

Ripple Shapley: Data Influence Attribution in One Federated Training Run

Dewen Zeng¹, Wenlong Tian^{1*}, Haozhao Wang², Jianfeng Lu³, Weijun Xiao⁴, Zhiyong Xu⁵

¹University of South China

²Huazhong University of Science and Technology

³Wuhan University of Science and Technology

⁴Virginia Commonwealth University

⁵Suffolk University

dewenzeng@stu.usc.edu.cn, wenlongtian@usc.edu.cn

hz_wang@hust.edu.cn, lujianfeng@wust.edu.cn, wxiao@vcu.edu, zxu@suffolk.edu

Abstract

Contribution evaluation is essential for incentivizing high-quality data sharing in federated learning (FL), yet existing Shapley-value-based methods are prohibitively expensive and overlook temporal influence propagation. In this paper, we propose *Ripple Shapley*, a novel attribution framework that enables accurate, real-time data valuation within a single federated training run. Our method decomposes each sample’s impact into an instantaneous *drop* term and a recursive *ripple* term, the latter capturing downstream influence via a Jacobian chain over global updates. To scale computation, we introduce a low-rank approximation of the Jacobian product and construct a shared subspace for efficient ripple accumulation. Extensive experiments on CIFAR-10 and MNIST show that Ripple Shapley achieves up to $62\times$ speedup over existing Shapley-based FL methods while maintaining high attribution fidelity, significantly improving efficiency, robustness, and fairness in federated environments. We further demonstrate its effectiveness in dynamic federated learning scenarios and its potential for real-time data pricing.

Introduction

Federated Learning (FL) (McMahan et al. 2017; Wang et al. 2024b) enables collaborative learning across decentralized data silos without exposing raw data, offering a privacy-preserving framework for machine learning at scale. However, the lack of visibility into individual participants’ data and behavior introduces a critical challenge: how to assess the value of each client’s contribution. Accurate contribution evaluation is essential for building trust, allocating incentives, and defending against data poisoning or free-riding behaviors in real-world deployments (Wang et al. 2025a; Tian et al. 2024; Wang et al. 2024a).

A principled approach to contribution evaluation uses the Shapley value from cooperative game theory, which quantifies a participant’s marginal impact on the global model by averaging over all possible participation orders (Song, Tong, and Wei 2019; Sun et al. 2023; Wang et al. 2025b). While theoretically sound, direct Shapley computation is intractable in FL due to the exponential number of client

subsets. To reduce cost, recent methods (Wang, Dang, and Zhou 2019; Wei et al. 2020; Sun et al. 2023) employ sampling or heuristics to approximate per-round marginal utilities. However, these approaches still require multiple forward and backward passes across rounds and fail to capture the recursive nature of model updates: *early-round updates persist and propagate, influencing future convergence paths*. Consequently, they overlook critical temporal dependencies and may yield attribution that is biased or delayed.

This omission of influence propagation not only weakens attribution fidelity but also poses practical risks: for instance, in data marketplaces, inaccurate valuation of contribution—whether an overestimation or underestimation—can cause significant financial losses, unfair reward allocation, and even disputes that jeopardize the trust and sustainability of the federated ecosystem (Kairouz et al. 2021).

To address these limitations, we propose **Ripple Shapley**, a novel framework for real-time, fine-grained attribution in federated learning. Our approach is inspired by the metaphor of ripples: each training sample initiates a “drop” in the optimization trajectory whose effects propagate through subsequent updates. We formalize this through a two-part decomposition. The *drop term* quantifies each sample’s immediate marginal utility using intermediate parameters collected during local updates. The *ripple term* models the downstream influence of that update as it flows through the global model, using a chain of Jacobian transformations over successive communication rounds.

A key challenge lies in the efficient computation of this recursive influence. Naively evaluating Jacobian products scales cubically with model dimension and linearly with propagation depth, making it impractical for practical deployment. To overcome this, we further propose a low-rank spectral approximation of the Jacobian chain by dynamically constructing a compact global subspace from historical Hessian eigendirections. This enables fast and memory-efficient ripple tracking without storing full Hessians or re-training surrogate models. Our method preserves essential Shapley fairness properties—symmetry, dummy, and linearity—while also admits bounded approximation error under smoothness assumptions.

We empirically validate Ripple Shapley on two standard

*Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

benchmarks (MNIST and CIFAR-10) under non-IID and adversarial settings. Results show that our method achieves up to $62\times$ speedup over existing Shapley-based FL approaches while maintaining high attribution fidelity. Moreover, we demonstrate its advantage in dynamic federated settings, where attribution must adapt to shifting participation and distribution drift. Ripple Shapley offers a scalable, theoretically grounded solution for real-time data valuation in federated learning—bridging the gap between rigorous attribution and practical deployment.

To summarize, our main contributions are as follows:

- We propose the first federated attribution framework that explicitly models the recursive, cross-round influence propagation of sample contributions within a single federated training run, overcoming the limitations of prior fed-Shapley methods.
- We decompose contributions into drop and ripple terms, and introduce a low-rank Jacobian chain approximation with an efficient global subspace update mechanism, supported by theoretical guarantees on error bounds and fairness invariance.
- Experiments show Ripple Shapley achieves up to $62\times$ speedup over prior approximate Shapley methods while maintaining attribution accuracy, and demonstrate its robustness in dynamic federated settings and potential for real-time data pricing.

