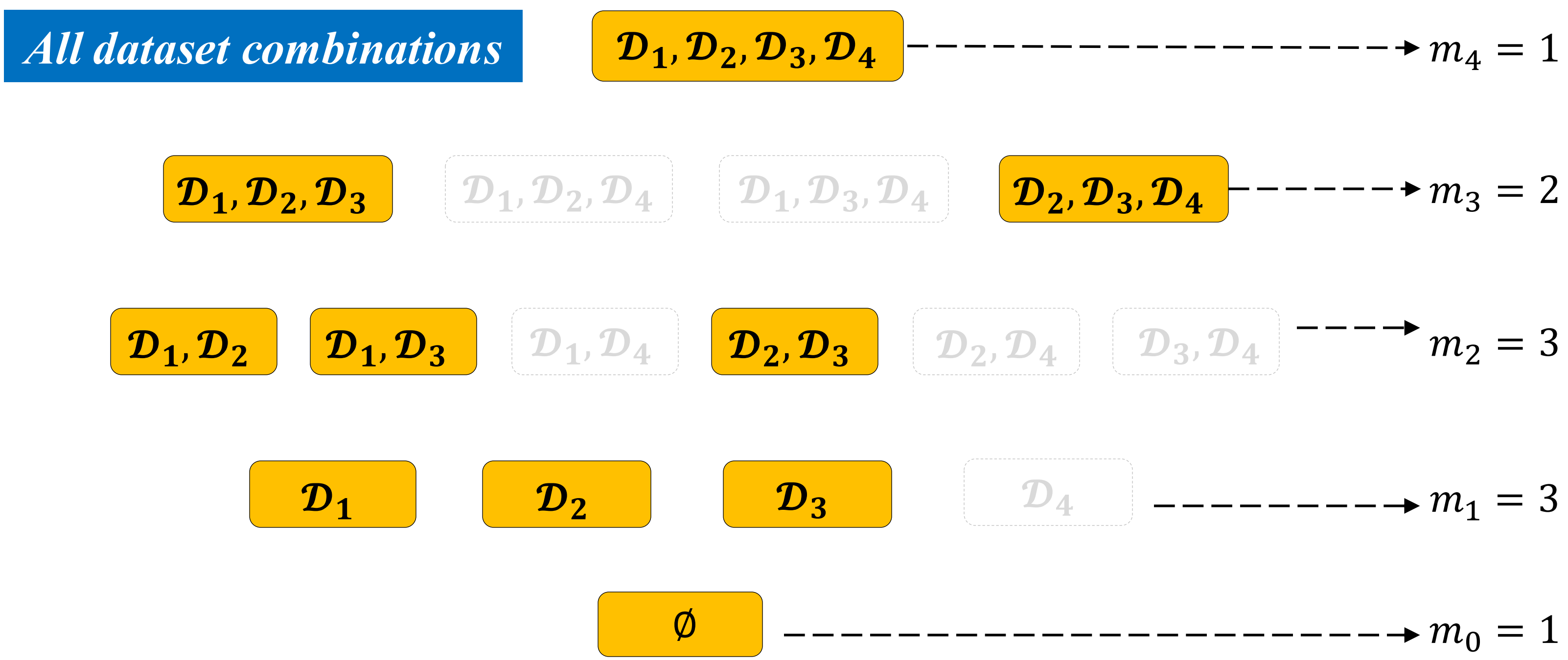


Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2: Train and Evaluate the FL Models** → **Step 3: Compute Data Value of FL Clients**

Train and evaluate FL models M_S with chosen combinations: $S \in \{\emptyset, \{\mathcal{D}_1\}, \{\mathcal{D}_2\}, \{\mathcal{D}_3\}, \{\mathcal{D}_1, \mathcal{D}_2\}, \{\mathcal{D}_1, \mathcal{D}_3\}, \{\mathcal{D}_2, \mathcal{D}_3\}, \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}, \{\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}, \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}, \}$



Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2: Train and Evaluate the FL Models** → **Step 3: Compute Data Value of FL Clients**

Train and evaluate FL models M_S with chosen combinations: $S \in \{\emptyset, \{\mathcal{D}_1\}, \{\mathcal{D}_2\}, \{\mathcal{D}_3\}, \{\mathcal{D}_1, \mathcal{D}_2\}, \{\mathcal{D}_1, \mathcal{D}_3\}, \{\mathcal{D}_2, \mathcal{D}_3\}, \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}, \{\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}, \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}, \}$

All dataset combinations

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ → $m_4 = 1$
utility:0.95

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ | $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$ | $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$ | $\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ → $m_3 = 2$
utility:0.92 *utility:0.85*

$\mathcal{D}_1, \mathcal{D}_2$ | $\mathcal{D}_1, \mathcal{D}_3$ | $\mathcal{D}_1, \mathcal{D}_4$ | $\mathcal{D}_2, \mathcal{D}_3$ | $\mathcal{D}_2, \mathcal{D}_4$ | $\mathcal{D}_3, \mathcal{D}_4$ → $m_2 = 3$
utility:0.90 *utility:0.92* *utility:0.85*

$\mathcal{D}_1$ | $\mathcal{D}_2$ | $\mathcal{D}_3$ | $\mathcal{D}_4$ → $m_1 = 3$
utility:0.88 *utility:0.85* *utility:0.80*

utility:0.10 $\emptyset$ → $m_0 = 1$

Our solution：Framework

- **Unified Stratified Sampling Framework**

Step 1: Choose FL client Combinations → **Step 2:** Train and Evaluate the FL Models → **Step 3:** Compute Data Value of FL Clients

$$\widehat{\phi_i} \leftarrow \frac{1}{n}\sum_{k=1}^{n}\frac{\sum_{S,\bar{S}\in \mathcal{S}} U(M_S) - U(M_{\bar{S}})}{m_{i,k}}$$

