CONSERVATION IS NOT ALLOCATION: VALUING CLIENTS AND EDGES IN HIERARCHICAL FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Hierarchical federated learning routes client updates through edge servers, so contribution accounting must value both clients and edges. A common shortcut sums client values within each edge. For exact Owen values under a fixed partition this sum equals the edge's Shapley value in the quotient game, yet practical methods estimate client values from sampled permutations. We show that the summed ledger is then a Monte Carlo estimator of the quotient game whose efficiency residual is identically zero, so the total is always conserved while individual edges can be misallocated without any visible signal. We propose Federated Composable Contribution Evaluation (FedCCE), which gives each resolution its own estimator. Clients upload per-class feature summaries whose counts, means, and variances compose exactly under union, so the utility of any coalition of clients or edges follows from cached summaries without retraining. FedCCE samples client Owen values under a fixed budget and solves the much smaller edge quotient game exactly, and we prove that the extra utility calls this requires depend on the number of edges but not on the number of clients. Across four image benchmarks, summing budgeted client values identifies the most valuable edge in only 46% of 232 runs with 50 or 100 clients even at the largest budget, whereas FedCCE identifies it in every run and needs roughly 700× fewer utility evaluations than computing both ledgers exactly at 100 clients.

1 INTRODUCTION

Federated learning (FL) builds a shared model by aggregating updates trained locally on clients' private data, allowing multiple parties to collaborate without centralizing raw data (McMahan et al., 2017; Kairouz et al., 2021). Because these parties may contribute unequally to the resulting model, contribution accounting provides the auditable ledger needed to distribute rewards, admit participants, and assign responsibility when the model fails (Wang et al., 2020b; Liu et al., 2022; Xu et al., 2021; Chen et al., 2023; Tastan et al., 2024; Ukaye et al., 2026). The Shapley value gives a principled answer under a declared utility (Shapley, 2016; Ghorbani & Zou, 2019), and the Owen value extends it to a known coalition structure (Owen, 1977). Most of this line of work returns a single client-level ledger, even when the system itself is hierarchical (Ghazi et al., 2026). However, real deployments place a layer of edge nodes between the cloud and the participants (Fig. 1(a)), and each node is responsible for a domain of clients (Liu et al., 2020; Abad et al., 2020; Wu et al., 2024). Hierarchical incentive and reputation mechanisms already price edge servers through strategic games (Chu et al., 2023) or weight participants by approximate Shapley reputation (Yang et al., 2025), yet to our knowledge, how much an edge domain as a whole contributes to the model has not been answered in principle. Decisions about edge domains therefore rest on evidence that is either absent or would have to be assembled from client estimates, a resolution mismatch between decisions and evidence.

A natural shortcut is to compute client values and add them within each edge, since under one utility and one fixed partition the exact Owen values of the clients in an edge sum to that edge's Shapley value in the quotient game (Owen, 1977). Because coalition contexts grow exponentially with the number of clients, scalable methods instead estimate client values from finitely many sampled permutations (Castro et al., 2009; Maleki et al., 2013; Wang et al., 2020b; Lei et al., 2025), improving each query through truncation (Ghorbani & Zou, 2019), guided ordering (Liu et al., 2022), or refined

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

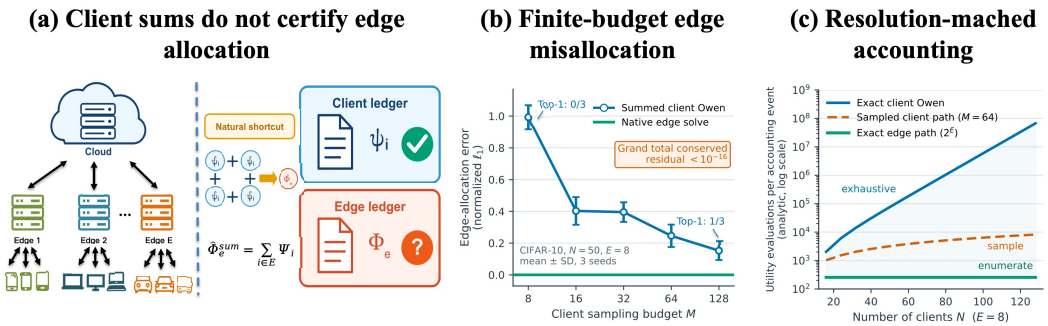

Figure 1: **Why accounting should match the deployment resolution.** **(a)** A cloud–edge deployment needs contribution records for clients and for whole edge domains, and adding client scores is only a shortcut. **(b)** Under a limited sampling budget the shortcut misallocates edge shares while the total stays exactly conserved (CIFAR-10, $N = 50$, $E = 8$, mean $\pm$ standard deviation over three seeds). **(c)** Enumerating client groups quickly becomes infeasible as the federation grows, whereas enumerating edge groups stays manageable, so FedCCE samples clients and enumerates edges.

sampling schemes (KhademSohi et al., 2025), and the identity says nothing about such estimates. Within each sampled edge order, the client marginals telescope, so the summed ledger is a Monte Carlo estimate of the quotient-game value that becomes exact only asymptotically. At any finite budget, every sampled permutation still allocates the full grand-coalition surplus, so overestimated edges are offset by underestimated ones and the efficiency residual stays zero while individual edges are misallocated. Fig. 1(b) shows this silent failure, where a sixteenfold larger client budget substantially reduces edge-allocation error yet recovers the highest-valued edge in only one of three seeds. Because the argument relies only on how marginals telescope, the failure is not tied to a particular dataset or utility. Client summation is not theoretically invalid, but a finite budget designed for a large client game is silently reused as the evidence budget for a different operational resolution. *Conservation does not certify allocation, and each resolution needs its own error control.*

A native edge ledger must evaluate unions of whole edge domains under the utility used for clients. Data valuation scores a coalition by a learner fitted to its pooled data (Ghorbani & Zou, 2019; Jia et al., 2019b). FL never pools data, so model-based utilities assemble coalition models from member updates (Wang et al., 2020b; Liu et al., 2022). Under the Non-Independent and Identically Distributed (Non-IID) real-world scenario, averaged parameters depart from the pooled-data fit by an amount bounded by the members' class skew relative to the pool (Karimireddy et al., 2020; Wang et al., 2020a), so such values also reflect how updates interact during aggregation. Two observations suggest a complementary route. First, for a learner that sees data only through class-conditional statistics (Snell et al., 2017), the utility of a union can be computed from member summaries without retraining, because the per-class counts, means, and variances of a union are determined by those of its members (Chan et al., 1983). Second, deployments usually contain far fewer edges than clients (Liu et al., 2020; Abad et al., 2020), so with such a utility the edge game is small enough to solve exactly, while the client game is estimated under a fixed budget (Fig. 1(c)).

Building on these observations, we propose Federated Composable Contribution Evaluation (FedCCE), a framework that builds client and edge ledgers from class statistics without changing federated optimization. FedCCE declares a transparent summary game on these statistics, in the line of closed-form data valuation utilities, and realizes it in four modules along the existing participant, edge, and cloud path. First, in distributed evidence preparation, clients upload per-class summaries computed under a shared encoder alongside ordinary model updates, while edge nodes keep fixed membership and relay the summaries without scoring them. Second, the cached summary utility pools member summaries into coalition summaries on demand, scores them with a variance-weighted nearest-prototype evaluator on the global validation view, and shares cached values across both solvers, so every coalition value is exact relative to the transmitted summaries. Third, resolution-matched contribution accounting estimates client Owen values with budgeted partition-respecting permutations and solves the edge quotient game by enumerating its 2^E coalitions, at a

cost that grows with the number of edges rather than clients. Finally, the independent temporal ledgers accumulate evidence across accounting rounds into separate client and edge accounts, completing absent clients from their histories while leaving inactive edges unobserved. Because both solvers query the same characteristic function, the exact targets of the two ledgers remain group-consistent while each keeps its own budget and uncertainty. The accounting takes that function as an input, so model-based utilities remain legitimate references and do not conflict with FedCCE.

In summary, we make three contributions.

1. We prove that adding up sampled client values within each edge always preserves the total yet can misallocate individual edges, so a conserved total never certifies an edge ledger. We also show that computing edge values exactly needs a number of extra utility evaluations that depends on the number of edges but not on the number of clients.
2. We propose FedCCE, in which clients share compact per-class summaries that merge exactly, so the value of any group of clients or edges is computed without retraining. FedCCE estimates client values under a fixed budget and computes edge values exactly, giving each level its own ledger without changing how the model is trained.
3. On four image benchmarks with up to 100 clients and 17 baselines, summing client values finds the most valuable edge in only 46% of 232 runs, whereas FedCCE finds it in every run with roughly 700$\times$ fewer utility evaluations than exact computation. Its exact client values also give every label-corrupted client a negative value, and removing the flagged clients raises accuracy on CIFAR-10 by 4.4 points.

2 Related Work

Contribution evaluation in federated learning. Shapley-based data valuation credits each data source with its average marginal effect on a declared utility (Shapley, 2016; Ghorbani & Zou, 2019; Jia et al., 2019b), and later work extends and analyzes this value (Ghorbani et al., 2020; Kwon et al., 2021; Kwon & Zou, 2022; Wang et al., 2024; Liu et al., 2023). Federated Shapley adapts it to the round structure of training (Wang et al., 2020b). GTG-Shapley (Liu et al., 2022) and FedOwen (KhademSohi et al., 2025) reduce the number of model-utility queries through gradient-based reconstruction and Owen sampling, and FedDSV handles clients that join and leave (Lei et al., 2025). Lightweight estimators replace retrained coalition models with cheaper signals such as gradient alignment (Xu et al., 2021), prototype amalgamation (Chen et al., 2023), class-specific gradients (Tastan et al., 2024), gradient spectral entropy (Ukaye et al., 2026), and influence propagated through a single training run (Zeng et al., 2026). These methods return client-level ledgers, and we use several of them as baselines. An account for an edge domain therefore has to be assembled from client scores. Our summary game follows closed-form data valuation utilities such as KNN-Shapley (Jia et al., 2019a), with a nearest-prototype learner (Mensink et al., 2013; Snell et al., 2017) whose class statistics, already exchanged in prototype-based federated learning (Tan et al., 2022; Dai et al., 2023), merge exactly under coalition unions (Chan et al., 1983).

Hierarchies and quotient games. Hierarchical federated learning places edge servers between the cloud and the clients (Liu et al., 2020; Abad et al., 2020; Lim et al., 2020; Wu et al., 2024; Wang et al., 2022; Yang et al., 2023). Mechanisms built on this architecture price edge servers (Chu et al., 2023), weight participants by reputation (Yang et al., 2025), or select and reward clients (Ghazi et al., 2026), each for its own control objective rather than as a counterfactual value of an edge domain. HCDV values data through a hierarchy learned by the estimator (Xiao et al., 2026), whereas our partition is fixed by the deployment. The edge value we compute is the classical Shapley value of the quotient game, which equals the sum of the exact client Owen values in that edge (Owen, 1977). Our contribution lies in how this value is obtained. Summing sampled client values fails silently because the edge errors cancel in the total, whereas FedCCE computes the value exactly under the same composable game as the client ledger.

3 Method

FedCCE adds an accounting path with four modules to standard federated training (Figure 2), described in Sections 3.2 to 3.5 after we fix the setting and the two games.