Related Work

Accurately quantifying client contributions is central to fair and robust federated learning. Prior works (Wang et al. 2022; Sun et al. 2023; Guo et al. 2024) address this challenge via incentive mechanisms, dynamic aggregation, and transparency tools. Among these, Shapley value-based methods stand out for their rigorous fairness guarantees, modeling each participant’s marginal utility over all possible coalitions (Song, Tong, and Wei 2019; Liu et al. 2022; Wang, Dang, and Zhou 2019; Sun et al. 2023). In centralized learning, such valuations face exponential costs and have prompted approximations via Monte Carlo sampling (Ghorbani and Zou 2019; Ghorbani, Kim, and Zou 2020), group testing (Du and Hwang 1999), or one-pass gradient-based surrogates (Wang et al. 2025b).

In federated settings, Shapley values have been extended to client-level attribution across training rounds (Wei et al. 2020; Lei et al. 2025; Sun et al. 2023; Nagalapatti and Narayanam 2021a), often by computing per-round marginal utilities and applying temporal discounting. These methods enhance robustness by filtering noisy clients and adjusting aggregation weights. However, they remain inherently *round-local*: each round’s utility is treated independently, and the recursive influence of early updates—propagated through future aggregation—is ignored. This simplification leads to biased or delayed attributions, particularly under dynamic participation or non-IID distributions.

Alternative methods based on statistical heuristics (Ding, Fang, and Huang 2020; Yu et al. 2020), prototypical representations (Chen et al. 2023; Guo et al. 2024), or update

similarity (Zhang, Wu, and Pan 2021; Lv et al. 2021) offer reduced complexity but often lack theoretical guarantees and generalization across tasks. More fundamentally, none of these approaches account for how a single sample’s impact compounds through the FL trajectory.

In contrast, our proposed **Ripple Shapley** explicitly models this recursive influence by tracing the propagation of each sample’s effect via a chain of Jacobians across rounds. This allows us to capture temporal dependencies in attribution, while our low-rank approximation strategy enables efficient, one-pass estimation without retraining or permutation sampling—bridging the gap between theoretical rigor and practical scalability.

Preliminaries

Centralized In-Run Data Shapley

In-Run Data Shapley (IRDS) (Wang et al. 2025b) is a one-pass Shapley-value-based data attribution method for centralized training that quantifies each sample’s instantaneous marginal utility without requiring multiple retrains or permutations. Due to its scalability and gradient-based formulation, IRDS provides a natural foundation for evaluating local sample utility in our method.

Formally, let $\mathcal{D}_{\text{tr}} = \{z_1, \dots, z_n\}$ denote the training dataset, where $z_i = (x_i, y_i)$, and let $\ell(w, z)$ denote the loss incurred by model parameters w on sample z . At iteration t , parameters are updated using mini-batch stochastic gradient descent over $B_t \subseteq \mathcal{D}_{\text{tr}}$:

$$w_{t+1} = w_t - \eta_t \sum_{z \in B_t} \nabla \ell(w_t, z), \quad (1)$$

where η_t is the step-size. IRDS defines the marginal utility of a subset $S \subseteq B_t$ by its impact on validation loss:

$$\mathcal{U}^{(t)}(S; z^{\text{val}}) := \ell(w_{t+1}, z^{\text{val}}) - \ell(w_t, z^{\text{val}}). \quad (2)$$

The per-step Shapley value $\phi_z^{(t)}$ is computed based on this utility, and the overall contribution of sample z across training steps is accumulated as:

$$\phi_z(U) \approx \sum_{t=0}^{T-1} \phi_z^{(t)}. \quad (3)$$

To reduce computational overhead, IRDS approximates $\phi_z^{(t)}$ via a first-order Taylor expansion:

$$\phi_z^{(t)} \approx -\eta_t \nabla \ell(w_t, z^{\text{val}}) \cdot \nabla \ell(w_t, z), \quad (4)$$

where the dot product captures alignment between a sample’s gradient and the validation objective. While originally proposed in centralized settings, IRDS could be naturally extended to federated training by estimating per-sample influence within each client. We adopt this formulation to define the *Drop Term*, capturing immediate utility, and further complement it with a *Ripple Term* that models cross-round influence propagation.