Unify the MC-SV and CC-SV by setting $\bar{S}$ to be $S\backslash\{i\}$ or $N\backslash S$

All dataset combinations

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ (*utility:0.95*) → $m_4 = 1$

$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ (*utility:0.92*); $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4$; $\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4$; $\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ (*utility:0.85*) → $m_3 = 2$

$\mathcal{D}_1, \mathcal{D}_2$ (*utility:0.90*); $\mathcal{D}_1, \mathcal{D}_3$ (*utility:0.92*); $\mathcal{D}_1, \mathcal{D}_4$; $\mathcal{D}_2, \mathcal{D}_3$ (*utility:0.85*); $\mathcal{D}_2, \mathcal{D}_4$; $\mathcal{D}_3, \mathcal{D}_4$ → $m_2 = 3$

$\mathcal{D}_1$ (*utility:0.88*); $\mathcal{D}_2$ (*utility:0.85*); $\mathcal{D}_3$ (*utility:0.80*); $\mathcal{D}_4$ → $m_1 = 3$

utility:0.10 $\emptyset$ → $m_0 = 1$

Our solution：Framework

- **Analysis of the stratified sampling framework**

$$\widehat{\phi_i} \leftarrow \frac{1}{n}\sum_{k=1}^{n} \frac{\sum_{S,\bar{S}\in \boldsymbol{S}} \boldsymbol{U(M_S) - U(M_{\bar{S}})}}{m_{i,k}}$$

Unify *the MC-SV and CC-SV by setting* $\bar{S}$ *to be* $S\backslash\{i\}$ *or* $N\backslash S$

Expectation Analysis

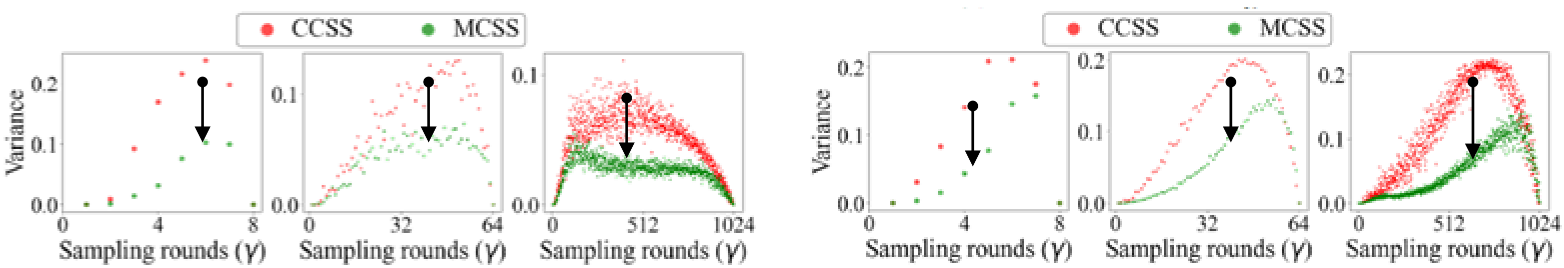
Theorem 1. The SS framework can provide unbiased estimation of SV in expectation for both the MC-SV and CC-SV based scheme

$$\mathbb{E}[\hat{\phi}_i^{MC}] = \mathbb{E}[\hat{\phi}_i^{CC}] = \sum_{S\subseteq N\backslash\{i\}} \frac{U(M_S) - U(\boldsymbol{M_{\bar{S}}})}{n\cdot\binom{|S|}{n-1}} = \phi_i$$

Variance Analysis

Theorem 2. For any sampling strategy using CC-SV based scheme, using MC-SV based scheme can yield lower estimation variance

$$\mathbb{V}[\hat{\phi}_i^{MC}] - \mathbb{V}[\hat{\phi}_i^{CC}] \geq \sum_{k=1}^{n}\sum_{S}\frac{1}{n^2\cdot m_{i,k}^2}|D_S|^2\sigma^2 > 0$$



(a) Client #3∼#10 using MLP model

(b) Client #3∼#10 using CNN model

Experimental results on FEMNIST show that MC-SV ***variance is clearly lower*** than CC-SV.

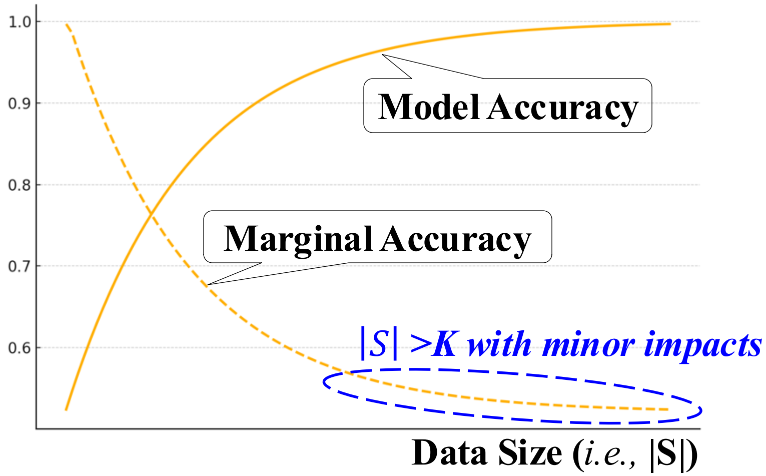
Takeaway： ***MC-SV*** is ***more appropriate*** for proposed framework

Our solution: Optimization

- **Key Observations on MC-SV based Scheme**

$$\phi_i = \frac{1}{n} \sum_{S \in (N \setminus \{i\})} \frac{U\left(M_{S\cup\{i\}}\right) - U(M_S)}{\binom{n-1}{|S|}}$$

Observation 1: *As size of data combination $|S|$ increases,* ***marginal contribution decreases noticeably***.