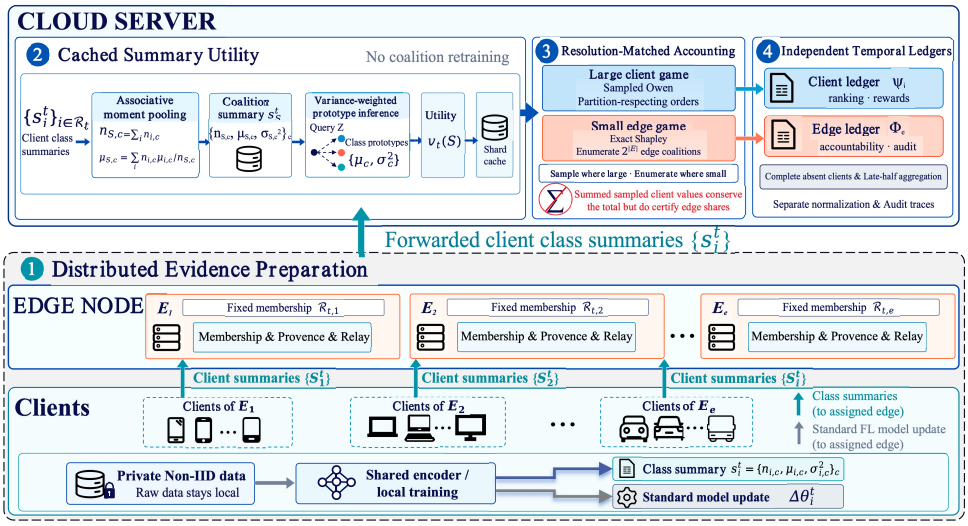

Figure 2: **FedCCE performs contribution accounting at both client and edge resolutions without changing federated training.** **(1)** Clients compute class counts, feature means, and diagonal variances from local data, while their assigned edge nodes maintain membership and provenance and relay the summaries to the cloud without scoring them. **(2)** The cloud combines client summaries to represent a requested coalition, evaluates its utility with a diagonal-variance-weighted prototype classifier, and caches the result without coalition retraining. **(3)** The shared utility cache supports sampled, partition-respecting Owen estimation for the large client game and exact Shapley enumeration for the smaller edge quotient game. **(4)** Evidence from multiple accounting rounds is aggregated into separately normalized client and edge ledgers for rewards, accountability, and audit.

3.1 Problem Setting

Federation. Let $\mathcal{N} = \{1, \dots, N\}$ be the clients and $\mathcal{E} = \{1, \dots, E\}$ the edge servers, and let client i hold private data $\mathcal{D}_i$ over C classes. The model $f_{\boldsymbol{\theta}} = p_{\boldsymbol{\theta}} \circ h_{\boldsymbol{\theta}}$ consists of a shared encoder $h_{\boldsymbol{\theta}}$ with d-dimensional output and a classifier head $p_{\boldsymbol{\theta}}$. Training proceeds in synchronous rounds $t = 1, 2, \dots$. In round t each client i in the training set $\mathcal{A}_t \subseteq \mathcal{N}$ runs local stochastic gradient steps on its cross-entropy loss from the global model $\boldsymbol{\theta}^{t-1}$ and returns the model update $\Delta\boldsymbol{\theta}_i^t$, and the server averages $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} + |\mathcal{A}_t|^{-1} \sum_{i \in \mathcal{A}_t} \Delta\boldsymbol{\theta}_i^t$.

Accounting rounds. Accounting runs in a subset $\mathcal{T}$ of rounds placed at a fixed interval, and all accounting quantities below refer to an accounting round $t \in \mathcal{T}$. In round t the deployment assigns every client to one edge, which gives the membership $\mathcal{N}_{t,e}$ of edge e, and these memberships partition $\mathcal{N}$. We write $e(i)$ for the edge of client i in round t, $\mathcal{R}_t \subseteq \mathcal{N}$ for the clients whose summaries reach the cloud, $\mathcal{R}_{t,e} = \mathcal{R}_t \cap \mathcal{N}_{t,e}$ for the reporting clients of edge e, and $\mathcal{E}_t = \{e \in \mathcal{E} : \mathcal{R}_{t,e} \neq \varnothing\}$ for the active edges, with sizes $N_t = |\mathcal{R}_t|$ and $E_t = |\mathcal{E}_t|$. The partition may change between accounting rounds, and FedCCE takes it as an input rather than estimating it. A summary costs far less than local training, so $\mathcal{R}_t$ may be larger than the training set $\mathcal{A}_t$.

Two games on one partition. Let v_t be a declared characteristic function on subsets of $\mathcal{R}_t$ with $v_t(\varnothing) = 0$. The client game is v_t with the coalition structure $\{\mathcal{R}_{t,e}\}_{e \in \mathcal{E}_t}$, and its Owen values are ψ_i^t. The quotient game treats each active edge as one player,

$$v_t^{\mathcal{E}}(U) = v_t\big(\mathcal{R}_t(U)\big), \qquad \mathcal{R}_t(U) = \bigcup_{e \in U} \mathcal{R}_{t,e}, \qquad U \subseteq \mathcal{E}_t, \tag{1}$$

and its Shapley values are Φ_e^t, collected in the vector $\Phi^t = (\Phi_e^t)_{e \in \mathcal{E}_t}$.

Lemma 1 (Group consistency). *For every accounting round t and edge $e \in \mathcal{E}_t$, $\sum_{i \in \mathcal{R}_{t,e}} \psi_i^t = \Phi_e^t$.*

Lemma 1 (Owen, 1977) links the exact targets of the two ledgers but says nothing about their finite-budget estimates, which Section 3.4 analyzes. Deployments have far fewer active edges than re-

porting clients, so FedCCE samples the large client game, solves the small edge game at its own resolution, and lets both query the same v_t.

3.2 Distributed Evidence Preparation

In accounting round t each client $i \in \mathcal{R}_t$ encodes at most three local batches with the encoder $h_{\boldsymbol{\theta}^{t-1}}$ of the current global model, before round t is aggregated. Let $\mathcal{H}_i^t$ be the resulting labeled features in $\mathbb{R}^d$. For each class c the client computes the number $n_{i,c}^t$ of features in $\mathcal{H}_i^t$ with label c, which is zero for classes it does not hold, their mean $\boldsymbol{\mu}_{i,c}^t$, and their diagonal variance $(\boldsymbol{\sigma}_{i,c}^t)^2$, floored coordinatewise at $\tau = 10^{-6}$, and forms the class summary $\boldsymbol{s}_i^t = \{n_{i,c}^t, \boldsymbol{\mu}_{i,c}^t, (\boldsymbol{\sigma}_{i,c}^t)^2\}_{c=1}^C$. The summary holds $O(Cd)$ scalars, and raw data never leave the client. A client that trains in round t sends $\boldsymbol{s}_i^t$ alongside $\Delta\boldsymbol{\theta}_i^t$, and any other client in $\mathcal{R}_t$ sends the summary alone. Each edge e keeps its membership $\mathcal{N}_{t,e}$ fixed during the round, tags every summary with its client and round for provenance, and forwards $\{\boldsymbol{s}_i^t\}_{i\in\mathcal{R}_{t,e}}$ to the cloud without scoring them.

3.3 Cached Summary Utility

For a coalition $S \subseteq \mathcal{R}_t$, the cloud pools the member summaries class by class as follows:

$$
\begin{aligned}
n_{S,c}^t &= \sum_{i\in S} n_{i,c}^t, \qquad \boldsymbol{\mu}_{S,c}^t = \frac{1}{n_{S,c}^t} \sum_{i\in S} n_{i,c}^t \boldsymbol{\mu}_{i,c}^t, \\
(\boldsymbol{\sigma}_{S,c}^t)^2 &= \frac{1}{n_{S,c}^t} \sum_{i\in S} n_{i,c}^t \big[(\boldsymbol{\sigma}_{i,c}^t)^2 + \boldsymbol{\mu}_{i,c}^t \odot \boldsymbol{\mu}_{i,c}^t\big] - \boldsymbol{\mu}_{S,c}^t \odot \boldsymbol{\mu}_{S,c}^t,
\end{aligned}
\tag{2}
$$

where $\odot$ is the elementwise product, and floors each coordinate of the pooled variance at τ. The coalition summary $\bar{\boldsymbol{s}}_S^t = \{n_{S,c}^t, \boldsymbol{\mu}_{S,c}^t, (\boldsymbol{\sigma}_{S,c}^t)^2\}_{c=1}^C$ defines a nearest-prototype classifier weighted by the diagonal variances. It assigns a validation feature $\boldsymbol{z} \in \mathbb{R}^d$ to the represented class with the smallest distance, and the coalition utility is its accuracy above chance,

$$
\mathrm{dist}_{S,c}^t(\boldsymbol{z}) = \sum_{j=1}^d \frac{(z_j - \mu_{S,c,j}^t)^2}{(\sigma_{S,c,j}^t)^2 + \varepsilon}, \qquad v_t(S) = \mathrm{Acc}_{\mathcal{V}}\Big(\arg \min_{c:\, n_{S,c}^t > 0} \mathrm{dist}_{S,c}^t\Big) - \frac{1}{C}, \tag{3}
$$

where j indexes coordinates, $\varepsilon = 10^{-8}$ guards the division, and $\mathrm{Acc}_{\mathcal{V}}$ is the accuracy on the global validation view $\mathcal{V}$, which is held out from client training data. Classes absent from S are excluded from the minimization, so their validation features count as errors, and $v_t(\varnothing) = 0$ by definition.

Proposition 1 (Composition). *For every coalition $S \subseteq \mathcal{R}_t$ and every class c with $n_{S,c}^t > 0$, Eq. 2 returns the exact count and mean of class c in $\bigcup_{i\in S} \mathcal{H}_i^t$. Its variance differs from the floored variance of that union by at most τ in every coordinate, and it is exact whenever every member variance exceeds τ. Hence $v_t(S)$ is a deterministic function of the transmitted summaries and, up to the variance floor, equals the utility of the classifier of Eq. 3 fitted to the pooled features of S.*

The characteristic function is therefore a Data Shapley utility (Ghorbani & Zou, 2019) induced by a learner whose fit composes exactly, in the line of closed-form utilities such as KNN-Shapley (Jia et al., 2019a), and no coalition is ever retrained. The cloud memoizes $v_t(S)$ for every queried coalition in a cache that both solvers share.

3.4 Resolution-Matched Accounting

Both solvers take v_t as input, so a model-based utility can replace Section 3.3 without change.

Sampled client Owen values. For each $m = 1, \ldots, M$, FedCCE draws a uniformly random order of $\mathcal{E}_t$ and, independently, a uniformly random order of the clients within each $\mathcal{R}_{t,e}$. Let $P_i^{t,m}$ contain the clients of all edges that precede $e(i)$ together with the clients that precede i within $\mathcal{R}_{t,e(i)}$. The client ledger of round t is as follows:

$$
\widehat{\psi}_i^t = \frac{1}{M} \sum_{m=1}^M \big[v_t(P_i^{t,m} \cup \{i\}) - v_t(P_i^{t,m})\big]. \tag{4}
$$

Consecutive clients of one sampled order share their prefixes, so most marginals reuse cached values.

Why summation fails silently. Summing Eq. 4 within each edge is the natural shortcut to an edge ledger. For a uniformly random order o of $\mathcal{E}_t$, let $B_e(o)$ be the set of edges that precede e in o and $X_e(o) = v_t^{\mathcal{E}}(B_e(o) \cup \{e\}) - v_t^{\mathcal{E}}(B_e(o))$, so that $\Phi_e^t = \mathbb{E}\, X_e(o)$, and let $\omega_e^2 = \operatorname{Var} X_e(o)$.