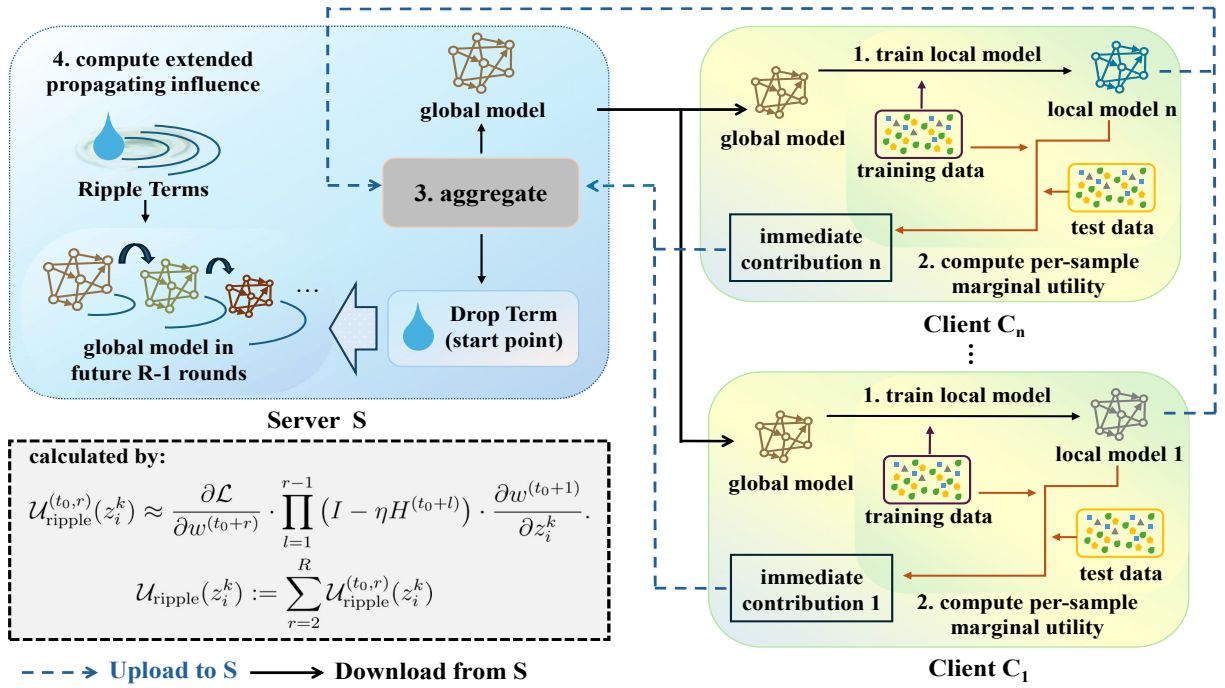


Figure 1: Workflow of Ripple Shapley: clients compute and upload the immediate contributions of local samples; the server aggregates these contributions to form drop term for the current round and accumulates ripple terms over subsequent rounds.

Method

Overview

We propose **Ripple Shapley**, a unified attribution framework tailored for federated learning, which decomposes each training sample’s contribution into two components: an *instantaneous drop term* and a *recursive ripple term*. This decomposition captures both the immediate effect of a local update and its downstream influence as it propagates through the global optimization trajectory. The overall workflow is illustrated in Figure 1.

At each communication round, selected clients download the global model and perform local training on their private data. During local updates, each client estimates per-sample marginal utility based on the intermediate model state, forming the drop term. These local utilities and model updates are then uploaded to the server. To account for temporal effects, the server tracks how each sample’s influence recursively propagates across future global rounds, yielding the ripple term. The total contribution of a sample is the sum of its drop and ripple terms across all *birth rounds*—i.e., rounds in which the sample was used in training.

Drop Term of Ripple Shapley

We begin by quantifying each sample’s *immediate* effect during local training, which serves as the origin of its downstream ripple. To this end, we adopt the centralized utility estimation strategy from IRDS (Wang et al. 2025b) to evaluate sample-level impact within each federated client.

Let z_k^i denote the i -th sample on client k , and let w_k^t be the local model parameters at local iteration t . During a com-

munication round, local training proceeds via multiple SGD steps. The utility of sample z_k^i is defined as the cumulative improvement in validation loss it induces over T local steps:

$$\mathcal{U}_{\text{local}}(z_k^i) = \sum_{t=0}^{T-1} \left[\ell(w_k^{t+1}(z_k^i), z^{\text{val}}) - \ell(w_k^t, z^{\text{val}}) \right], \quad (5)$$

where $w_k^{t+1}(z_k^i) = w_k^t - \eta_t \nabla \ell(w_k^t, z_k^i)$ denotes the model updated by a single gradient step on z_k^i , and z^{val} is a local or shared validation sample. Accordingly, this metric measures the immediate contribution of z_k^i to the client model’s validation performance.

In FL, local updates are aggregated to form the new global model. Under FedAvg (McMahan et al. 2017), each client’s contribution is weighted proportionally to its local data size:

$$\alpha_k = \frac{n_k}{n_s}, \quad (6)$$

where n_k is the number of samples on client k , and n_s is the total number of samples across selected clients. Accordingly, we define the global influence of sample z_k^i in round t as its **drop term**:

$$\mathcal{U}_{\text{drop}}(z_k^i) = \alpha_k \cdot \mathcal{U}_{\text{local}}(z_k^i). \quad (7)$$

This formulation reflects the sample’s direct impact on the aggregated global update. Unlike prior FedShapley methods that average client utilities per round (Song, Tong, and Wei 2019; Wei et al. 2020), our drop term offers finer granularity and avoids over-smoothing contributions. However, it re-

mains limited to instantaneous effects and does not account for how early updates influence later training stages.

To address this, we treat the drop term as the entry point of a recursive influence propagation process. In the next section, we formalize how this influence diffuses through subsequent rounds via a Jacobian chain, yielding the ripple term.