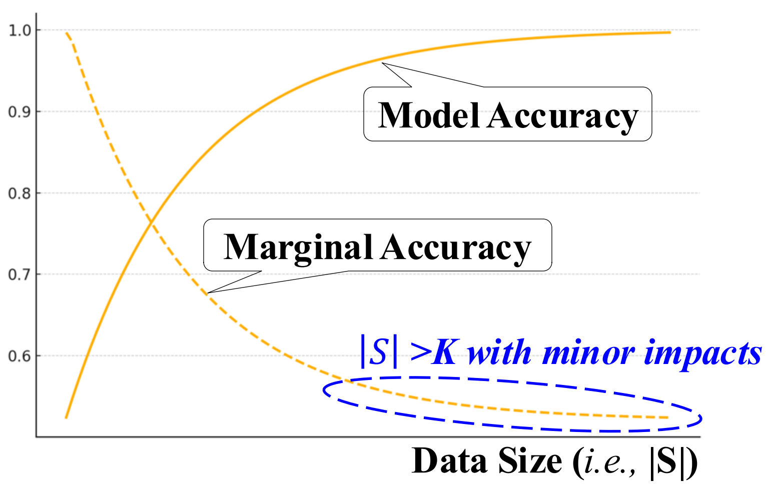


Our solution：Optimization

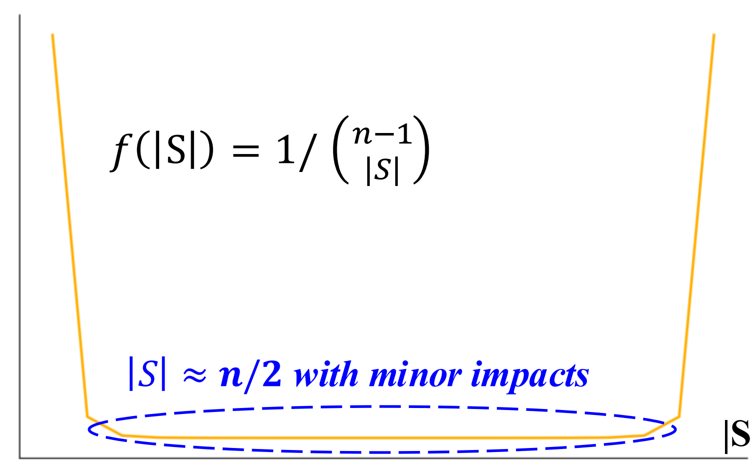
- **Key Observations on MC-SV based Scheme**

$$\phi_i = \frac{1}{n} \sum_{S \in (N \setminus \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}$$

Observation 1*: As size of data combination $|S|$ increases,* ***marginal contribution decreases noticeably****.*



Observation 2*: Different datasets combinations S* ***have varying impacts*** *on the final computed data value.*

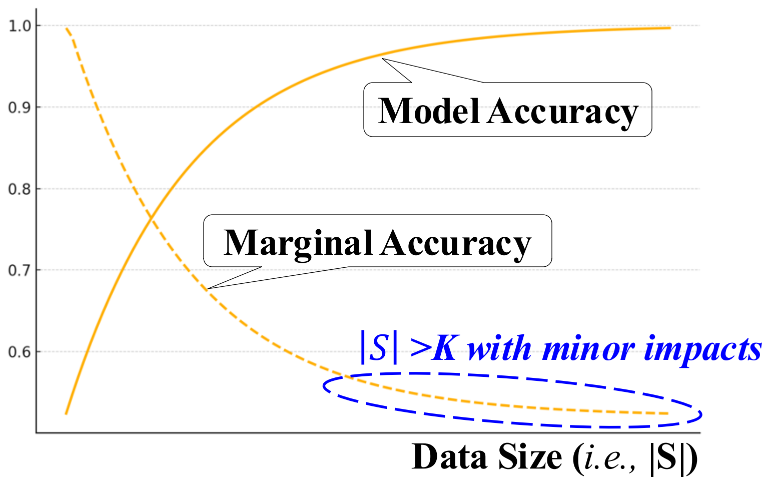


Our solution: Optimization

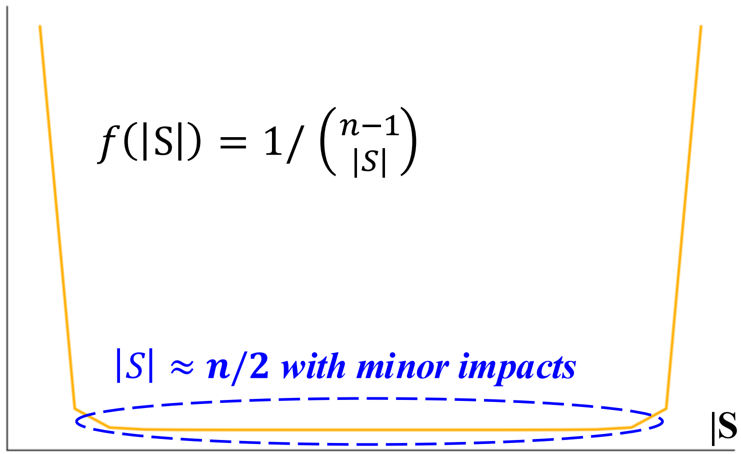
- **Key Observations on MC-SV based Scheme**

$$\phi_i = \frac{1}{n} \sum_{S \in (N \setminus \{i\})} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}$$