Theorem 1 (Summation is an edge-order Monte Carlo estimator)**.** *Let $\widehat{\Phi}_e^{\mathrm{sum}} = \sum_{i \in \mathcal{R}_{t,e}} \widehat{\psi}_i^t$ with $\widehat{\psi}_i^t$ computed from M independent orders, and let $\widehat{\Phi}^{\mathrm{sum}} = (\widehat{\Phi}_e^{\mathrm{sum}})_{e \in \mathcal{E}_t}$. (i) For every realization, $\widehat{\Phi}_e^{\mathrm{sum}} = \frac{1}{M} \sum_{m=1}^{M} X_e(o_m)$, where o_m is the edge order of the mth sample. (ii) For every realization and every M, $\sum_{e \in \mathcal{E}_t} \widehat{\Phi}_e^{\mathrm{sum}} = v_t^{\mathcal{E}}(\mathcal{E}_t)$, so the efficiency residual is identically zero. (iii) $\mathbb{E}\, \widehat{\Phi}_e^{\mathrm{sum}} = \Phi_e^t$, $\operatorname{Var} \widehat{\Phi}_e^{\mathrm{sum}} = \omega_e^2 / M$, and $\sqrt{M}\, \mathbb{E}\, \|\widehat{\Phi}^{\mathrm{sum}} - \Phi^t\|_1 \to \sqrt{2/\pi} \sum_{e \in \mathcal{E}_t} \omega_e$ as $M \to \infty$, where π is the circle constant.*

Theorem 1 formalizes the silent failure of Section 1. The residual that an efficiency check inspects is identically zero, while the allocation error shrinks only at rate $M^{-1/2}$, and Proposition 3 in Appendix A.1 shows that the budget to find the highest-valued edge grows as edge values get closer.

Exact edge Shapley values. FedCCE instead enumerates all 2^{E_t} coalitions of the quotient game in Eq. 1 and computes every Φ_e^t by the Shapley formula. Both solvers query v_t, so the exact targets satisfy Lemma 1, while the finite-budget errors of the two ledgers stay separate because neither ledger is derived from the other. With Φ^t in hand, the edge-level error of the summed client ledger becomes observable at no extra cost (Proposition 4 in Appendix A.1). The enumeration is cheap because sampled client orders already visit most unions of whole edges.

Proposition 2 (Cost of the exact edge ledger)**.** *With M independent orders, the expected number of distinct utility evaluations that exact enumeration adds to the client pass is*

$$\Gamma(E_t, M) = \sum_{k=1}^{E_t - 1} \binom{E_t}{k} \left(1 - \binom{E_t}{k}^{-1}\right)^M, \tag{5}$$

which does not depend on the number of clients, and decays geometrically in M.

Many edges. When 2^{E_t} evaluations exceed the budget, FedCCE samples edge orders of the quotient game directly. Spent at the budget of the client pass, this divides every variance in Theorem 1(iii) by N_t / E_t (Corollary 1 in Appendix A.1), so the edge ledger is still estimated at its own resolution.

3.5 Independent Temporal Ledgers

Each accounting round records the partition, $\mathcal{R}_t$, the signed values $\widehat{\psi}_i^t$ and Φ_e^t, the utility-call counts, and the provenance tags as an audit trace. A client outside $\mathcal{R}_t$ is completed by a persistence policy that uses its mean value over past accounting rounds in which it reported, or the mean of the reporting clients if it never reported, which gives the completed value $\widetilde{\psi}_i^t$, with $\widetilde{\psi}_i^t = \widehat{\psi}_i^t$ for $i \in \mathcal{R}_t$. An edge outside $\mathcal{E}_t$ stays unobserved rather than imputed. With $\mathcal{T}_{\mathrm{late}} \subseteq \mathcal{T}$ the pre-registered latter half of the accounting rounds, the client and edge ledgers are

$$\psi_i = \frac{1}{|\mathcal{T}_{\mathrm{late}}|} \sum_{t \in \mathcal{T}_{\mathrm{late}}} \widetilde{\psi}_i^t, \qquad \Phi_e = \frac{\sum_{t \in \mathcal{T}_{\mathrm{late}}} \mathbb{I}[e \in \mathcal{E}_t]\, \Phi_e^t}{\sum_{t \in \mathcal{T}_{\mathrm{late}}} \mathbb{I}[e \in \mathcal{E}_t]}, \tag{6}$$

where $\mathbb{I}[\cdot]$ is the indicator function and Φ_e is defined for every edge active in at least one round of $\mathcal{T}_{\mathrm{late}}$. Each ledger is normalized on its own into nonnegative weights,

$$w_i = \frac{\psi_i - \psi_{\min}}{\sum_{i' \in \mathcal{N}} (\psi_{i'} - \psi_{\min})}, \qquad W_e = \frac{\Phi_e - \Phi_{\min}}{\sum_{e'} (\Phi_{e'} - \Phi_{\min})}, \tag{7}$$

where $\psi_{\min} = \min\{0, \min_{i'} \psi_{i'}\}$, $\Phi_{\min} = \min\{0, \min_{e'} \Phi_{e'}\}$, the edge sums run over edges with a defined Φ_e, and weights are uniform when a denominator vanishes. Client weights w_i serve rankings and rewards, edge weights W_e serve accountability and are never derived from client weights, and the signed values stay in the audit trace, where negative values flag harmful participants.

Table 1: **Agreement with the declared summary-game target.** Client (C) and edge (E) Spearman correlation with the exact hierarchical Proto-Owen value (N=12, E=4, three seeds per dataset on shared training trajectories), with the mean and sample standard deviation over twelve runs in the average columns. Util. marks the queried utility, namely the summary game (P), a reconstructed-model utility (M), the rule's own gradient or prototype signal (G), or none (D). M and G rows measure agreement across utilities rather than estimation error. Cost is utility calls per accounting round under each rule's own utility. Bold and underline mark the best and second best, and †marks values equal to the reference by construction.

		CIFAR-10		CIFAR-100		F-MNIST		Tiny		Average		
Method	Util.	C	E	C	E	C	E	C	E	C	E	Cost
FedCCE (ours)	P	**.935**	1.000†	**.979**	1.000†	**.944**	1.000†	**.981**	1.000†	**.960**$_{\pm.027}$	1.000†	180
Exact flat Proto-Shapley	P	<u>.904</u>	<u>.867</u>	<u>.970</u>	**1.000**	<u>.883</u>	<u>.867</u>	<u>.958</u>	**1.000**	<u>.929</u>$_{\pm.052}$	<u>.933</u>$_{\pm.098}$	4,096
Flat Proto-TMC	P	.844	.733	.953	**1.000**	.865	<u>.867</u>	.956	**1.000**	.904$_{\pm.081}$	.900$_{\pm.160}$	520
HCDV-native	P	.825	.800	.862	**1.000**	.774	**.933**	.855	.667	.829$_{\pm.120}$	.850$_{\pm.228}$	76
HCDV-edge	P	.487	**.933**	.424	<u>.933</u>	.653	**.933**	.231	**1.000**	.449$_{\pm.286}$	**.950**$_{\pm.090}$	48
Hier. TMC-KMeans	P	.669	.600	.510	**1.000**	.682	**.933**	.394	**1.000**	.564$_{\pm.247}$	.883$_{\pm.199}$	48
Hier. TMC-Random	P	.828	.667	.869	.867	.744	.467	.797	<u>.933</u>	.809$_{\pm.111}$	.733$_{\pm.345}$	48
Model-TMC	M	.522	.667	.427	.800	.075	−.333	.189	.467	.303$_{\pm.241}$	.400$_{\pm.578}$	746
GTG-Shapley	M	.455	.800	.436	.600	.121	.467	−.021	.467	.248$_{\pm.331}$	.583$_{\pm.335}$	57
FedOwen	M	.396	.733	.345	.400	−.016	.133	.123	−.133	.212$_{\pm.274}$	.283$_{\pm.439}$	153
FedDSV	M	.350	.667	.375	.600	.005	−.200	−.028	.200	.175$_{\pm.354}$	.317$_{\pm.569}$	850
DFCE	G	.772	.600	.443	.400	.408	.067	−.079	.267	.386$_{\pm.382}$	.333$_{\pm.535}$	2
ShapFed	G	.252	.067	.228	−.133	.406	.733	−.084	.333	.200$_{\pm.351}$	.250$_{\pm.616}$	2
SPACE	G	−.483	−.333	−.114	.267	−.555	−.267	−.100	−.133	−.313$_{\pm.273}$	−.117$_{\pm.618}$	531
DataSize	D	.298	.067	.422	.400	.403	.800	−.221	.333	.226$_{\pm.317}$	.400$_{\pm.435}$	0
Label entropy	D	.776	.400	.284	.467	.676	.467	−.156	−.200	.395$_{\pm.442}$	.283$_{\pm.575}$	0
Label coverage	D	.520	.650	.099	.183	.463	.723	−.126	−.095	.239$_{\pm.417}$	.365$_{\pm.550}$	0
Majority balance	D	.636	.600	.406	.533	.541	.533	−.212	.067	.343$_{\pm.416}$	.433$_{\pm.450}$	0

4 Experiments

Following Section 1, we show that FedCCE computes the declared value at both resolutions, that summing client values misallocates edges while its total stays exact, that the ledgers explain and repair harmful participation, and that the summary game makes both ledgers cheap.

4.1 Setup

We evaluate on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), Fashion-MNIST (Xiao et al., 2017), and Tiny-ImageNet (Le et al., 2015) with ResNet-20 (He et al., 2016) on the color datasets and LeNet (LeCun et al., 1998) on Fashion-MNIST, using Dirichlet label partitions (Hsu et al., 2019) with $\alpha = 0.1$ unless stated otherwise, and 40 rounds of FedAvg (McMahan et al., 2017) shared by all methods, so accounting never changes the model. A small deployment with 12 clients and four edges keeps every exact reference tractable, and deployment-scale federations with 50 clients and eight edges or 100 clients and ten edges train one-tenth of the clients per round while every client uploads a class summary in each accounting round. Partitions are fixed before accounting, utilities use a validation view held out from client training data, and the official test split is used only for final accuracy. We compare FedCCE with the 17 baselines of Table 1, covering summary-game estimators including client-level adaptations of HCDV (Xiao et al., 2026), published model-utility estimators (Ghorbani & Zou, 2019; Liu et al., 2022; KhademSohi et al., 2025; Lei et al., 2025), signal-based estimators (Ukaye et al., 2026; Tastan et al., 2024; Chen et al., 2023), and data proxies. Methods without a native edge ledger sum client scores within each edge (Appendix A.2).