Ripple Term of Ripple Shapley

To capture how a training sample’s influence propagates through the federated optimization trajectory, we define the **ripple term** as its recursive impact on future global models. As illustrated in Equation (8), a local sample z_i^k first contributes to a client update $\Delta w_k^{(t_0)}$, which in turn alters the global model $w^{(t_0+1)}$. This update subsequently affects downstream global models and client retrainings in later rounds, captured by the following influence path:

$$z_i^k \rightarrow \Delta w_k^{(t_0)} \rightarrow w^{(t_0+1)} \dots \rightarrow w^{(t_0+r)} \rightarrow \mathcal{L}(w^{(t_0+r)}). \quad (8)$$

To formalize this recursive influence, we model the training process as a differentiable computation graph and apply the chain rule to obtain the total effect of z_i^k on the validation loss at round $t_0 + r$:

$$\frac{\partial \mathcal{L}(w^{(t_0+r)})}{\partial z_i^k} = \frac{\partial \mathcal{L}}{\partial w^{(t_0+r)}} \cdot \prod_{l=1}^{r-1} \frac{\partial w^{(t_0+l+1)}}{\partial w^{(t_0+l)}} \cdot \frac{\partial w^{(t_0+1)}}{\partial z_i^k}. \quad (9)$$

The final Jacobian term $\frac{\partial w^{(t_0+1)}}{\partial z_i^k}$ is identical to the one computed in the drop term (cf. Equation (7)) and thus incurs no additional computation. Under FedAvg, the global update follows the formula:

$$w^{(t+1)} = w^{(t)} - \eta \sum_{k \in S^{(t)}} \frac{n_k}{n_s} \nabla \mathcal{L}_k(w^{(t)}), \quad (10)$$

with the global curvature approximated by:

$$H^{(t)} := \sum_{k \in S^{(t)}} \frac{n_k}{n_s} \nabla^2 \mathcal{L}_k(w^{(t)}).$$

Assuming local objectives are β -smooth, the Jacobian can be linearized as:

$$\begin{aligned} J^{(t)} &:= \frac{\partial w^{(t+1)}}{\partial w^{(t)}} \approx I - \eta H^{(t)}, \\ \left\| J^{(t)} - (I - \eta H^{(t)}) \right\| &\leq \frac{\beta}{2} \eta^2. \end{aligned} \quad (11)$$

We then define the sample’s ripple contribution from birth round t_0 to evaluation round $t_0 + r$ as:

$$\mathcal{U}_{\text{ripple}}^{(t_0, r)}(z_i^k) \approx \frac{\partial \mathcal{L}}{\partial w^{(t_0+r)}} \cdot \prod_{l=1}^{r-1} \left(I - \eta H^{(t_0+l)} \right) \cdot \frac{\partial w^{(t_0+1)}}{\partial z_i^k}. \quad (12)$$

Aggregating over a propagation window of depth R , the total ripple term becomes:

$$\mathcal{U}_{\text{ripple}}(z_i^k) := \sum_{r=2}^R \mathcal{U}_{\text{ripple}}^{(t_0, r)}(z_i^k). \quad (13)$$

Remark. Despite the discrete nature of FedAvg updates, smooth client objectives allow the FL process to be approximated as a continuous trajectory (Smith et al. 2017; Li et al. 2019). This legitimizes our use of Jacobian chains to capture recursive influence.

Efficient Computation

Naive evaluation of the Jacobian product $P = \prod_{t=1}^r (I - \eta H^{(t)})$ requires $\mathcal{O}(rd^3)$ computation and $\mathcal{O}(rd^2)$ memory—prohibitive for high-dimensional models. However, the Hessians of deep networks are known to have rapidly decaying spectra (Ghorbani, Krishnan, and Xiao 2019), enabling low-rank approximation. We thus approximate each Hessian via spectral truncation:

$$I - \eta H^{(t)} \approx I - \eta U^{(t)} \Lambda^{(t)} (U^{(t)})^\top, \quad (14)$$

where $U^{(t)} \in \mathbb{R}^{d \times k}$ contains the top- k eigenvectors and $\Lambda^{(t)} \in \mathbb{R}^{k \times k}$ the associated eigenvalues.

To enable recursive computation across communication rounds, all spectral directions are consolidated into a unified low-dimensional basis. Specifically, instead of storing or recomputing full Hessians, we adopt a **progressive subspace construction** strategy, where at each round t , the top- k eigendirections $U^{(t)}$ are computed and used to incrementally update the global basis Q via orthonormalization of the concatenated subspaces:

$$Q^{(t)} = \text{OrthoProj}([Q^{(t-1)} \ U^{(t)}]), \quad Q^{(0)} = \emptyset, \quad (15)$$

where $\text{OrthoProj}(\cdot)$ denotes an orthonormalization operator such as QR decomposition. This dynamic update mechanism efficiently captures the dominant curvature subspace across rounds while controlling memory usage.

After obtaining the consolidated basis $Q := Q^{(r)} \in \mathbb{R}^{d \times m}$ (with $m \leq rk$), each Hessian basis is projected as

$$U^{(t)} = QB^{(t)}, \quad B^{(t)} = Q^\top U^{(t)} \in \mathbb{R}^{m \times k}. \quad (16)$$

This yields a low-rank Jacobian operator:

$$P \approx Q \left(\prod_{t=1}^r \left(I_m - \eta B^{(t)} \Lambda^{(t)} (B^{(t)})^\top \right) \right) Q^\top, \quad (17)$$

where I_m is the $m \times m$ identity matrix. Letting $P_{\text{low}, 0} = I_m$ and defining the recursive chain:

$$P_{\text{low}, t} = \left(I_m - \eta B^{(t)} \Lambda^{(t)} (B^{(t)})^\top \right) P_{\text{low}, t-1}, \quad (18)$$

we obtain the final ripple approximation:

$$\mathcal{U}_{\text{ripple}}^{(t_0, r)}(z_i^k) \approx \frac{\partial \mathcal{L}}{\partial w^{(t_0+r)}} Q P_{\text{low}} Q^\top \frac{\partial w^{(t_0+1)}}{\partial z_i^k}. \quad (19)$$

