FROM CLIENTS TO SAMPLES: TRACING DATA INFLUENCE IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how individual training data shape model behavior is essential for error diagnosis and data curation, particularly in medical applications. In Federated Learning (FL), tracing training-data influence is particularly challenging because data remain decentralized, and requires separating the effects of decentralized client updates from those of individual local samples. Existing methods primarily focus on quantifying client-level contributions, offering limited insight into the individual samples that shape their local models. Such aggregate assessments can mask samples with heterogeneous or opposing effects on model behavior, making targeted data inspection difficult. To address these challenges, we propose Federated Hierarchical Influence Tracing (FedHIT), an attribution framework for medical FL that traces influence at two complementary levels. Level 1 attributes query-specific global model behavior to individual clients' local updates, identifying the clients most responsible for performance on a given query. Level 2, applied within a client identified at Level 1, attributes that client's local model behavior to its own training samples, identifying which examples drive its local model. This decomposition connects global behavior to responsible clients and, separately, a client's local behavior to responsible samples, without requiring influence to be traced end-to-end from individual samples to the global model. We validate FedHIT on three benchmark datasets, demonstrating that it reliably identifies influential clients and samples at each level, enabling targeted inspection of the data underlying client- and global-model behavior in federated medical settings. By restricting sample-level analysis to selected clients, FedHIT avoids exhaustive attribution over all decentralized samples, providing a computationally efficient approach to targeted data inspection and curation.

1 INTRODUCTION

Machine learning models are sensitive to variations in data across sources and settings Liu et al. (2025b); Agarwal et al. (2025). This sensitivity is particularly consequential in medical domain, where institutions differ in patient populations, acquisition protocols, and annotation practices Yang et al. (2025); Jiang & Zhang (2025); Liu et al. (2025a). Federated learning (FL) enables these institutions to collaboratively train a shared model by aggregating locally computed updates while keeping their raw training data decentralized McMahan et al. (2017). This decentralized training process, however, makes model diagnosis difficult. Aggregate performance metrics do not reveal which client updates shaped a particular global-model behavior, nor do they identify the training samples that shaped the corresponding local models. Addressing these questions requires attribution methods that reflect the hierarchical structure of federated training Kumar et al. (2025); Zeng et al. (2026); Wang et al. (2025); Rubinstein & Hopkins (2026).

Hierarchical tracing vs. end-to-end influence propagation. In FL, influence arises at both the client and sample levels Tang et al. (2025); Zeng et al. (2026). At the global level, the target is the effect of a client's update at a particular communication round on model behavior specified by a query, such as performance on an individual prediction or evaluation subset Bae et al. (2024); Wang et al. (2025). At the local level, the target is the effect of individual training samples on the local model produced by that client during the selected round Zhang et al. (2025); Deng et al. (2025). These levels correspond to different attribution targets where client-round influence is measured *w.r.t.* a query on the global model, whereas sample influence is measured *w.r.t.* the selected client's

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.

local model. These targets support a hierarchical diagnostic workflow: *global query → influential client-round update → influential samples within that client's local trajectory*. Importantly, this hierarchy does not constitute end-to-end propagation of sample influence through global aggregation and subsequent training rounds. The sample-level score characterizes how that sample affects the selected client's local model, and does not directly quantify the sample's effect on the final global model Hu et al. (2026).

Challenges in hierarchical influence attribution. A client-level score alone provides limited guidance for inspecting the data within an influential client Zhang et al. (2025); He et al. (2025). Samples belonging to the same client may have heterogeneous or opposing effects on its local model, which can be obscured by an aggregate client score Bae et al. (2024); Deng et al. (2025). For example, identifying an influential hospital does not reveal whether its local behavior reflects informative clinical patterns, inconsistent annotations, or acquisition artifacts; distinguishing among these possibilities requires examining the samples that shaped its local model Yang et al. (2025); Liu et al. (2025a); Jiang & Zhang (2025). Influence is also query-dependent: clients that strongly affect a particular prediction or evaluation subset may differ from those ranked highly according to overall performance Park et al. (2023); Deng et al. (2025). Moreover, the temporal structure of FL complicates attribution because the effect of a client update can depend on the round in which it is introduced and on how later updates transform the global model Wang et al. (2025); Zeng et al. (2026). These challenges motivate a query-driven analysis that preserves the distinction between global client-update attribution and local sample attribution.