4.2 Main Results

Following the standard test for Shapley estimators, Table 1 compares every accounting rule with the exact hierarchical Proto-Owen value of the declared game. FedCCE has the highest client correlation on all four datasets, averaging 0.960 with 180 utility calls per accounting round, and its edge ledger

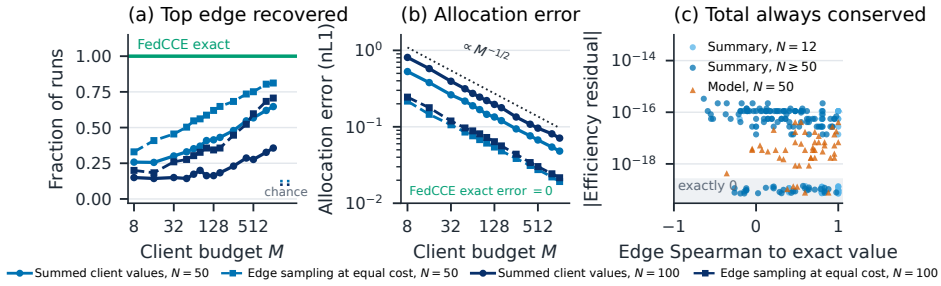

Figure 3: **Summation audit at deployment scale** (20 deployments with N=50 and 6 with N=100, 50 sampling streams each). Solid lines sum budgeted client Owen values, and dashed lines sample the quotient game at the same number of marginal evaluations. (a) Runs recovering the highest-valued edge, with the exact FedCCE edge ledger on top and chance $1/E$ dotted. (b) Normalized allocation error with the $M^{-1/2}$ rate of Theorem 1. (c) Efficiency residual of every summed ledger against its edge Spearman correlation, with exact zeros shaded.

matches the exact value in every run. Exact flat Proto-Shapley spends 23 times more calls and still reaches only 0.929 because the flat value ignores the partition, while HCDV and the hierarchical TMC controls fall at least 0.13 behind. Published estimators that query model, gradient, or prototype utilities agree with the declared value at 0.386 or less, and data proxies reach at most 0.395, so the declared game captures contribution beyond data volume and label statistics.

4.3 Does Summing Client Values Recover Edge Shares?

Figure 3 audits the common shortcut of summing budgeted client Owen values within each edge. With 50 clients and eight edges the summed ledger recovers the highest-valued edge in 41 percent of runs at $M = 128$ and in only 65 percent even at $M = 1024$, and with 100 clients and ten edges in 16 and 36 percent, while its efficiency residual stays below 3.3×10^{-16} whether the ranking is right or reversed (Figure 3c). Its error falls only at the $M^{-1/2}$ rate of Theorem 1, with a fitted exponent of -0.49 over 294 run-budget pairs (Appendix Table A7). Across 232 runs with 50 or 100 clients, four roster designs, and clean or corrupted data, summation finds the leading edge in only 46 percent of runs at the largest budget (Appendix A.7), and the same failure appears under a model utility, where summed Model-Owen values find it in 28 of 48 cases against the exact quotient value under the same model utility. Resolution matching removes the problem. Sampling the quotient game at the same cost cuts the allocation error by a factor of about 2.5 with 50 clients and 3 with 100 at every budget, and FedCCE, which enumerates the quotient game, recovers every leading edge for at most 173 extra utility calls with eight edges and 904 with ten.

4.4 Downstream Applications

Figure 4a and b turn the ledgers into decisions, with ground truth from injected label corruption. The sign of an exact client value explains a flag, since across CIFAR-10 and four Fashion-MNIST placements all 33 corrupted clients receive negative values and all 147 benign clients positive ones, and the corrupted signs persist under a Brier utility (Appendix Table A12). Acting on the ledger repairs the model. Removing the two clients flagged by FedCCE from the same checkpoint removes both corrupted clients in every seed and raises final accuracy by 4.35 ± 1.44 points over no removal, the largest gain among all methods, whereas the hierarchical control removes a benign client and DataSize and random removal never remove a corrupted one. At deployment scale FedCCE also ranks hidden corrupted clients above a random scorer on every dataset and beats Model-Owen, Model-TMC, and GTG-Shapley in at least ten of twelve dataset-seed clusters (Appendix A.8).

4.5 What Do the Two Ledgers Cost?

Figure 5 times both utilities on the same GPU and coalitions. A summary query costs 0.31 to 0.98 milliseconds, 42, 32, 3.4, and 71 times less than a model query on the four datasets at any number

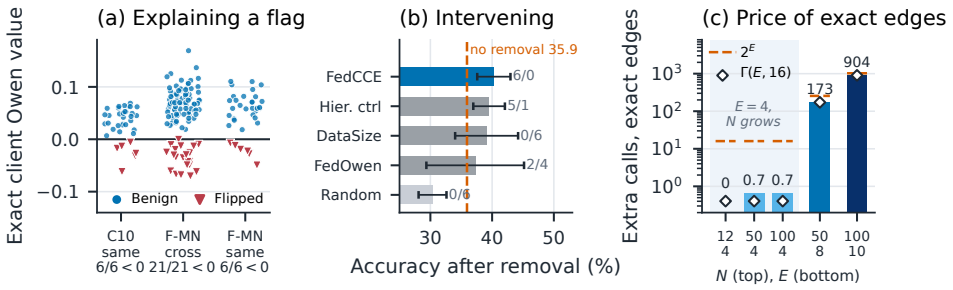

Figure 4: **Using the ledgers and their price.** (a) Exact client Owen values with fully label-flipped clients on CIFAR-10 (C10) and Fashion-MNIST (F-MN), placed in one edge or across edges (N=12, E=4, three seeds). (b) CIFAR-10 accuracy after each method removes two clients from the same checkpoint, with mean and standard deviation over three seeds and the numbers of corrupted and benign clients removed. (c) Extra calls of the exact edge ledger beyond client sampling (M=16, three seeds), with 2^E dashed and Proposition 2 as diamonds.

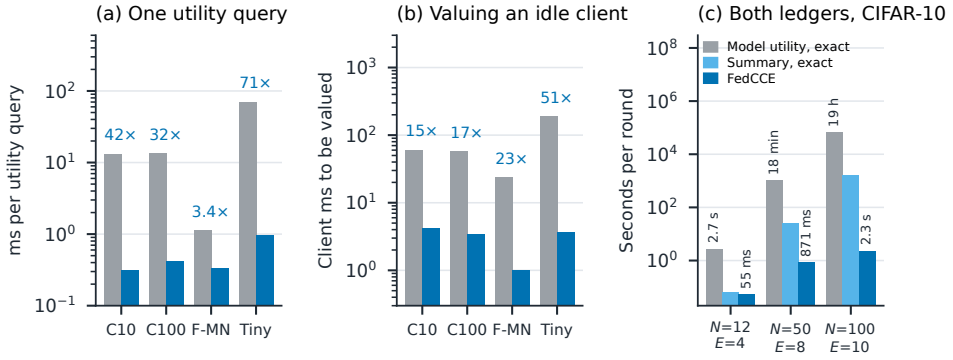

Figure 5: **What the summary game saves** (one GPU, three seeds, model utility in gray and FedCCE in blue). (a) Time of one utility query on the same coalitions (N=12). (b) Client compute to be valued in a round at N=50, a local update against at most three forward batches. (c) Time per accounting round on CIFAR-10, call counts times the query costs of (a).

of clients, a client that did not train is valued with 15 to 51 times less compute from at most three forward batches, and its summary is 0.47% of a model update on CIFAR-10. With 100 clients both ledgers take 2.3 seconds with FedCCE against 19 hours for exact enumeration under a model utility (Appendix A.13). The exact edge ledger adds fewer than one, 173, and 904 calls with four, eight, and ten edges, matching Proposition 2 whatever the number of clients (Figure 4c).

5 Conclusion

Hierarchical federated learning needs contribution accounting at the resolution where decisions are made, and summing sampled client values within each edge cannot provide it, since the summed ledger conserves its total exactly at every budget while misallocating individual edges. FedCCE matches the estimator to the resolution. Class summaries compose exactly under union, so one cached utility scores any coalition of clients or edges without retraining, client Owen values are sampled under a fixed budget, and the small edge quotient game is solved exactly at a cost that grows with the number of edges rather than clients. FedCCE identifies the most valuable edge in every run, whereas summation finds it in only 46 percent of 232 deployment-scale runs, needs roughly 700 times fewer utility evaluations than exact accounting at 100 clients, and yields ledgers that expose every label-corrupted client and improve accuracy when acted on.

AI USE STATEMENT

We used large language model assistants in this work. They helped polish and refine the text of the paper, and check the consistency of notation and proofs. They did not generate experimental data. The authors reviewed every AI-assisted passage, script and derivation, reran the analysis scripts against the recorded logs, and take full responsibility for its content, including text, claims and artifacts.

REFERENCES

Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. Hierarchical federated learning across heterogeneous cellular networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8866–8870. IEEE, 2020.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & operations research*, 36(5):1726–1730, 2009.

Tony F Chan, Gene H Golub, and Randall J LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247, 1983.

Yi-Chung Chen, Hsi-Wen Chen, Shun-Gui Wang, and Ming-Syan Chen. SPACE: single-round participant amalgamation for contribution evaluation in federated learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/14a812fa4b6bf244d055e37a7cd2f557-Abstract-Conference.html.

Shunfeng Chu, Jun Li, Kang Wei, Yuwen Qian, Kunlun Wang, Feng Shu, and Wen Chen. Design of two-level incentive mechanisms for hierarchical federated learning. *arXiv preprint arXiv:2304.04162*, 2023.

Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. Tackling data heterogeneity in federated learning with class prototypes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7314–7322, 2023.

Seyed Salar Ghazi, Kaiwen Zhang, Hans-Arno Jacobsen, et al. Scope-fl: A strategy-proof chain-based optimal pareto efficient federated learning system. *arXiv preprint arXiv:2606.18384*, 2026.

Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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*, pp. 2242–2251. PMLR, 2019. URL http://proceedings.mlr.press/v97/ghorbani19c.html.

Amirata Ghorbani, Michael P. Kim, and James Zou. 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*, pp. 3535–3544. PMLR, 2020. URL http://proceedings.mlr.press/v119/ghorbani20a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019a.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1167–1176. PMLR, 2019b. URL http://proceedings.mlr.press/v89/jia19a.html.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. 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*, pp. 5132–5143. PMLR, 2020. URL http://proceedings.mlr.press/v119/karimireddy20a.html.

Hossein KhademSohi, Hadi Hemmati, Jiayu Zhou, and Steve Drew. Owen sampling accelerates contribution estimation in federated learning. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2778–2785. IOS Press, 2025. doi: 10.3233/FAIA251133. URL https://doi.org/10.3233/FAIA251133.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 2022. URL https://proceedings.mlr.press/v151/kwon22a.html.

Yongchan Kwon, Manuel A. Rivas, and James Zou. Efficient computation and analysis of distributional shapley values. In Arindam Banerjee and Kenji Fukumizu (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 793–801. PMLR, 2021. URL http://proceedings.mlr.press/v130/kwon21a.html.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kaijia Lei, Xuebin Ren, Shusen Yang, Xiaocheng Wang, and Fangyuan Zhao. Feddsv: Shapley value-based contribution estimation in federated learning with dynamic participation. *IEEE Trans. Mob. Comput.*, 24(10):11019–11033, 2025. doi: 10.1109/TMC.2025.3574784. URL https://doi.org/10.1109/TMC.2025.3574784.

Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A

comprehensive survey. *IEEE Commun. Surv. Tutorials*, 22(3):2031–2063, 2020. doi: 10.1109/COMST.2020.2986024. URL https://doi.org/10.1109/COMST.2020.2986024.

Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pp. 1–6. IEEE, 2020. doi: 10.1109/ICC40277.2020.9148862. URL https://doi.org/10.1109/ICC40277.2020.9148862.

Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Trans. Intell. Syst. Technol.*, 13(4): 60:1–60:21, 2022. doi: 10.1145/3501811. URL https://doi.org/10.1145/3501811.

Zhihong Liu, Hoang Anh Just, Xiangyu Chang, Xi Chen, and Ruoxi Jia. 2d-shapley: A framework for fragmented data valuation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21730–21755. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23s.html.