Remark. The proposed low-rank approximation preserves essential Shapley axioms under mild assumptions, specifically satisfying:

- **Symmetry:** samples with identical gradients at the same training stage produce equivalent influence on the model update, as their projected contributions under the same Jacobian transformation are indistinguishable;

Algorithm 1: Ripple Shapley FL Protocol

Require: Initial model $w^{(0)}$, total rounds T , ripple depth R , learning rate η

Ensure: Final model $w^{(T)}$, sample Shapley values $\{\phi(z)\}$

- 1: **for** $t = 0$ **to** $T - 1$ **do**
- 2: Server selects client set $S^{(t)}$ and broadcasts $w^{(t)}$
- 3: **for all** client $k \in S^{(t)}$ **in parallel do**
- 4: **Local update:** Train $w^{(t)}$ on local data \mathcal{D}_k for 5 epochs
- 5: **Drop term:** Store intermediate parameters and estimate local utility $\mathcal{U}_{local,k}^t$
- 6: **Hessian sketch:** Estimate eigenpairs $(U_k^{(t)}, \Lambda_k^{(t)})$ of local Hessian
- 7: Upload model update $\Delta w_k^{(t)}$, $\mathcal{U}_{local,k}^t$ & Hessian sketch to server
- 8: **end for**
- 9: **Aggregation of drop & model:** Update via FedAvg:

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \sum_k \alpha_k \Delta w_k^{(t)}$$

- 10: **Ripple initialization:** Build global subspace $Q^{(t)}$ from $\{U_k^{(t)}\}$
 - 11: Project Hessians: $B_k^{(t)} \leftarrow Q^{(t)\top} U_k^{(t)}$
 - 12: Initialize ripple propagator $P \leftarrow I$
 - 13: **for** $r = 1$ **to** R **do**
 - 14: **Ripple update:**
$$P \leftarrow (I - \eta B_k^{(t)} \Lambda_k^{(t)} B_k^{(t)\top}) \cdot P$$
 - 15: **Propagate influence:** Update $\phi(z_i^k)$ for samples at round $t - r$ using P
 - 16: **end for**
 - 17: **end for**
 - 18: **return** Final model $w^{(T)}$, Ripple Shapley $\{\phi(z)\}$
-

- **Dummy Player:** samples whose gradient vectors lie entirely outside the low-rank Hessian subspace (i.e., orthogonal to all columns of Q) have no effect on the propagated update, thus yielding zero ripple contribution;
- **Linearity:** the contribution of a union of disjoint sample subsets equals the sum of their individual ripple values, as the Jacobian operator acts linearly on the aggregated gradients within the subspace;

The complete Ripple Shapley federated learning protocol is summarized in Algorithm 1. Detailed error bounds for Jacobian chain approximation, spectral truncation, and recursive propagation are provided in the Appendix A.

Experiments

We evaluate our method from three perspectives: (1) computational efficiency compared to prior Shapley-based robust FL methods; (2) effectiveness in enhancing robustness under various data corruption scenarios; and (3) a case study in dynamic FL environments and real-time pricing.

Datasets. Experiments are conducted on three benchmark image datasets: MNIST and CIFAR-10. Following standard federated learning protocols, we simulate challenging Non-IID settings via label-distribution skew as (McMahan et al. 2017). To assess robustness, we incorporate four representative data poisoning scenarios (Pillutla, Kakade, and Harchaoui 2022; Sun et al. 2023):

- (1) **Long-tailed imbalance:** class imbalance across clients;
- (2) **Open-set noise:** injected samples with labels from unseen classes;
- (3) **Data noise:** corrupted or low-quality inputs from adversarial clients.

These settings reflect practical challenges in real-world FL deployments, such as unbalanced sampling, annotation errors, and adversarial data injection.

Main Results

Compared Methods. We compare Ripple Shapley with the following five representative baselines in robust Federated Learning:

- **FedAvg** (McMahan et al. 2017): the standard aggregation method, serving as a base reference.
- **FedProx** (Li et al. 2020): addresses Non-IID issues by penalizing local-global model divergence.
- **s-FedAvg** (Nagalapatti and Narayanam 2021b): integrates MC-sampling Shapley scores to filter uninformative clients.
- **FedSV** (Wang et al. 2020): estimates client contributions via a round-wise Shapley utility framework.
- **AFedSV+** (Sun et al. 2023): a state-of-the-art adaptive Shapley-based method for robust federated learning.

Training Setup. Following the protocol of (Sun et al. 2023) for fair comparison. On CIFAR-10, we employ a standard CNN as the base model; on MNIST, we use a 2-layer MLP. Each client performs 5 local epochs per round, and the system is trained for 100 global communication rounds. We fix the batch size to 10 and the learning rate to 0.01 across all settings. Each experiment is independently repeated five times, and we report mean time costs and accuracy. For consistency, all methods are implemented in PyTorch and evaluated on two Tesla V100 GPUs under identical environments.