Limitations of existing methods. Existing methods address individual parts of federated data influence, but not the hierarchical diagnostic setting considered here. FedIF Tang et al. (2025) estimates client contributions from updates and validation gradients but does not identify influential local samples, while FedBary Li et al. (2024) performs distribution-based client or dataset valuation rather than query-specific client-round attribution followed by sample inspection. At the sample level, classical influence functions rely on optimality and curvature assumptions that may not hold for neural networks Koh & Liang (2017), whereas TRAK Park et al. (2023) scales training-example attribution without modeling federated aggregation or distinguishing global and local targets. Full trajectory differentiation captures optimization dynamics at substantial storage or recomputation cost; SOURCE Bae et al. (2024) reduces this overhead through approximate unrolling but is not formulated as a two-level FL diagnostic. Ripple Shapley Zeng et al. (2026) instead propagates sample contributions across global rounds. Thus, existing approaches provide client valuation, sample attribution, or end-to-end propagation, but not a query-driven workflow that first localizes global behavior to client-round updates and then examines samples shaping the selected local models. A detailed discussion of related works is provided in the Appendix.

We address this gap with **Federated Hierarchical Influence Tracing (FedHIT)**, a two-level diagnostic framework for medical FL. Given a query specifying the global-model behavior of interest, Level 1 attributes that behavior to individual client-round updates, identifying *where* to investigate. Level 2 then attributes the selected client's local-model behavior to its training samples, identifying *what* to inspect within that local context. Crucially, FedHIT combines two distinct attribution targets without treating them as a single end-to-end influence chain. This query-driven hierarchy distinguishes FedHIT from prior approaches that focus on only one part of the attribution problem: FedIF Tang et al. (2025) estimates client-level contributions, TRAK Park et al. (2023) and SOURCE Bae et al. (2024) provide training-example attribution, and Ripple Shapley Zeng et al. (2026) propagates sample influence across federated rounds. In contrast, FedHIT combines client localization with targeted local sample inspection while deliberately preserving the two attribution targets as distinct. Level 2 scores are not propagated through aggregation or subsequent rounds, and therefore are interpreted as local diagnostic quantities rather than direct estimates of a sample's influence on the final global model. By restricting sample-level analysis to influential client-rounds, FedHIT avoids exhaustive inspection across all decentralized samples and provides a focused workflow for investigating the distributed data involved in a queried behavior. It also extends the client- or dataset-level assessments of FedIF Tang et al. (2025) and FedBary Li et al. (2024) by using global attribution to guide targeted local sample inspection. FedHIT thus organizes complementary global and local analyses into a query-driven hierarchy while preserving their distinct interpretations.

Our main contributions are summarized below.

- **Query-specific, two-target formulation.** We formulate hierarchical influence analysis in FL using two explicit attribution targets: the influence of client-round updates on query-specific global-model behavior and the influence of training samples on selected local models. By separating these targets, the formulation distinguishes hierarchical diagnostic tracing from end-to-end propagation of sample influence to the final global model.
- **Gradient-based hierarchical attribution.** We introduce FedHIT, which follows the diagnostic workflow *global query → influential client-round update → influential local samples*. The client-level analysis guides targeted sample-level investigation while preserving the distinct global and local attribution targets. To the best of our knowledge, FedHIT is the first query-driven, gradient-based FL framework to organize these targets into a hierarchical diagnostic workflow.
- **Medical benchmark evaluation.** We evaluate FedHIT on three benchmark medical datasets, namely, HAM10000 (HAM10K) Tschandl et al. (2018), BreakHis Spanhol et al. (2016), and Matek-19 Matek et al. (2019), to assess its client-round and local sample attributions and its ability to support targeted data inspection and curation in medical FL.

2 PRELIMINARIES

FL setup. We consider a federated system with K clients and a coordinating server. At communication round $r \in \{0, \dots, R-1\}$, the participating clients form the set $\mathcal{P}^{(r)}$. Client $i \in \mathcal{P}^{(r)}$ holds a local dataset $\mathcal{D}_i^{(r)}$ with $n_i^{(r)} > 0$ examples, and $\ell(z;\theta)$ denotes the training loss of a model $\theta \in \mathbb{R}^d$ on example z. The local objective is $F_i^{(r)}(\theta) = \frac{1}{n_i^{(r)}} \sum_{z \in \mathcal{D}_i^{(r)}} \ell(z;\theta)$. The server broadcasts the global parameters $\theta^{(r)}$; each participating client initializes at $\theta^{(r)}$, performs local training, and returns $\theta_i^{(r)}$. Its update and aggregation weight are $\Delta\theta_i^{(r)} = \theta_i^{(r)} - \theta^{(r)}, \alpha_i^{(r)} = \frac{n_i^{(r)}}{N^{(r)}}, N^{(r)} = \sum_{j \in \mathcal{P}^{(r)}} n_j^{(r)}$. Under FedAvg McMahan et al. (2017), the server aggregates $\theta^{(r+1)} = \theta^{(r)} + \sum_{i \in \mathcal{P}^{(r)}} \alpha_i^{(r)} \Delta\theta_i^{(r)}$. Thus, $\alpha_i^{(r)} \Delta\theta_i^{(r)}$ is client i's weighted contribution to the global update at round r.