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

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

Guilliermo Owen. Values of games with a priori unions. In *Mathematical economics and game theory: Essays in honor of Oskar Morgenstern*, pp. 76–88. Springer, 1977.

LS Shapley. 17. a value for n-person games. In *Contributions to the Theory of Games, Volume II*, pp. 307–318. Princeton University Press, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8432–8440, 2022.

Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horváth, and Karthik Nandakumar. Redefining contributions: Shapley-driven federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 5009–5017. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/554.

Asim Ukaye, Mubarak Abdu-Aguye, Nurbek Tastan, and Karthik Nandakumar. Data-free contribution estimation in federated learning using gradient von neumann entropy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3369–3378, 2026.

Jiachen T. Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. Rethinking data shapley for data selection tasks: Misleads and merits. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52033–52063. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/wang24cg.html.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/564127c03caab942e503ee6f810f54fd-Abstract.html.

Jiayi Wang, Shiqiang Wang, Rong-Rong Chen, and Mingyue Ji. Demystifying why local aggregation helps: Convergence analysis of hierarchical SGD. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 8548–8556. AAAI Press, 2022. doi: 10.1609/AAAI.V36I8.20832. URL https://doi.org/10.1609/aaai.v36i8.20832.

Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. In Qiang Yang, Lixin Fan, and Han Yu (eds.), *Federated Learning - Privacy and Incentive*, volume 12500 of *Lecture Notes in Computer Science*, pp. 153–167. Springer, 2020b. doi: 10.1007/978-3-030-63076-8_11. URL https://doi.org/10.1007/978-3-030-63076-8_11.

Jiajun Wu, Fan Dong, Henry Leung, Zhuangdi Zhu, Jiayu Zhou, and Steve Drew. Topology-aware federated learning in edge computing: A comprehensive survey. *ACM Comput. Surv.*, 56(10):262, 2024. doi: 10.1145/3659205. URL https://doi.org/10.1145/3659205.

Canran Xiao, Jiabao Dou, Zhiming Lin, Zong Ke, and Liwei Hou. From points to coalitions: Hierarchical contrastive shapley values for prioritizing data samples. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 15995–16003. AAAI Press, 2026. doi: 10.1609/AAAI.V40I19.38633. URL https://doi.org/10.1609/aaai.v40i19.38633.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 16104–16117, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8682cc30db9c025ecd3fee433f8ab54c-Abstract.html.

Jixiao Yang, Jinyu Chen, Zixiao Huang, Chengda Xu, Chi Zhang, and Sijia Li. Cost-trustfl: Cost-aware hierarchical federated learning with lightweight reputation evaluation across multi-cloud. In *Proceedings of the 2025 3rd International Conference on Artificial Intelligence, Systems and Network Security*, pp. 450–455, 2025.

Zhengjie Yang, Sen Fu, Wei Bao, Dong Yuan, and Albert Y. Zomaya. Hierarchical federated learning with momentum acceleration in multi-tier networks. *IEEE Trans. Parallel Distributed Syst.*, 34 (10):2629–2641, 2023. doi: 10.1109/TPDS.2023.3294688. URL https://doi.org/10.1109/TPDS.2023.3294688.

Dewen Zeng, Wenlong Tian, Haozhao Wang, Jianfeng Lu, Weijun Xiao, and Zhiyong Xu. Ripple shapley: Data influence attribution in one federated training run. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 28085–28093. AAAI Press, 2026. doi: 10.1609/AAAI.V40I33.40034. URL https://doi.org/10.1609/aaai.v40i33.40034.

A Appendix

This appendix supports Sections 3 and 4 in their order. Appendix A.1 proves the results of Section 3 and states three additional results used there. Appendix A.2 completes the setup of Section 4.1, Appendices A.3 to A.6 support the agreement results of Section 4.2, Appendix A.7 extends the summation audit of Section 4.3, Appendices A.8 to A.11 extend the downstream results of Section 4.4, Appendix A.12 reports the temporal ledger of Section 3.5 together with wall-clock availability, Appendix A.13 details the wall-clock benchmark of Section 4.5, and Appendix A.14 discusses limitations and future work.

A.1 Proofs and Additional Results

Throughout this subsection we fix an accounting round t and use the notation of Section 3. We additionally write $L = \max_{e\in\mathcal{E}_t,\, o} |X_e(o)|$ for the largest edge marginal, where the maximum runs over all orders o of $\mathcal{E}_t$.

Proof of Proposition 1. Class counts and first moments are additive over disjoint feature sets, so the pooled count and mean are exact. Write the floored member variance as $(\boldsymbol{\sigma}^t_{i,c})^2 = \boldsymbol{\varsigma}^2_{i,c} + \boldsymbol{\eta}_{i,c}$, where $\boldsymbol{\varsigma}^2_{i,c}$ is the unfloored variance of class c in $\mathcal{H}^t_i$ and $0 \le \boldsymbol{\eta}_{i,c} \le \tau$ coordinatewise. The pooled second moment then exceeds that of the union by the count-weighted average of the $\boldsymbol{\eta}_{i,c}$, which lies in $[0, \tau]$, so before its own floor the pooled variance lies between the union variance and the union variance plus τ. Flooring both quantities at τ keeps their difference within τ. If every member variance exceeds τ, all $\boldsymbol{\eta}_{i,c}$ vanish, and the union variance is at least the count-weighted average of the member variances and hence exceeds τ, so no floor is applied and the variance is exact. The classifier of Equation equation 3 depends on its features only through these class moments, which gives the last claim.

Proof of Theorem 1. (i) Fix a sample with edge order o and, for edge e, the within-edge order $i_1, \dots, i_q$ of $\mathcal{R}_{t,e}$ with $q = |\mathcal{R}_{t,e}|$. Let $S_0 = \mathcal{R}_t(B_e(o))$ and $S_k = S_0 \cup \{i_1, \dots, i_k\}$ for $k = 1, \dots, q$. The marginals of these clients are $v_t(S_k) - v_t(S_{k-1})$, whose sum telescopes to $v_t(S_0 \cup \mathcal{R}_{t,e}) - v_t(S_0) = X_e(o)$. Averaging over the M samples gives the claim. (ii) Within one order the edge marginals $X_e(o)$ telescope along o to $v^{\mathcal{E}}_t(\mathcal{E}_t) - v^{\mathcal{E}}_t(\varnothing) = v^{\mathcal{E}}_t(\mathcal{E}_t)$, and averaging preserves the identity. (iii) The edge order of each sample is uniform over the $E_t!$ orders, and the permutation form of the Shapley value gives $\mathbb{E}\, X_e(o) = \Phi^t_e$. Samples are independent, which gives the variance. The $X_e(o_m)$ are bounded by L, so the central limit theorem and uniform integrability give $\sqrt{M}\,\mathbb{E}|\widehat{\Phi}^{\mathrm{sum}}_e - \Phi^t_e| \to \omega_e\sqrt{2/\pi}$, and summing over the E_t edges gives the stated limit.

Proposition 3 (Identifying the highest-valued edge)**.** *If the highest edge value exceeds every other by at least $\kappa > 0$, the summed ledger misidentifies the highest-valued edge with probability at most $(E_t - 1)\exp\big(-M\kappa^2/(8L^2)\big)$. For any $\delta \in (0,1)$ it therefore identifies that edge with probability at least $1-\delta$ once $M \ge 8L^2\kappa^{-2}\log\big((E_t-1)/\delta\big)$, whereas exact enumeration needs 2^{E_t} evaluations regardless of κ.*

Proof. Let $e^\star$ be the highest-valued edge and $e' \ne e^\star$. By Theorem 1(i) the difference $\widehat{\Phi}^{\mathrm{sum}}_{e^\star} - \widehat{\Phi}^{\mathrm{sum}}_{e'}$ is the average of the independent terms $X_{e^\star}(o_m) - X_{e'}(o_m)$, which lie in $[-2L, 2L]$ and have mean $\Phi^t_{e^\star} - \Phi^t_{e'} \ge \kappa$. Hoeffding's inequality bounds the probability that this average is not positive by $\exp(-M\kappa^2/(8L^2))$, and a union bound over the $E_t - 1$ edges e' gives the first claim. Setting the bound equal to δ gives the second.

Proposition 4 (Observable error)**.** *Let $\xi_e = \widehat{\Phi}^{\mathrm{sum}}_e - \Phi^t_e$ for $e \in \mathcal{E}_t$. Then ξ_e is exactly the edge-level error of the summed client ledger, $\sum_{e\in\mathcal{E}_t}\xi_e = 0$, $\mathbb{E}\,\xi_e = 0$, and $\mathrm{Var}\,\xi_e = \omega_e^2/M$.*

Proof. By Lemma 1 the exact client Owen values of edge e sum to Φ^t_e, so ξ_e is the difference between the summed estimate and its exact target, which is its error. Summing over edges and applying Theorem 1(ii) gives $\sum_e \xi_e = v^{\mathcal{E}}_t(\mathcal{E}_t) - \sum_e \Phi^t_e = 0$ by efficiency of the Shapley value. The mean and variance follow from Theorem 1(iii). Once FedCCE has computed Φ^t, every ξ_e is available without further utility evaluations, so an operator may report it or raise M until $\max_e|\xi_e|$ falls below a tolerance. Theorem 1 holds for a fixed M, so such stopping is a budget-selection rule rather than part of the estimator.

Proof of Proposition 2. In each sample the client pass evaluates v_t at the union of the first k edges of o for every k, since this union is the base of the $(k{+}1)$th edge and the completed coalition after the kth edge. The empty union costs nothing and the full union is always visited, so only unions of $1 \le k \le E_t - 1$ edges can be missing. A fixed set of k edges is the prefix of a uniformly random order with probability $k!(E_t - k)!/E_t! = \binom{E_t}{k}^{-1}$, independently across samples, so it is missed by all M samples with probability $\left(1 - \binom{E_t}{k}^{-1}\right)^M$. Linearity of expectation over the $\binom{E_t}{k}$ sets of each size gives Equation equation 5. No term depends on the number of clients. Each missing probability is at most one, so the sum never exceeds $\sum_{k=1}^{E_t-1} \binom{E_t}{k} = 2^{E_t} - 2$, and each term decays geometrically in M.

Corollary 1 (Resolution matching at equal cost)**.** *Spending the MN_t marginal evaluations of the client pass on the quotient game directly yields MN_t/E_t independent edge orders and divides every variance in Theorem 1(iii) by N_t/E_t.*

Proof. One edge order of the quotient game costs E_t marginal evaluations, so the budget buys MN_t/E_t independent orders of the estimator in Theorem 1(i), whose variance is $\omega_e^2 E_t/(MN_t)$.