Runtime Analysis. We evaluated the computational efficiency of the efficient calculation strategy proposed in the Method section for Ripple Shapley on the standard MNIST, and CIFAR-10 datasets, comparing it against three existing Shapley-based robust federated learning algorithms. To eliminate the overhead differences caused by varying aggregation strategies, all experiments consistently employed FedAvg as the global model aggregation rule. The baseline runtime corresponds to the duration of a conventional training run, serving as a reference to quantify the additional time overhead incurred by Shapley value computations. Specifically, we set the low-rank approximation parameters to $k = 20$ and $m = 50$. As shown in Table 1, after 100 full training rounds, the total average runtime of Ripple Shapley

Round	Ripple Shapley	S-FedAvg	FedSV	AFedSV+	Plain Training	FedProx
1	11.33	42.64	247.47	860.03	5.60	6.42
10	105.21	430.25	4427.16	7820.16	49.13	52.35
25	264.35	1104.92	11771.45	17762.11	123.35	137.14
50	497.63	2244.29	24092.00	31988.98	242.93	271.86
75	731.26	3401.39	36510.85	46520.65	361.43	402.61
100	984.98	4531.91	48282.84	61437.57	480.91	530.39

Table 1: Average Cumulative Computation Time(s) across Communication Rounds on Two Datasets.

is only 2.05 times that of plain training. Ripple Shapley not only enables sample-level attribution in federated training, but also achieves approximately 4.6×, 49.06×, and 62.37× improvements in computational efficiency compared to the other methods, respectively, which clearly demonstrate the significant efficiency advantage of Ripple Shapley, making real-time contribution evaluation feasible in FL.

Measure of Attribution Accuracy. Unlike prior works that evaluate Shapley values using numerical error metrics such as mean squared error (MSE), we argue that such metrics fail to reflect their practical utility in FL. Given the fundamentally different calculation principles, numerical alignment of Shapley scores may not correspond to meaningful training dynamics. Instead, we adopt a task-driven evaluation: the quality of Shapley estimates is assessed by their ability to improve robustness under various data and model poisoning scenarios.

To ensure a fair comparison, each baseline is paired with its original aggregation strategy. For Ripple Shapley, we employ a simple aggregation rule that integrates Shapley values into client weighting without additional complexity. The aggregation weight for client i is defined as:

$$w_i = \lambda \cdot u_i + (1 - \lambda) \cdot \frac{n_i}{\sum_j n_j}, \quad (20)$$

where u_i denotes the estimated utility of client i in the current round, and n_i is its local dataset size. We fix the mixing coefficient $\lambda = 0.5$ to eliminate tuning effects and isolate the utility of attribution-based weighting. A more faithful contribution estimate is expected to support more effective aggregation, leading to enhanced robustness and faster convergence. To validate this, we set only 100 global rounds for the training budget.

As shown in Figures 2, our method (denoted as RShapley) achieves robustness that is comparable to or even surpasses existing approaches across nearly all evaluated datasets. This suggests that even without carefully engineered aggregation strategies, our method can accomplish both effective training and attribution within a single federated training run—or even fewer communication rounds. A notable observation is that Ripple Shapley often outperforms baselines during the early stages of training. Since early-round aggregation decisions tend to shape the trajectory of global model evolution, this early advantage frequently results in higher final accuracy by the end of truncated training.

For example, under the open-set label noise scenario, Ripple Shapley achieves an average accuracy improvement of over 10% compared to baseline methods, which advantage can be attributed to two key factors. First, existing methods typically approximate the classic Shapley value to enhance theoretical accuracy—commonly through Monte Carlo sampling—whose estimation quality heavily depends on the number of samples. This creates a trade-off between estimation fidelity and computational efficiency. Second, the classical Shapley value measures the expected marginal contribution, which may not faithfully reflect the actual utility of clients under the current model state. This limitation reduces its attribution accuracy in a single training run. These two reasons also explain why S-FedAvg, despite adopting the same aggregation strategy as Ripple Shapley, still exhibits gap in robustness performance.

Case Study: Dynamic Federated Setting & Real-Time Data Pricing. Most existing FL studies are grounded in a static participation assumption, where the set of participants remains fixed throughout training. However, this assumption rarely holds in practical deployments. One of the defining characteristics that distinguishes FL from conventional distributed learning is its dynamic client participation—where clients may drop out due to unstable connections, limited resources, or insufficient incentives.

The dynamic nature of FL fundamentally challenges classical Shapley value methods, which assume a fixed participant set and compute contributions by averaging marginal gains over all possible subsets. In practice, however, client participation in FL is highly stochastic, non-reproducible, and temporally sensitive. Many theoretical coalitions never occur, and the impact of individual clients varies over time. Consequently, expectation-based attribution fails to capture real-world influence dynamics. This key challenge underscores the limitations of the classical Shapley framework and motivates the design of Ripple Shapley.