***Observation 1**: As size of data combination $|S|$ increases, **marginal contribution decreases noticeably**.*



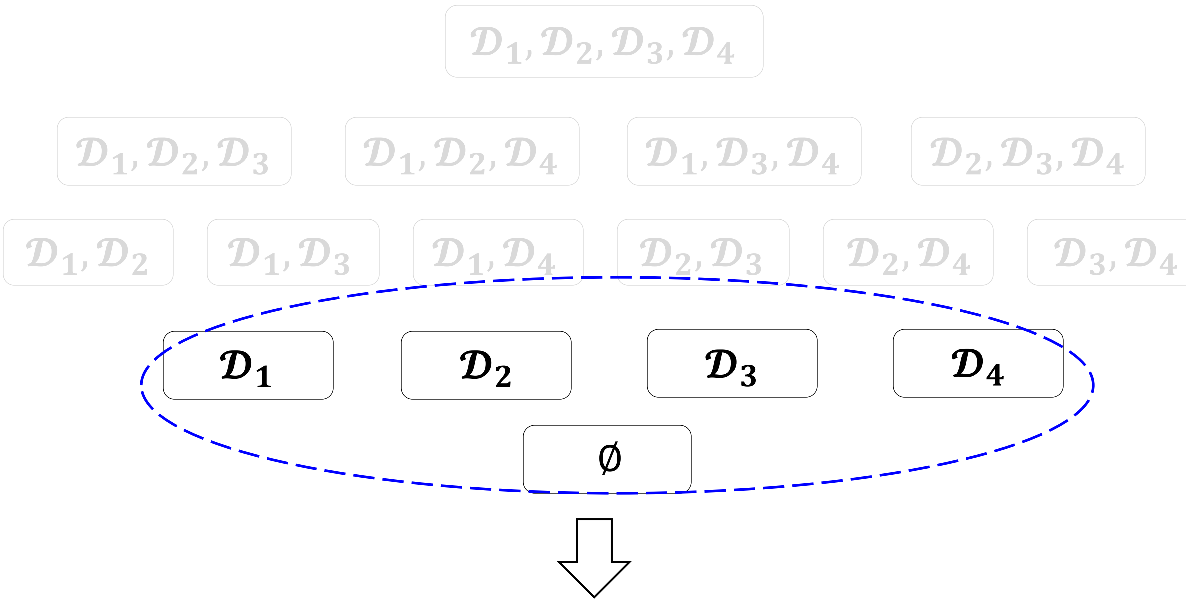
***Observation 2**: Different datasets combinations S **have varying impacts** on the final computed data value.*



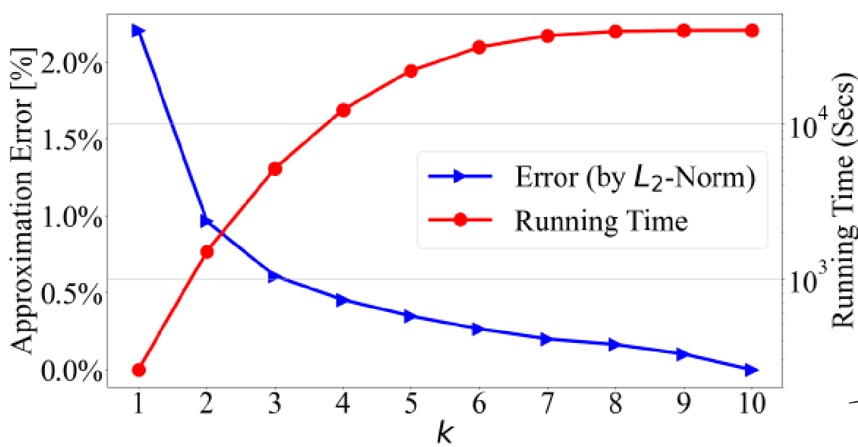
*Key Insights: Only combinations with **small |S|** have high impacts*