A.2 Implementation and Baselines

The independent unit of every statistical comparison is a dataset-seed pair. The 17 baselines are exact and sampled flat Proto-Shapley, two hierarchical TMC controls, two client-level adaptations of HCDV (Xiao et al., 2026), Model-TMC (Ghorbani & Zou, 2019), GTG-Shapley (Liu et al., 2022), FedOwen (KhademSohi et al., 2025), FedDSV (Lei et al., 2025), DFCE (Ukaye et al., 2026), ShapFed (Tastan et al., 2024), SPACE (Chen et al., 2023), DataSize, and three label statistics. All accounting methods read the same cached class summaries, validation embeddings, and partitions in each accounting round, so differences come from the accounting rule alone. Summary-game baselines share the FedCCE utility cache. HCDV-native and HCDV-edge are our client-level adaptations of HCDV on the fixed deployment partition rather than official reproductions, because HCDV learns its own hierarchy. FedOwen uses the released scoring component of the official code, which computes flat Shapley scores without the client-selection loop of the original method, and the same component is used for the intervention case study and for Adult. Hierarchical TMC-Random and TMC-KMeans sample two-level permutations over random or learned groups. Model-TMC and budgeted Model-Owen apply truncated Monte Carlo sampling and the FedCCE client sampler to a reconstructed-model utility, so the latter solves the model quotient game with the same machinery as FedCCE. Model-TMC, GTG-Shapley, FedOwen, FedDSV, DFCE, ShapFed, and SPACE in Table 1 run on the same training trajectories as the other rows, and their exact Proto-Owen references coincide with those of the summary-game rows to within 2×10^{-8}. At $N \ge 50$ the deployment roster is drawn from the seed before any data are allocated and split into equal edges, and at $N = 12$ edges come from a balanced k-means on client label histograms that is fixed before accounting. In both cases the partition is an input to every method. The data proxies of Table 1 score a client by its number of training samples, the normalized entropy of its label distribution, the number of classes it holds, or one minus its largest class share, and score an edge by the sum over its clients. An audit of all 160 cached deployments confirms that every utility is computed on a validation view drawn from the training split, that the official test split is never used for scoring or selection, that deployment-scale rosters are fixed before data allocation, and that training uses the same uniform aggregation for every method. In each accounting round every client computes its class summary with the global encoder available before that round's aggregation on at most three local batches, even when it did not train in that round, and client and pooled variances are floored at 10^{-6}. A class summary stores per-class counts, feature means, and diagonal variances of the shared encoder output, which is 5,160 bytes per client on CIFAR-10 and 20,520 bytes on Fashion-MNIST.

A.3 Fidelity of the Client Ledger against Cost

Figure A1 places the methods of Table 1 on a cost axis and follows the FedCCE client ledger at $N = 50$ as the budget grows. Figure A2 gives the budget curves for each dataset.

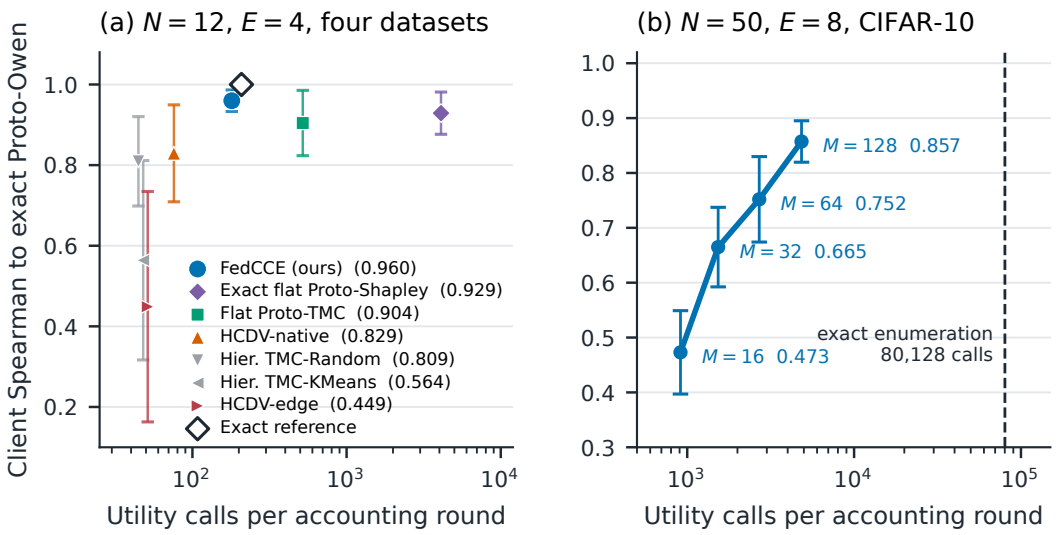

Figure A1: **Fidelity against cost.** (a) Mean client Spearman correlation with the exact hierarchical Proto-Owen value over twelve runs (four datasets, three seeds) against utility calls per accounting round, with sample standard deviation. The legend gives each mean. (b) Client Spearman correlation of FedCCE with a sampled full-Owen reference on CIFAR-10 with $N = 50$ and $E = 8$ as the permutation budget M grows, three seeds. The dashed line marks the calls needed for exact hierarchical enumeration.

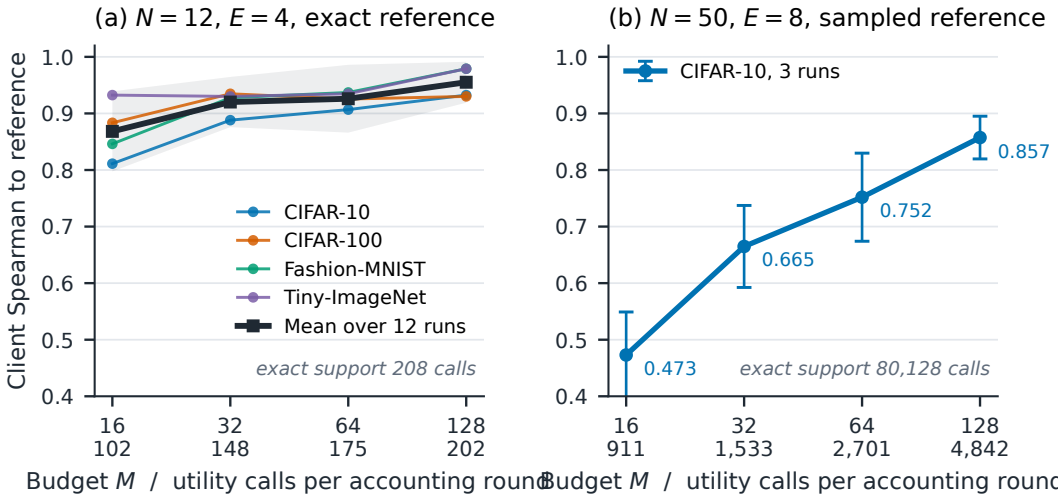

Figure A2: **Fidelity of the budgeted client ledger to the declared summary game.** (a) $N = 12$, $E = 4$, four datasets and three seeds, against exact Proto-Owen, with the budget M and the utility calls per accounting round on the tick labels. (b) $N = 50$, $E = 8$, CIFAR-10, three seeds, against a sampled full-Owen reference with 1,024 permutations repeated five times. Exact client enumeration would need 80,128 calls per accounting round at this size.

A.4 Consistency of the Two Ledgers

Exact targets are consistent by construction. On CIFAR-10 with two corrupted clients, the exact client Owen values of each edge sum to its exact quotient value within 3×10^{-17} in all three seeds. At a finite budget FedCCE does not force this identity. The observable error ξ_e of Proposition 4

reaches at most 2.7% of the total edge mass across the three seeds, so the finite-budget error of the client ledger stays visible instead of being absorbed into the edge ledger.

A.5 Agreement with Model Attribution

Table A1 scores the edge ledger of each model-utility rule against the exact Model-Owen value of its own game and places the summary-game rules beside them, and Table A10 gives the detection AUPRC of the same methods. The two games measure related but distinct quantities. At the edge level FedCCE agrees with the exact Model-Owen value at 0.550, more than twice the 0.250 obtained by summing the same client values. Table A2 locates the difference between the two games on Adult, where a closed-form ridge learner makes exact retraining references available for both. There the exact Proto-Owen value itself correlates with the exact Model-Owen value at 0.662, and FedCCE reaches 0.642, so the difference comes from the declared game rather than from the FedCCE estimator.

Table A1: **Edge agreement with the model game, by dataset.** Edge Spearman correlation with the exact checkpoint-conditional Model-Owen value (N=12, E=4, three seeds per dataset, 900-second budget for every rule). Model-utility rules (M) are scored against the exact value of their own game, and summary-game rules (P) appear as a cross-utility comparison. The average column gives the mean and sample standard deviation over twelve runs. Bold and underline mark the best and second best. †Budgeted Model-Owen solves the quotient game that defines the reference, so its edge value is 1 by construction and is not ranked. ‡Same client ledger as FedCCE with edge values obtained by summation.

Method	Util.	CIFAR-10	CIFAR-100	F-MNIST	Tiny	Average
Exact Model-Owen (ref.)	M	1.000	1.000	1.000	1.000	1.000
Model-Owen, budgeted	M	1.000†	1.000†	1.000†	1.000†	1.000†
Model-TMC	M	.200	.000	**.267**	.467	.233±.718
FedOwen	M	<u>.600</u>	.400	**.267**	.200	.367±.503
GTG-Shapley	M	.133	<u>.467</u>	−.267	.133	.117±.501
FedCCE (ours)	P	**.733**	**.600**	<u>.200</u>	<u>.667</u>	**.550**±.444
FedCCE, edge sum‡	P	.267	.400	−.200	.533	.250±.693
Flat Proto-TMC	P	.533	**.600**	.067	**.733**	<u>.483</u>±.395

Table A2: **Agreement with exact retraining-based references on Adult** ($N = 12$, $E = 4$, $\alpha = 0.5$, closed-form ridge, five seeds). Contributions are computed on a stratified validation half of the Adult test file, and all downstream curves use the disjoint final-test half. Client Spearman correlation.

Estimator	Utility and structure	Reference	Spearman
Exact Proto-Owen	Summary game, exact	exact Model-Owen	.662±.118
FedCCE	Summary game, sampled	exact Model-Owen	.642±.129
FedOwen (official)	Model, flat sampled	exact flat Model-SV	.771±.144
FedOwen (official)	Model, flat sampled	exact Model-Owen	.740±.093
Model LOO	Model, leave-one-out	exact Model-Owen	−.263±.299
DataSize	Size proxy	exact Model-Owen	.345±.259

A.6 Edge Ledgers against an Independent Model Reference

Figure A3 keeps the FedCCE client ledger fixed and changes only how its edge ledger is formed. The direct ledger is exact for the summary game and not for the model game, so this is a cross-utility test rather than a solver test. The exact edge ledger agrees better than the summed ledger in all six untied runs at $E = 4$ (exact sign test $p = 0.031$), and the two are close at $E = 8$, where the two games themselves agree less.

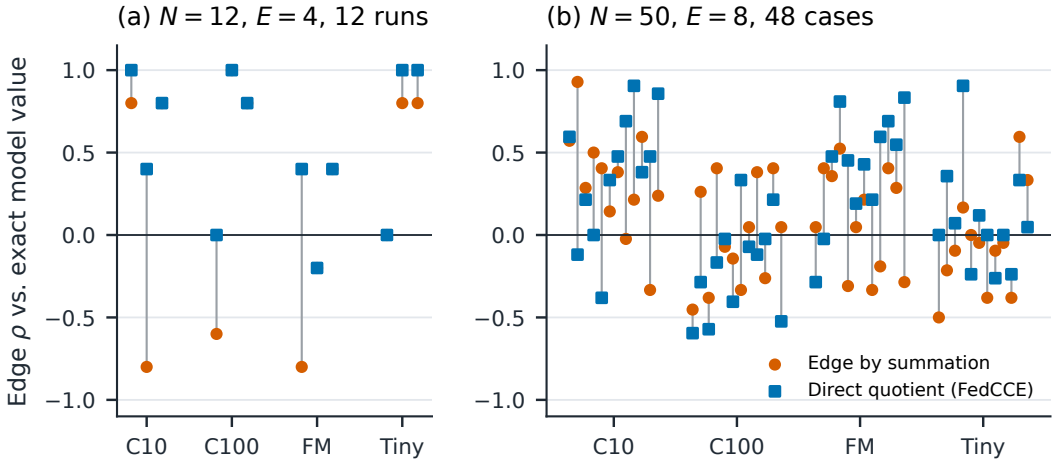

Figure A3: **Edge ledgers against an independent model reference, run by run.** Each vertical segment joins the edge Spearman correlation of the summed ledger (circle) and of the direct ledger (square) for the same FedCCE client ledger. (a) $N = 12$, $E = 4$, twelve runs. (b) $N = 50$, $E = 8$, 48 detection cases.