We present the case study to investigate the applicability of Ripple Shapley under dynamic FL settings. The partial participation was simulated by randomly selecting clients at each round with a participation rate between $[0.5, 1.0]$, and tracked the evolution of the ripple for representative clients. A key feature of our approach is that we can predefine a maximum propagation depth R for each client. When a client drops out, the contribution of its local samples does not terminate immediately but continues to propagate until the depth threshold is reached. It ensures that sample-level

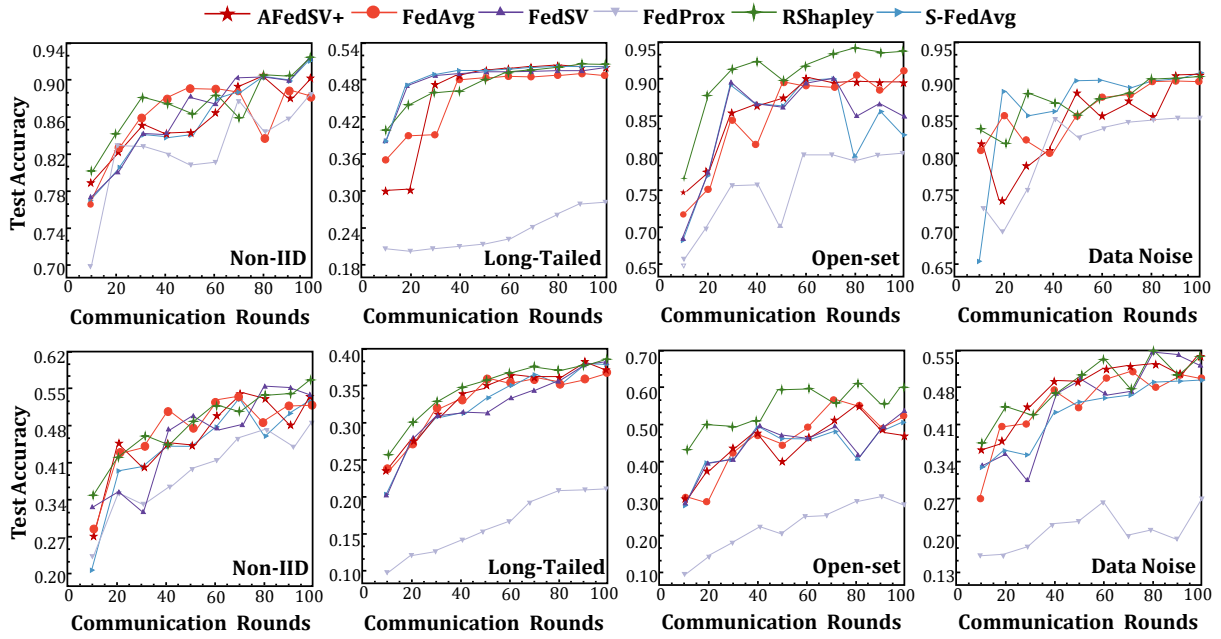


Figure 2: Robustness Evaluation on MNIST (top) and CIFAR-10 (bottom).

attribution remains valid and traceable even under client dropout, preserving the integrity of Shapley-based valuation.

For conventional static FL scenarios, if sufficient memory and compute are available, one may set the propagation depth equal to the total number of training rounds. However, to balance attribution fidelity with runtime cost, we further visualize the ripple term trajectories of several representative samples over 100 rounds. We observe that beyond a propagation depth of 20, most ripple values converge and exhibit negligible further change. This empirically supports the use of a moderate fixed propagation depth (e.g., 20) as an effective trade-off between accuracy and efficiency. In contrast, traditional FedShapley approaches, which rely on retraining or static coalition sampling, lack compatibility with dynamic participation and remain difficult to scale in deployments.

Given Ripple Shapley’s natural compatibility with dynamic participation, we further explore its potential in real-time data pricing scenarios. Consider a trusted coordinator (e.g., a data marketplace or regulatory agency) orchestrating federated training across distributed data holders. By computing per-sample Ripple Shapley values, it can derive accurate, context-sensitive valuations of contributed data. In such settings, the marginal utility of data evolves over time as new clients enter and reshape the learning trajectory. For instance, suppose Client-A and Client-B hold identical datasets. If A participates early while B joins only in the final rounds, their contributions to model improvement differ significantly—yielding distinct valuations. This seemingly counterintuitive disparity reflects the temporal influence modeled by the ripple term, which captures not only how but also when a sample affects downstream updates.

Moreover, the contribution of early participants may shift

as additional clients join and reconfigure the optimization path. Ripple Shapley captures this dynamic through propagation: a sample’s value evolves as it ripples through successive updates. This enables data attribution aligned with practical valuation principles. Similar to how early market entrants often benefit from first-mover advantages under scarcity, early-stage data tends to exert a longer-lasting impact. Importantly, although Ripple Shapley accumulates over time, this property does not compromise fairness. In FL markets, incentive allocation is subject to finite budget constraints. Accordingly, final payouts are computed based on normalized contributions, ensuring equitable compensation regardless of absolute value magnitudes. Furthermore, as more data is reused or replicated across participants, its marginal utility naturally decays—a phenomenon faithfully reflected in the recursive attenuation of the ripple term.

Overall, Ripple Shapley provides not only a theoretically sound but also a practically deployable mechanism for dynamic data valuation. By capturing influence over time and across clients, it empowers FL deployments with transparent, incentive-aligned pricing strategies.