Our solution：Optimization

- **Empirical study of the above insights**



Main Idea:
Taking the first K *stratums for data valuation in FL.*



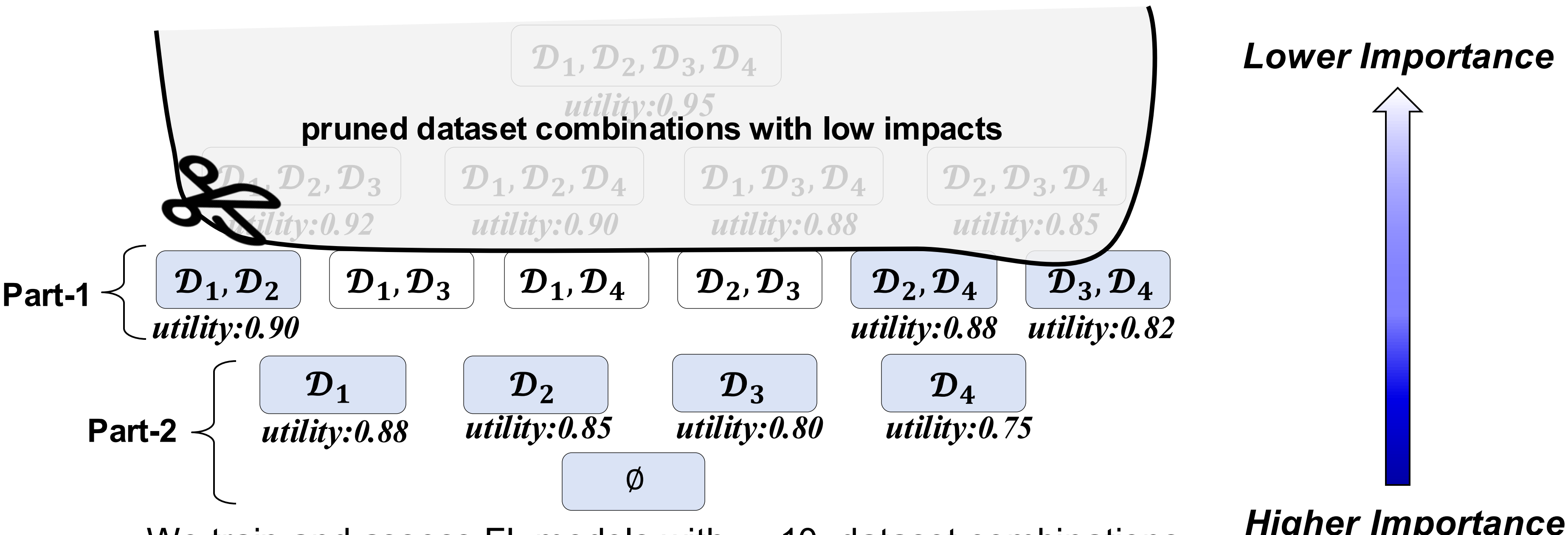
Experimental Setups:

- FEMNIST dataset
- FedAvg algorithm
- 10 FL clients

Observation: *The approximation error decrease quickly when* K *increases, (**K≥2, Error≤1.5%**)*

Our solution

- **Importance-Pruned Stratified Sampling (IPSS)**



We train and assess FL models with γ=10 dataset combinations.

Sampling and Evaluation

Part-1: $k^* \leftarrow \max\{k \mid \sum_{j=0}^{k} \binom{n}{j} \le \gamma\}$

The first k^* stratums will be selected

Part-2: *dataset combinations $\mathcal{P}$ in k^*+1 stratum*

(1) $|\mathcal{P}| \le \gamma - k^*$ *(2)* $\forall S \in \mathcal{P}, |S| = k^*+1$

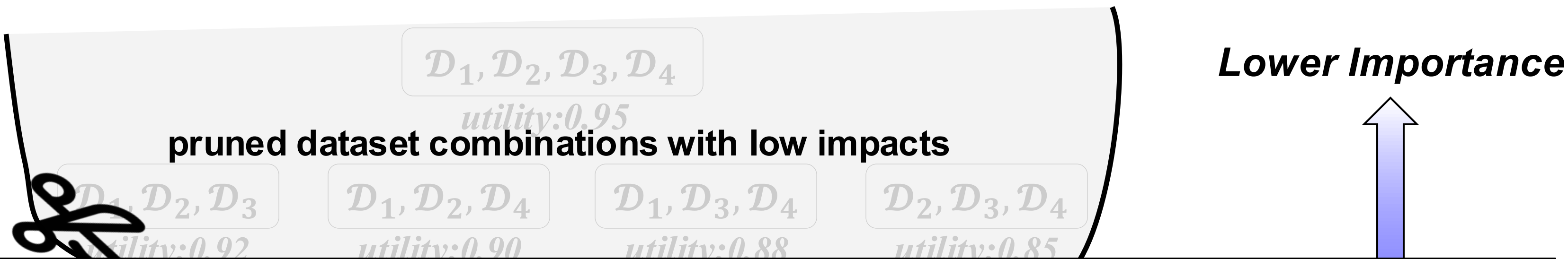
(3) $\forall i, j \in N,\ C_i = C_j$, *where* $C_k = \sum_{S \in \mathcal{P}} \mathbb{I}[k \in S]$

MC-SV based data valuation

$$\widehat{\phi_i} = \frac{1}{n} \sum_{S \subseteq (N \setminus \{i\}), |S| < k^*} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}} + \frac{1}{n} \sum_{S \subseteq (N \setminus \{i\}), (S \cup \{i\}) \in \mathcal{P}} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}$$

Our solution

- **Importance-Pruned Stratified Sampling (IPSS)**



Time Complexity: $\mathcal{O}(\tau \cdot \gamma)$

$\boldsymbol{\gamma}$*:total sampling round,* $\boldsymbol{\tau}$*: time to train and evaluate an FL model*

Approximation error: $\mathcal{O}(\frac{n-k^*}{k^* n t})$

k^*:$\max\{k \mid \sum_{j=0}^{k} \binom{n}{j} \leq \gamma\}$, $\boldsymbol{n}$: *number of all FL clients,* $\boldsymbol{t}$*: training samples of an FL client*

Sampling and Evaluation

Part-1: $k^* \leftarrow \max\{k \mid \sum_{j=0}^{k} \binom{n}{j} \leq \gamma\}$

The first k^* *stratums will be selected*

Part-2: *dataset combinations* $\mathcal{P}$ *in* k^*+1 *stratum*

(1) $|\mathcal{P}| \leq \gamma - k^*$ *(2)*$\forall S \in \mathcal{P}, |S| = k^*+1$

(3) $\forall i, j \in N$, $C_i = C_j$, *where* $C_k = \sum_{S \in \mathcal{P}} \mathbb{I}[k \in S]$