2.1 ATTRIBUTION TARGETS

Query-conditioned evaluation. FedHIT formulates attribution in a query-conditioned manner: z_q denotes the query whose behavior we analyze, and $g_q^{(r)} = \nabla_\theta \ell\big(z_q; \theta^{(r)}\big)$. This is the query-loss gradient at the round-start global model, measuring the sensitivity of the query loss to $\theta^{(r)}$. All Level-1 quantities are conditioned on z_q, following the pipeline

$$z_q \to g_q^{(r)} \to s_i^{(r)}(z_q) \to \text{routed pairs } (c, r) \to \text{Level-2 sample attribution},$$

so a single query fixes both the client scores and the client-round pairs sent for sample-level analysis.

Sample-level counterfactual. Sample attribution is performed for a routed client–round pair (c, r). Let the client's local dataset in that round be $\mathcal{D}_c^{(r)} = \{z_j\}$ of size $n_c = n_c^{(r)}$, and let $\vartheta_T(S; \lambda, \xi)$ denote the local training map returning the parameters after T local steps on S, initialized from $\theta^{(r)}$, with hyperparameters λ (learning-rate schedule, batch size, local horizon T) and randomness ξ governing the local training procedure. We use ϑ for local-trajectory parameters and θ for global parameters, and reserve T for local steps rather than communication rounds.

For a differentiable scalar measurement $f(z_q, \vartheta_T)$ of the query at the local endpoint, the ideal counterfactual is leave-one-out (LOO) retraining,

$$\tau_{\mathrm{LOO}}(z_q, z_j; \xi) = f\Big(z_q, \vartheta_T\Big(\mathcal{D}_c^{(r)} \setminus \{z_j\}; \lambda, \xi\Big)\Big) - f\Big(z_q, \vartheta_T\Big(\mathcal{D}_c^{(r)}; \lambda, \xi\Big)\Big). \tag{1}$$

This isolates the effect of removing z_j while the initialization and training configuration are held fixed. As a single realization is noisy, we target the expected effect $\bar{\tau}_{\mathrm{LOO}}(z_q, z_j) = \mathbb{E}_\xi\left[\tau_{\mathrm{LOO}}(z_q, z_j; \xi)\right]$. Direct computation requires retraining for every sample; FedHIT instead adopts the gradient-based approximation of SOURCE Bae et al. (2024) $\bar{\tau}_{\mathrm{LOO}} \approx \tau_{\mathrm{SOURCE}}$, written $\tau_c^{(r)}$ for a routed pair (c, r) in the methodology, which needs no retraining. When f is the query

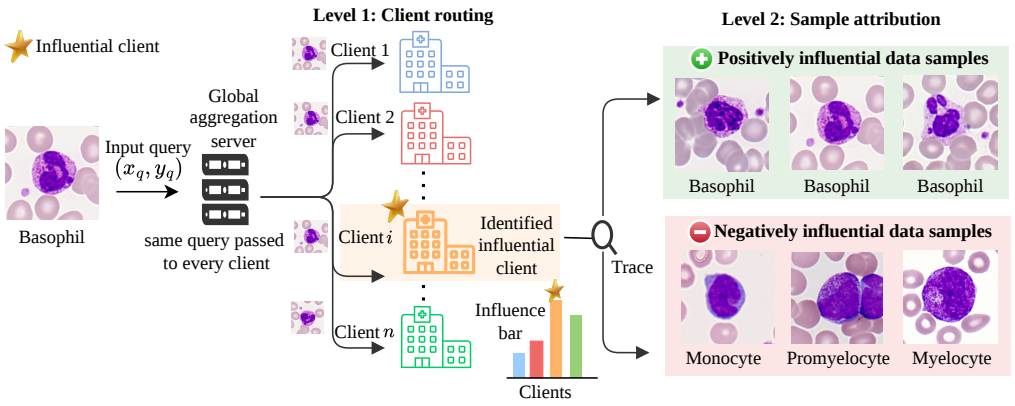


Figure 1: Overview of the FedHIT framework, illustrating Level 1 client-update attribution for query-specific global model behavior and Level 2 sample-level attribution within the selected clients.

loss, the Level-2 measurement coincides with the loss used for Level-1 routing; in general f may differ (e.g., a prediction margin), as discussed with the sign convention below.