Conclusion

We proposed Ripple Shapley, a framework for real-time, in-run data attribution in FL. By decomposing sample influence into drop and ripple terms and leveraging low-rank Jacobian propagation, our method achieves accurate, real-time valuation with minimal overhead. Our results demonstrate substantial efficiency improvements and robust attribution performance under adversarial scenarios, showcase Ripple Shapley’s robustness in dynamic federated environments and its promise for enabling real-time data pricing.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under grant 62402208, 62372343, 62302184;

References

- Chen, Y.-C.; Chen, H.-W.; Wang, S.-G.; and Chen, M.-S. 2023. Space: Single-round participant amalgamation for contribution evaluation in federated learning. *Advances in Neural Information Processing Systems*, 36: 6422–6441.
- Ding, N.; Fang, Z.; and Huang, J. 2020. Incentive mechanism design for federated learning with multi-dimensional private information. In *2020 18th international symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WiOPT)*, 1–8. IEEE.
- Du, D.-Z.; and Hwang, F. K.-m. 1999. *Combinatorial group testing and its applications*, volume 12. World Scientific.
- Ghorbani, A.; Kim, M. P.; and Zou, J. 2020. A Distributional Framework For Data Valuation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 3535–3544. PMLR.
- Ghorbani, A.; and Zou, J. Y. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2242–2251. PMLR.
- Ghorbani, B.; Krishnan, S.; and Xiao, Y. 2019. An investigation into neural net optimization via Hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, 2232–2241. PMLR.
- Guo, Q.; Yao, M.; Tian, Z.; Qi, S.; Qi, Y.; Lin, Y.; and Dong, J. S. 2024. Contribution evaluation of heterogeneous participants in federated learning via prototypical representations. *arXiv preprint arXiv:2407.02073*.
- Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K. A.; Charles, Z.; Cormode, G.; Cummings, R.; D’Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Qi, H.; Ramage, D.; Raskar, R.; Raykova, M.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*, 14(1-2): 1–210.
- Lei, K.; Ren, X.; Yang, S.; Wang, X.; and Zhao, F. 2025. FedDSV: Shapley Value-Based Contribution Estimation in Federated Learning With Dynamic Participation. *IEEE Transactions on Mobile Computing*.
- Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.
- Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.
- Liu, Z.; Chen, Y.; Yu, H.; Liu, Y.; and Cui, L. 2022. Gtshapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–21.
- Lv, H.; Zheng, Z.; Luo, T.; Wu, F.; Tang, S.; Hua, L.; Jia, R.; and Lv, C. 2021. Data-free evaluation of user contributions in federated learning. In *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, 1–8. IEEE.
- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.
- Nagalapatti, L.; and Narayanam, R. 2021a. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9046–9054.
- Nagalapatti, L.; and Narayanam, R. 2021b. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9046–9054.
- Pillutla, K.; Kakade, S. M.; and Harchaoui, Z. 2022. Robust Aggregation for Federated Learning. *IEEE Transactions on Signal Processing*, 70: 1142–1154.
- Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. *Advances in neural information processing systems*, 30.
- Song, T.; Tong, Y.; and Wei, S. 2019. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, 2577–2586. IEEE.
- Sun, Q.; Li, X.; Zhang, J.; Xiong, L.; Liu, W.; Liu, J.; Qin, Z.; and Ren, K. 2023. Shapleyfl: Robust federated learning based on shapley value. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2096–2108.
- Tian, W.; Guo, J.; Xu, Z.; Li, R.; and Xiao, W. 2024. PEO-Store: Delegation-Proof Based Oblivious Storage With Secure Redundancy Elimination. *IEEE Trans. Dependable Secur. Comput.*, 21(5): 4815–4826.
- Wang, G.; Dang, C. X.; and Zhou, Z. 2019. Measure contribution of participants in federated learning. In *2019 IEEE international conference on big data (Big Data)*, 2597–2604. IEEE.
- Wang, H.; Xu, H.; Li, Y.; Xu, Y.; Li, R.; and Zhang, T. 2024a. FedCDA: Federated Learning with Cross-rounds Divergence-aware Aggregation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

- Wang, H.; Zheng, P.; Han, X.; Xu, W.; Li, R.; and Zhang, T. 2024b. FedNLR: Federated Learning with Neuron-wise Learning Rates. In Baeza-Yates, R.; and Bonchi, F., eds., *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, 3069–3080. ACM.
- Wang, J.; Tian, W.; Tang, J.; Ye, X.; Wan, Y.; Xu, Z.; and Chen, L. 2025a. Sym-CS-HFL: A secure and efficient solution for privacy-preserving heterogeneous federated learning. *Journal of Information Security and Applications*, 94: 104253.
- Wang, J.; Zhang, L.; Li, A.; You, X.; and Cheng, H. 2022. Efficient participant contribution evaluation for horizontal and vertical federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 911–923. IEEE.
- Wang, J. T.; Mittal, P.; Song, D.; and Jia, R. 2025b. Data Shapley in One Training Run. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Wang, T.; Rausch, J.; Zhang, C.; Jia, R.; and Song, D. 2020. A principled approach to data valuation for federated learning. In *Federated Learning: Privacy and Incentive*, 153–167. Springer.
- Wei, S.; Tong, Y.; Zhou, Z.; and Song, T. 2020. Efficient and fair data valuation for horizontal federated learning. In *Federated Learning: Privacy and Incentive*, 139–152. Springer.
- Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; and Yang, Q. 2020. A sustainable incentive scheme for federated learning. *IEEE Intelligent Systems*, 35(4): 58–69.
- Zhang, J.; Wu, Y.; and Pan, R. 2021. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the Web Conference 2021*, 947–956.