A.7 Summation Audit Details

Table A3 gives the full audit at three scales, including flat Shapley sums, the small deployment, and the model-utility block. Table A4 splits the model-utility audit by dataset and heterogeneity, and Table A5 reports every dataset and budget of the summary-utility audit. The remaining tables extend the audit along four axes. Table A6 measures direct edge sampling at equal cost and checks the estimator identity of Theorem 1(i) against the real client-level sums. Table A7 varies the number of clients and edges separately. Table A8 changes how clients are assigned to edges, and Table A9 varies heterogeneity under hidden label corruption. Pooled over the 232 runs with 50 or 100 clients in these tables and in Table A3, summation recovers the leading edge in 28, 32, and 46 percent of runs at $M = 16$, 64, and 128, with mean allocation errors of 0.37, 0.20, and 0.13, while the efficiency residual never exceeds 5×10^{-16}.

Table A3: **Finite-budget edge aggregation audit.** Summed edge ledgers against the exact quotient-game Shapley value under the same utility. nL1 is the normalized ℓ_1 allocation error, ρ the edge Spearman correlation, Top-1 the runs that recover the highest-valued edge, and the last column the largest efficiency residual at $N \geq 50$. Deployment-scale blocks use a seeded random roster and the $N=12$ block a label-histogram clustering. The model-utility block uses 48 detection cases with a 900-second budget, and its reference is the FedCCE edge solver applied to the model characteristic function. †Equal to the reference by construction, a correctness check rather than a competing estimator. ‡Median, since nL1 is unstable when exact edge values are near zero. Flat estimates at $N=12$ are truncated and conserve only to 3.4×10^{-4}.

		$N=12$, $E=4$ 4 datasets, 12 runs		$N=50$, $E=8$ 4 datasets, 20 runs			$N=100$, $E=10$ 2 datasets, 6 runs		
Edge ledger	M	nL1↓	Top-1	nL1↓	ρ ↑	Top-1	nL1↓	Top-1	max |res|
Summary utility, reference = exact edge quotient Shapley under the same utility									
Client-Owen sum	16	.060±.024	8/12	.353±.112	.287	7/20	.545±.081	0/6	3.3×10^{-16}
	64	.072±.025	9/12	.201±.063	.527	6/20	.226±.083	0/6	
	128	.044±.022	8/12	.131±.046	.560	9/20	.172±.052	0/6	
Flat-Shapley sum	16	.089±.042	10/12	.225±.110	.507	5/20	.266±.076	1/6	3.3×10^{-16}
	64	.085±.030	11/12	.119±.040	.561	12/20	.153±.051	2/6	
	128	.063±.030	12/12	.096±.037	.665	10/20	.129±.032	2/6	
Exact quotient (reference)†	any	0	12/12	0	1.000	20/20	0	6/6	0
Model utility, reference = exact model quotient Shapley									
Model-Owen client sum	900 s	–	–	.986‡	.575	28/48	–	–	4.2×10^{-17}

Table A4: **Summation under a model utility, by dataset and heterogeneity.** Budgeted Model-Owen client values are summed within each edge and compared with the exact model quotient Shapley value on the 48 detection cases ($N = 50$, $E = 8$, 900-second budget).

Group	Cases	nL1 (median)	ρ	Top-1	$\max \lvert\text{res}\rvert$
CIFAR-10	12	1.318	.339	5/12	4.2×10^{-17}
CIFAR-100	12	1.398	.516	8/12	6.9×10^{-18}
Fashion-MNIST	12	.415	.883	10/12	4.2×10^{-17}
Tiny-ImageNet	12	2.132	.562	5/12	1.2×10^{-17}
$\alpha = 0.1$	12	1.583	.504	7/12	4.2×10^{-17}
$\alpha = 0.5$	12	.986	.538	10/12	1.7×10^{-17}
$\alpha = 1.0$	12	1.080	.623	5/12	4.2×10^{-17}
$\alpha = 10.0$	12	.591	.635	6/12	2.8×10^{-17}

Table A5: **Summation audit by dataset.** Runs in which within-edge sums of budgeted client Owen values recover the highest-valued edge, for each permutation budget M, together with the normalized ℓ_1 edge error at the smallest and largest budget. Clean runs, five seeds per dataset at N=50, E=8 and three at N=100, E=10. Direct quotient enumeration recovers the leading edge in every run. Budgets are nested prefixes of one permutation stream.

		Top-1 recovered at budget M					nL1	
N	Dataset	8	16	32	64	128	M=8	M=128
50	CIFAR-10	2/5	2/5	3/5	2/5	1/5	.638	.161
50	CIFAR-100	0/5	1/5	1/5	1/5	2/5	.480	.125
50	Fashion-MNIST	2/5	2/5	2/5	2/5	3/5	.599	.150
50	Tiny-ImageNet	1/5	2/5	2/5	1/5	3/5	.343	.088
100	CIFAR-10	0/3	0/3	1/3	0/3	0/3	.726	.169
100	Fashion-MNIST	0/3	0/3	0/3	0/3	0/3	.734	.174

Table A6: **Direct edge sampling at equal cost, measured.** Twenty-six clean deployments with 50 independent streams per budget. By Theorem 1(i) the summed ledger at budget M is the edge-order estimator with M orders, and 125 of 130 real client-level sums of Table A3 fall inside the central range of its simulated distribution. Spending the same MN marginal evaluations on the quotient game gives MN/E orders. Top-1 is the fraction of runs that recover the highest-valued edge. Direct sampling has the lower error in 364 of 364 case-budget cells, and its measured standard-error ratio lies within 25% of $\sqrt{E/N}$ in all of them.

				Summed, M orders		Direct, MN/E orders		Standard-error ratio	
N	E	M	MN/E	nL1	Top-1	nL1	Top-1	measured	$\sqrt{E/N}$
50	8	8	50	.528	26%	.217	33%	.406	.400
50	8	16	100	.379	26%	.146	41%	.384	.400
50	8	64	400	.189	35%	.076	55%	.406	.400
50	8	128	800	.134	41%	.054	62%	.406	.400
100	10	8	80	.809	15%	.245	20%	.305	.316
100	10	16	160	.577	14%	.178	18%	.315	.316
100	10	64	640	.274	17%	.088	30%	.325	.316
100	10	128	1,280	.196	16%	.064	34%	.322	.316

Table A7: **Summation audit across the number of clients and edges.** Clean deployments repartitioned into seeded random rosters (four datasets and five seeds at $N{=}50$, two datasets and three seeds at $N{=}100$). CV is the coefficient of variation of the exact edge values. A regression of $\log$ nL1 over the 294 run-budget pairs gives coefficients 0.47 ± 0.07 on $\log E$, 0.48 ± 0.09 on $\log N$, -0.49 ± 0.02 on $\log M$, and -0.23 ± 0.04 on $\log$ CV, so the error grows with both E and N and falls at the $M^{-1/2}$ rate of Theorem 1.

			Summed, $M{=}16$		Summed, $M{=}128$		Direct
N	E	CV	nL1	Top-1	nL1	Top-1	nL1, $M{=}128$
50	4	.065	.283	7/20	.101	6/20	.033
50	6	.103	.335	7/20	.126	12/20	.036
50	8	.123	.365	3/20	.133	10/20	.061
50	10	.128	.293	3/20	.119	6/20	.057
100	4	.032	.447	3/6	.157	6/6	.035
100	8	.077	.553	1/6	.217	2/6	.068
100	10	.087	.549	2/6	.173	0/6	.075

Table A8: **Summation audit across deployment topologies** ($N{=}50$, $E{=}8$, twenty clean deployments). Unbalanced edge sizes spread the edge values, which makes the leading edge easier to recover, yet the summed Owen ledger keeps an allocation error of 0.11 with an exact total. Flat Shapley sums stop improving under unbalanced edges because their target is not the quotient value. The last column is the largest efficiency residual of the summed Owen ledger.

		Client-Owen sum			Flat-Shapley sum		Direct	
Topology	CV	nL1, 16	nL1, 128	Top-1, 128	nL1, 16	nL1, 128	nL1, 128	$\max \lvert\text{res}\rvert$
Random roster	.141	.353	.131	9/20	.225	.096	.061	$2.2{\times}10^{-16}$
Label k-means	.164	.354	.121	9/20	.228	.111	.057	$2.2{\times}10^{-16}$
Data volume	.133	.364	.130	10/20	.215	.111	.060	$4.4{\times}10^{-16}$
Unbalanced sizes	.355	.377	.113	13/20	.347	.251	.057	$2.2{\times}10^{-16}$

Table A9: **Summation audit under the summary utility, by heterogeneity.** Forty-eight deployments with hidden label corruption ($N{=}50$, $E{=}8$, four datasets, three seeds per α). The failure appears at every heterogeneity level, as it does under the model utility in Table A4.

	$M{=}16$		$M{=}64$		$M{=}128$		
α	nL1	Top-1	nL1	Top-1	nL1	Top-1	$\max \lvert\text{res}\rvert$
0.1	.347	4/12	.193	5/12	.105	8/12	$2.2{\times}10^{-16}$
0.5	.406	4/12	.225	3/12	.121	5/12	$1.1{\times}10^{-16}$
1	.416	4/12	.231	5/12	.126	5/12	$2.2{\times}10^{-16}$
10	.420	4/12	.230	4/12	.124	6/12	$1.1{\times}10^{-16}$

A.8 Flagging Harmful Participants at Deployment Scale

Table A10 ranks hidden label-corrupted clients at deployment scale. FedCCE lies above the 95% band of a random scorer on every dataset, has the highest AUPRC on CIFAR-10, and beats Model-Owen, Model-TMC, and GTG-Shapley in at least ten of twelve dataset-seed clusters.

Table A10: **Flagging harmful participants.** Mean AUPRC for hidden label-corrupted clients at N=50, E=8 (twelve cases per dataset, 900-second budget). Wins counts dataset-seed clusters in which a method beats FedCCE, with the unadjusted exact sign-test p. After Holm correction over the six comparisons only the differences from GTG-Shapley and DataSize remain significant. A random scorer has mean AUPRC .257, with 95% bands [.217, .305] for a dataset mean and [.236, .280] for the average over 48 cases.