MC-SV based data valuation

$$\widehat{\phi_i} = \frac{1}{n} \sum_{S \subseteq (N \setminus \{i\}), |S| < k^*} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}} + \frac{1}{n} \sum_{S \subseteq (N \setminus \{i\}), (S \cup \{i\}) \in \mathcal{P}} \frac{U(M_{S \cup \{i\}}) - U(M_S)}{\binom{n-1}{|S|}}$$

Outline

- Background and Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Experiments

- **Synthetic Dataset:**
  - MNIST *with 60,000+ training samples and 10,000+ testing samples*
  - ***Five experimental setups*** *following [7,8]*
    - *1. FL clients with same data size and same distribution*
    - *2. FL clients with same data size and different distribution*
    - *3. FL clients with different data size and same distribution*
    - *4. FL clients with same data size, same distribution, and noise on label*
    - *5. FL clients with same data size, same distribution, and noise on feature*
- **Real-world Datasets:**
  - FEMNIST *(image data) with 805,000+ training samples from 3500+ users*
  - ADULT *(tabular data) with 48,800+ training samples and 14 features*
- **Learning Models:**
  - Multi-Layer Perceptron (MLP)
  - Convolutional neural network (CNN)
  - XGBoost (XGB)
- **Implementation:**
  - TensorFlow 2.4 and TensorFlow Federated 0.18
  - multi-processing simulation using the gRPC protocol

[7] Profit Allocation for Federated Learning. IEEE Bigdata 2019.
[8] GTG-Shapley: Efficient and accurate participant contribution evaluation in federated learning. ACM TIST 2022.

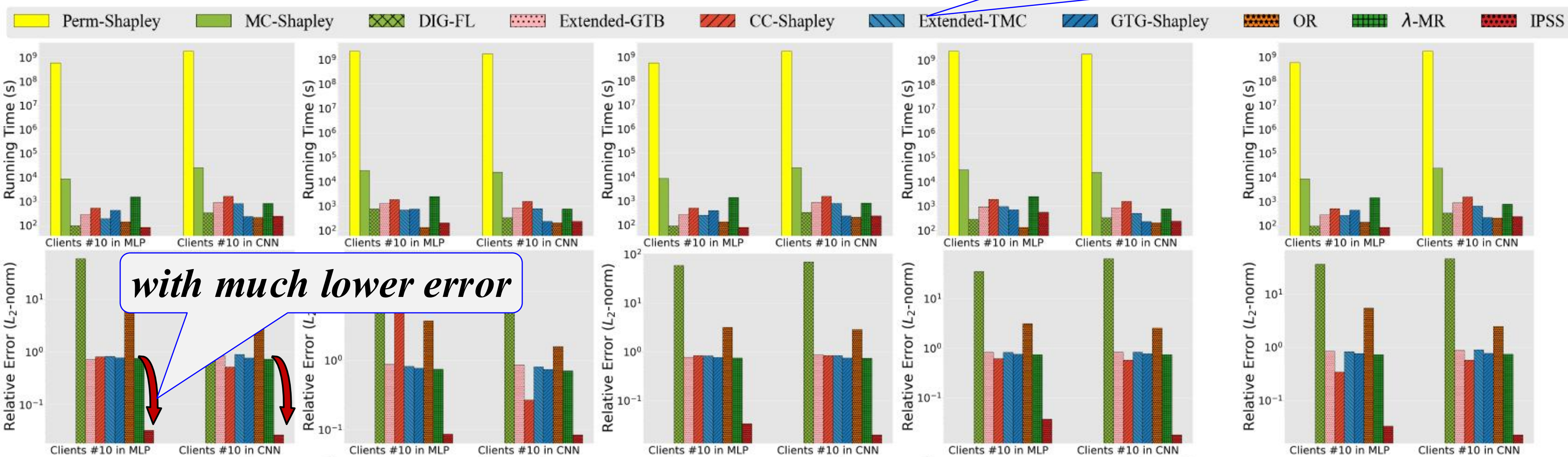
Experiments

- **Evaluation Metrics:**
  - *Running Time.*
  - *Approximation Error:* $l_2(\hat{\phi}, \phi) = \frac{\|\hat{\phi} - \phi\|_2}{\|\phi\|_2} = \sqrt{\sum_{i=1}^{n} (\hat{\phi}_i - \phi_i)^2} / \sqrt{\sum_{i=1}^{n} \phi_i^2}$
- **Nine Compared Algorithms:**
  - **Perm-Shapley** [definition]: *it directly calculates data value of clients in FL according to the definition of the permutation based Shapley value.*
  - **MC-Shapley** [definition]: *it directly calculates the data value through the MC-SV based computation scheme.*
  - **DIG-FL** [ICDE'22]: *it efficiently approximates the data value in FL, which only needs to evaluate O(n) numbers of dataset combinations under certain assumptions.*
  - **Extended-TMC** [ICML'19]: *it is an extension of widely-adopted data valuation scheme for general machine learning based on Truncated Monte Carlo algorithm.*
  - **Extended-GTB** [AISTATS'19]: *it is also an extension of a representative data valuation scheme, which use the group testing-based estimation techniques.*
  - **OR** [BigData'19]: *it takes gradients within the FL process with all clients the same as gradients under other combinations to avoids extra training of FL models.*
  - **λ-MR** [FLPI'20]: *it takes the MC-SV-based scheme and estimates data value in each training round of FL and aggregate them as the final results.*
  - **CC-Shapley** [SIGMOD'23]: *it is one of the* ***state-of-the-art*** *sampling methods to approximate the SV which estimates data value using the CC-SV-based schemes.*
  - **GTG-Shapley** [TIST'22]: *it also approximates the data value using gradients and Monte Carlo sampling approach to reduce number of reconstructed FL models.*

Experiments

- **Results on Synthetic datasets**

Set same total sampling round γ



(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*

- ***IPSS achieves the much lower approximation error with similar time cost***
- ***The results is consistent over various data size, distribution and noise***
- ***IPSS shows similar performance on both MLP and CNN models***

Experiments

- **Results on FEMNIST**

	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**

IPSS has lowest error

IPSS is fastest when $n \geq 6$

Experiments

- **Results on FEMNIST**

	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**

IPSS has lowest error

IPSS is fastest when $n \geq 6$

- **Results on ADULT**

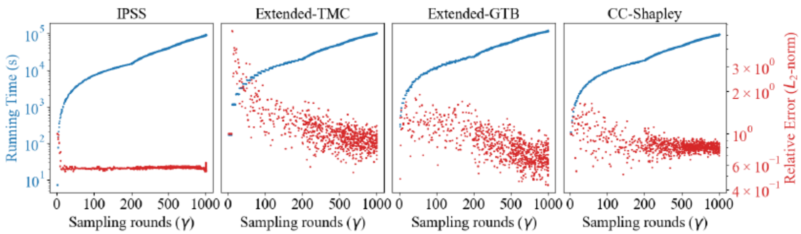
	n	Metrics	*Perm-Shap.*	*MC-Shap.*	*DIG-FL*	*Ext-TMC*	*Ext-GTB*	*CC-Shap.*	*GTG-Shap.*	*OR*	*λ-MR*	*IPSS*
MLP	3	Time(s)	720	164	94	95	138	199	59	**13**	48	69
		Error(l_2)	-	-	1.02	1.46	1.89	0.09	5.30	1.00	2.93	**0.05**
	6	Time (s)	3.3×10^5	2820	252	220	306	530	271	**74**	347	146
		Error(l_2)	-	-	1.12	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	[illegible]	[illegible]	[illegible]	\	\	\	**1.8**
		Error(l_2)	-	-	0.9	[illegible]	[illegible]	[illegible]				**0.04**
	6	Time(s)	13308	96	19	14	22	[illegible]	\	\	\	**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**

not applicable

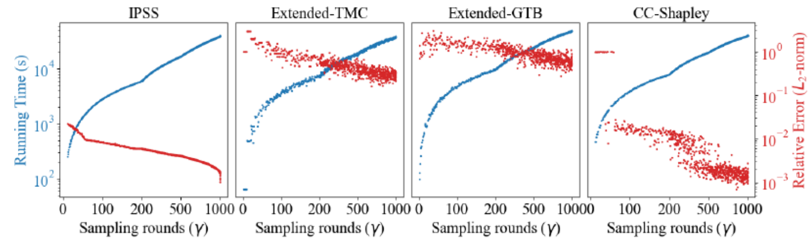
IPSS achieves the best accuracy and efficiency when FL client number ≥ 3

Experiments: In-depth analysis

1) Impacts of varying the sampling rounds



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

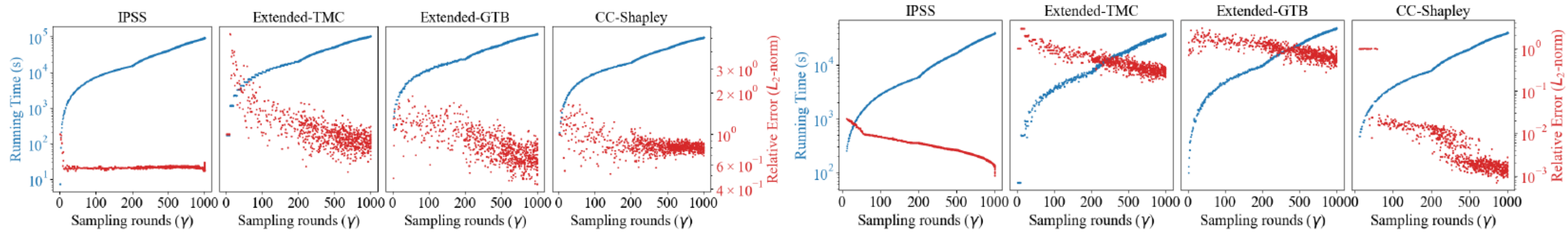


(b) Results of using the CNN as FL model.

IPSS has lower error using smaller sampling rounds

Experiments: In-depth analysis

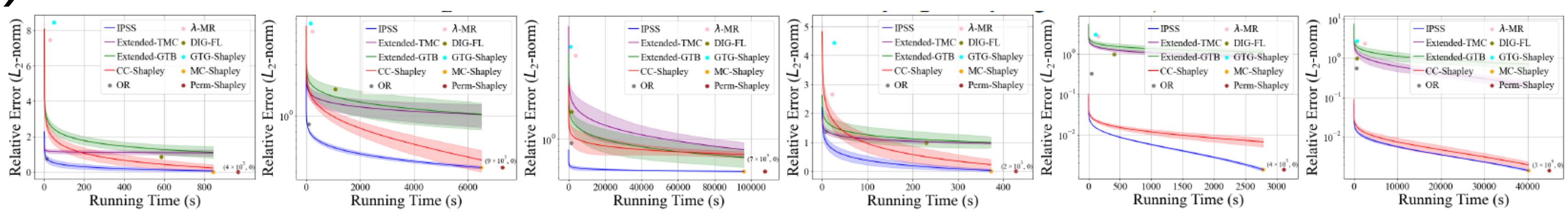
1) Impacts of varying the sampling rounds