	AUPRC							
Method	CIFAR-10	CIFAR-100	F-MNIST	Tiny	Average	AUROC	P@K	Wins vs. ours
Summary game								
FedCCE (ours)	**.756**	.537	.624	.457	**.593**	**.775**	**.521**	–
Flat Proto-TMC	.636	**.626**	.517	**.467**	.561	.773	.475	7/12 (.774)
Model utility								
FedOwen	.413	.363	**.766**	.314	.464	.694	.356	4/12 (.388)
Model-TMC	.389	.347	.676	.284	.424	.641	.342	2/12 (.039)
Model-Owen	.310	.304	.651	.269	.383	.631	.327	2/12 (.039)
GTG-Shapley	.433	.280	.370	.280	.341	.598	.238	0/12 (<.001)
Data proxy								
DataSize	.268	.221	.231	.224	.236	.483	.171	0/12 (<.001)
Independent random null	.257	.256	.259	.259	.258	.501	.198	

A.9 Ranking Corrupted Edges

Table A11 uses the injected corruption of Table A10 as an edge-level ground truth. The exact edge ledger correlates positively with the corrupted fraction in all 48 cases and finds the most corrupted edge in 56 percent of them against 12 percent by chance. This is about twice the rate of the summed ledger of the same client values, which reaches 27 percent, and the exact ledger is ahead in eight of twelve dataset-seed clusters.

Table A11: **Ranking corrupted edges.** Forty-eight deployments with hidden label corruption (N=50, E=8). Each edge score is divided by the edge size and compared with the fraction of flipped samples in the edge. The first column is the Spearman correlation with the negative flipped fraction, and the second the fraction of cases whose lowest-scored edge is the most corrupted one, against 12% for a random choice. The exact edge ledger correlates positively in 48 of 48 cases. Against the summed ledger of the same client values it wins 8 of 12 dataset-seed clusters, which is not significant (exact sign test $p = 0.39$). Exact edge values of eight cases that were not delivered within 900 seconds are taken from the deterministic replay of the same caches.

Edge ledger	Rank correlation	Lowest edge is most corrupted
FedCCE, exact edge ledger	.611	56%
FedCCE client ledger, summed	.318	27%
Flat Proto-TMC, summed	.434	50%
FedOwen, summed	.258	38%
Model-Owen, native edge	.249	29%
Model-TMC, summed	.121	6%
GTG-Shapley, summed	.108	4%

A.10 Sensitivity to the Utility Metric and to Covariance Weighting

Table A12 replaces the accuracy utility with a Brier utility on the Fashion-MNIST placements of Figure 4a. Every corrupted client stays negative, and the ranking agrees closely with the accuracy ledger. Table A13 shows that diagonal covariance weighting raises agreement with the model game at both resolutions at some cost in detection. We keep it fixed across all tasks rather than tuning it per dataset.

Table A12: **Sensitivity to the utility metric.** Exact client Owen values on the twelve Fashion-MNIST placements of Figure 4a (N=12, E=4, 27 corrupted and 117 benign clients) under the default accuracy utility and under a Brier utility that scores softmax probabilities of the negative Mahalanobis distances at three temperatures T. Margin is the smallest gap between the lowest benign and the highest corrupted value in a case, and a negative margin means the two groups overlap. Against the accuracy ledger, the T=1 Brier ledger has mean Spearman .946 (minimum .860) and sign agreement 91%. The recomputed accuracy utility matches the original exact values to 1.8×10^{-8}.

Utility	Corrupted < 0	Benign < 0	Bottom-k recall	Min. margin
Accuracy (default)	100%	0.0%	1.000	.019
Brier, T=0.5	100%	9.4%	.917	−.018
Brier, T=1	100%	11.1%	.917	−.005
Brier, T=2	100%	13.7%	1.000	.008

Table A13: **Diagonal covariance is a task-dependent trade-off.** Detection is measured on the 48 injected-corruption cases and alignment on the exact Model-Owen reference. Cluster statistics use the 12 dataset-seed clusters.

Variant	Detection AUPRC	Client alignment ρ	Edge alignment ρ
With covariance (default)	.593	.332	.550
Without covariance	.679	.251	.383
Clusters favoring no covariance	10/12 (p=.039)		

A.11 Removal Stress Test

The intervention of Figure 4b uses the small deployment, where the effect of a removal can be traced. At $N = 50$ and $E = 8$ we removed the clients flagged by each method and continued training for 20 and 40 rounds. Removal there changes label coverage, aggregation weights, and the optimization path together with data quality, so even an oracle that removes all ten corrupted clients gains only 3.33 and 1.48 points, and post-removal accuracy at this scale does not order methods by attribution quality.

A.12 Temporal Ledger and Availability

Table A14 measures how the temporal ledgers move across accounting rounds within one trajectory. The edge ledger is the most stable, and its first and last accounting rounds rank edges identically in all three trajectories. Table A15 counts runs whose complete ledgers finish within fixed wall-clock budgets. These timings were measured before validation-embedding caching and under concurrent load, so they bound availability from below, and the main text therefore reports costs in utility calls.

Table A14: **Stability across accounting rounds** within one trajectory (CIFAR-10, $N = 12$, $E = 4$, accounting at rounds 10, 20, 30, and 40, three trajectories). Adjacent pairs $n = 9$, first and last accounting rounds $n = 3$. Top-k overlap uses the pre-registered k of each ledger. This is same-trajectory stability, not cross-seed reproducibility.

	Adjacent rounds		First vs. last round	
Ledger	Spearman	Top-k overlap	Spearman	Top-k overlap
Client contribution ledger	.739	.444	.676	.167
Harmful-client set	.734	1.000	.552	1.000
Edge contribution ledger	.844	.778	1.000	1.000

Table A15: **Cost of the exact edge ledger and ledger availability.** Exact hierarchical support counts every coalition queried by exact client Owen plus exact quotient-game accounting. The incremental edge calls are the quotient-game queries not already issued by client sampling ($M = 16$, CIFAR-10 and Fashion-MNIST, three seeds each). Availability counts runs whose complete edge ledger finished within the wall-clock budget. Timings come from the unoptimized path before validation-embedding caching and were measured under concurrent load.

		Exact		Incremental	Edge ledger complete		
N	E	support	2^E	edge calls	90 s	300 s	900 s
12	4	208	16	0.0	6/6	6/6	6/6
50	4	196,560	16	0.7	3/6	6/6	6/6
100	4	1,073,741,776	16	0.7	0/6	6/6	6/6
50	8	80,128	256	173.0	0/6	3/6	6/6
100	10	5,233,664	1024	903.7	0/6	0/6	3/6

A.13 Wall-Clock Cost and Evidence Footprint

Figure 5 and this subsection time the utilities on one RTX 6000 Ada that runs no other computation. Each benchmark reads the stored accounting inputs of a finished deployment, namely the global state, the client updates, the class summaries, and the validation view, so no training or data loading enters the timings. Both utilities are queried on the coalition roster of each deployment in alternating order, five times over. The harness implementations behind the tables are timed together with GPU-resident implementations of the same computations. The GPU-resident summary utility returns the harness utility on every coalition to within 10^{-15}, and the GPU-resident model utility differs on at most four of 32 coalitions by one validation example, a difference of the convolution kernels. The per-round times multiply the call counts of Tables 1 and A16 by the measured query cost and are therefore estimates, whereas Table A15 reports end-to-end availability of the earlier harness under concurrent load. The client compute in Figure 5b uses inputs of each deployment's shape with data loading excluded for both utilities. The advantage is smallest on Fashion-MNIST, where the LeNet forward pass of a model query is itself cheap.

Table A16: **Cost of the client ledger** in utility calls per accounting round (a single round at N=12), FedCCE at M=64. Client ρ is relative to the exact value or *a sampled full-Owen reference.

		Calls per round			
N	E	Exact	FedCCE	Saving	Client ρ
12	4	208	175	1.2×	.926
50	8	80,128	2,701	30×	.752*
100	10	5,233,664	6,676	784×	–

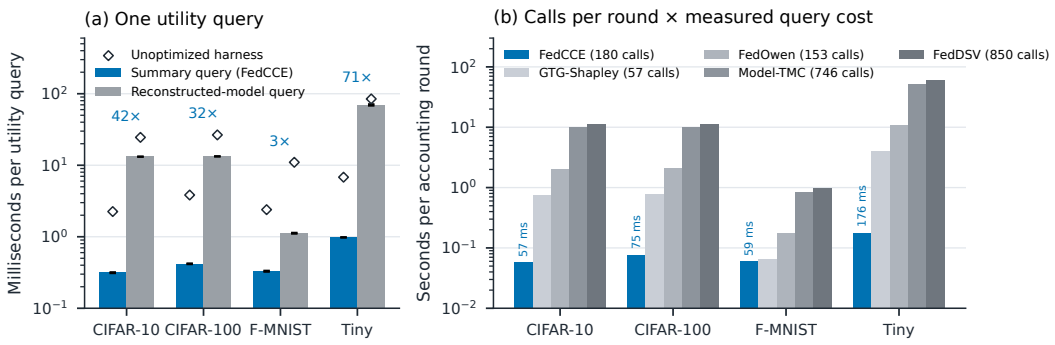

Figure A4: **Wall-clock cost of a utility query** (N=12, E=4, three seeds). (a) Median time of one utility query for the class summary game and a reconstructed-model utility that averages member updates and evaluates the validation view. Bars use GPU-resident implementations and diamonds the harness behind the tables. (b) Calls per accounting round from Table 1 multiplied by the measured query cost.

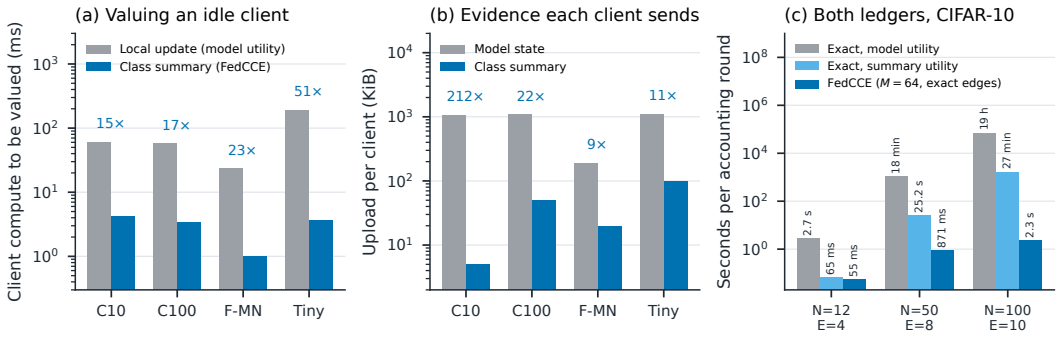

Figure A5: **Evidence footprint** (N=50, three seeds). (a) Client compute needed to be valued in an accounting round, one local update of 15 to 31 SGD steps for a model utility against a forward pass over at most three batches for a class summary. (b) Bytes each client sends for accounting. (c) Time per accounting round on CIFAR-10 for exact enumeration of both ledgers and for FedCCE with sampled client values (M=64) and exact edge values.

A.14 Limitations and Future Work

FedCCE takes the partition in force at each accounting round as given and does not optimize the association between clients and edges. The edge ledger is exact for the declared summary game and relative to the transmitted summaries, and exact enumeration is evaluated up to ten active edges, beyond which FedCCE samples the quotient game at the edge resolution as analyzed in Corollary 1. The summary game measures contribution through a nearest-class-mean learner on a shared encoder, which agrees with model-based attribution in part rather than in full (Appendix A.5), and exact model references are available only for the small deployment. Absent clients are completed by a simple historical rule. Class summaries reveal per-class feature moments of the shared encoder and come without a formal privacy guarantee, so combining them with secure aggregation or differential privacy is a natural extension, as is accounting for strategic reporting of summaries. Wall-clock costs depend on the implementation, so the main text reports costs in utility calls (Appendix Table A15). Future work includes deeper hierarchies with one quotient game per level, other learners whose sufficient statistics add up across clients, and adaptive accounting budgets driven by the observable edge error of Proposition 4.