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

IPSS has lower error using smaller sampling rounds

2) Pareto curves for time-error trade-off

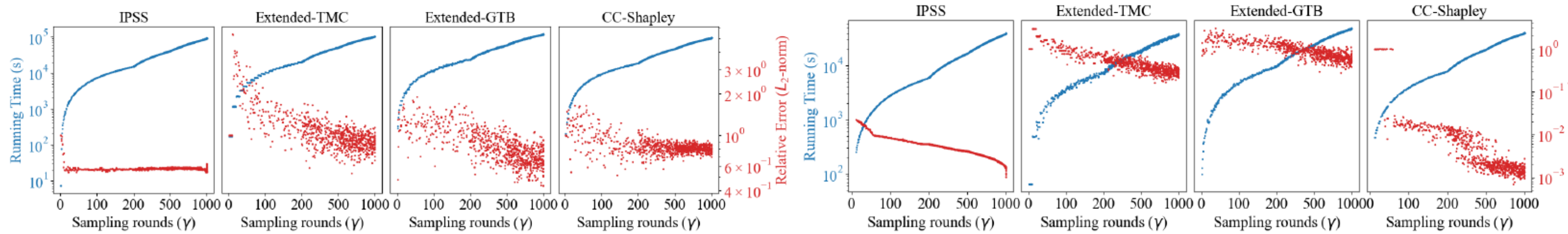


(a) Client#3+MLP *(b) Client#6+MLP* *(c) Client#10+MLP* *(d) Client#3+CNN* *(e) Client#6+CNN* *(f) Client#10+CNN*

IPSS achieves Pareto optimal for efficiency and accuracy

Experiments: In-depth analysis

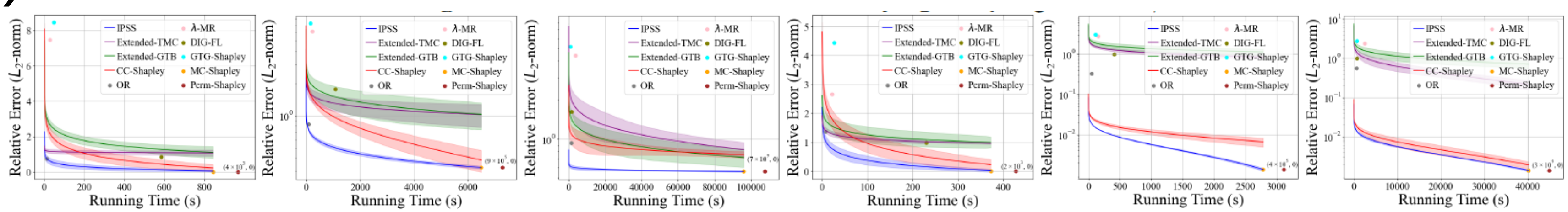
1) Impacts of varying the sampling rounds



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

IPSS has lower error using smaller sampling rounds

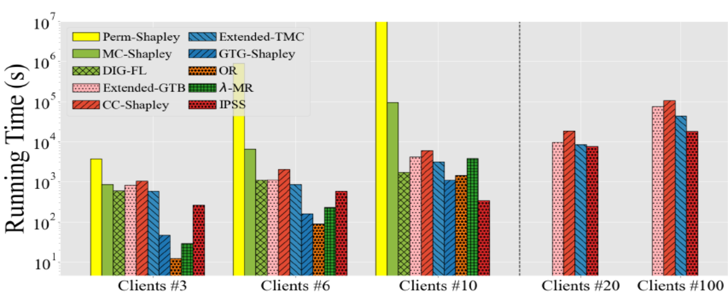
2) Pareto curves for time-error trade-off



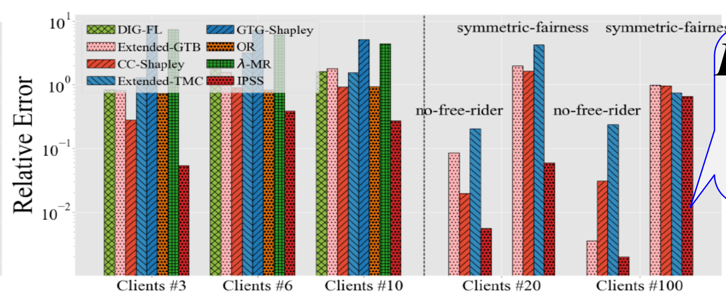
(a) Client#3+MLP (b) Client#6+MLP (c) Client#10+MLP (d) Client#3+CNN (e) Client#6+CNN (f) Client#10+CNN

IPSS achieves Pareto optimal for efficiency and accuracy

3) Scalability test for more FL clients *(up to 100 FL clients)*



(a) Time cost of varying the client number.



(b) Approximation error of varying the client number.

IPSS still performs better for a larger number of clients

Outline

- Background and Motivation
- Problem Statement
- Our Solutions
- Experiments
- Conclusion

Conclusion

- ***We propose a unified stratified sampling-based approximation framework that seamlessly integrates both the MC-SV-based and CC-SV-based computation schemes.***

- ***We identify a crucial phenomenon, where only limited dataset combinations highly impact final data value results in FL.***

- ***We propose a practical approximation algorithm, IPSS, which significantly improves the efficiency with high accuracy.***

- ***We conduct extensive evaluations on real and synthetic datasets to validate that the proposed IPSS algorithm outperforms nine representative baselines in efficiency and effectiveness.***

Q & A

THANK YOU

if you have further problems, feel free to email

weishuyue@buaa.edu.cn

source codes are available at

https://github.com/t0ush1/Shapley-Data-Valuation