3 Proposed Methodology

We propose **Federated Hierarchical Influence Tracing (FedHIT)**, illustrated in Figure 1, a query-conditioned attribution framework that connects global client-level contribution with local sample-level attribution through two complementary levels. In FL, the training data is partitioned across clients and never leaves them, so centralized training-data attribution that would explain a query by pointing to individual training samples is unavailable, as the server observes client updates but not the individual samples that produced them. Given a query z_q, FedHIT (Federated Hierarchical Influence Tracing) answers two nested questions: (i) which client-round pairs contributed most to the query, (ii) within a selected client, which local samples are its proponents and opponents, through two levels that run coarse-to-fine. Level 1 is a lightweight server-side screen that scores each participating client-round by aligning its FedAvg-weighted update McMahan et al. (2017) with the query-loss gradient. It reuses communicated updates and requires no sample-level information. The highest-scoring client rounds are routed to Level 2, where SOURCE Bae et al. (2024) analyzes each selected client's local trajectory to identify samples influencing the query on its local model. Because this analysis remains local and is restricted to a few clients, raw data stays on-device and only a small fraction of the federation is inspected. Thus, Level 1 determines where to investigate, and Level 2 identifies which local samples shaped the selected local model.

Level 1: Per-round client contribution. At round r, client i returns the update $\Delta\theta_i^{(r)} = \theta_i^{(r)} - \theta^{(r)}$, which enters the global model with weight $\alpha_i^{(r)}$. Its query-conditioned contribution score measures how far that weighted update moved the model in a direction that lowers the query loss, a first-order alignment between the contribution $\alpha_i^{(r)}\Delta\theta_i^{(r)}$ and the query-loss gradient $g_q^{(r)}$, in the spirit of gradient-tracing attribution lifted to the client-round level: $s_i^{(r)}(z_q) = -\alpha_i^{(r)}\left(\Delta\theta_i^{(r)}\right)^\top g_q^{(r)}$. A positive score means client i's weighted update is estimated to lower the query loss (a helpful contribution); a negative score, to raise it.

Per-round client ranking and routing. For a query z_q at round r, the scores $\{s_i^{(r)}(z_q) : i \in \mathcal{P}^{(r)}\}$ rank the participating clients from most positively to most negatively influential, e.g. ten clients in the round, z_q induces a full ordering of them for that round. No cumulation over rounds is taken: the ranking is per query and per round. Because Level 2 inspects only a few clients, FedHIT routes independently at each round; given a budget k, it selects the k most positively influential clients, $\mathcal{C}_k^{(r)}(z_q) = \mathrm{TopK}_{i\in\mathcal{P}^{(r)}}\, s_i^{(r)}(z_q)$. With ties broken by lowest client index, applied at every round, this routes the client-round pairs (c, r) with $c \in \mathcal{C}_k^{(r)}(z_q)$. The score is defined for every participating client; routing is only the computational choice of which pairs to expand at Level 2 and does not alter

the scores. Because Level 2 is meant to surface data that support the query behavior, we route the highest signed scores rather than the largest $|s_i^{(r)}(z_q)|$, so proponents are expanded first; negative-scoring client-rounds can be routed separately as harmful contributors. We use $k = 1$ by default and $k > 1$ to widen the search; since routing is independent per round, $k = 1$ routes at most one client per round, yielding up to R routed client-round pairs, each defining one Level-2 local trajectory.

Level 2: Sample-level attribution. Level 1 identifies the client-round pairs with the largest estimated contribution to the query. Level 2 then performs **conditional, client-local** sample attribution to trace which of a routed client's local training examples explain the query behavior of its own local trajectory. For a routed pair (c, r) we apply SOURCE Bae et al. (2024) to client c's local trajectory on $\mathcal{D}_c^{(r)}$ (size n_c), with the round-start model $\theta^{(r)}$, the local training configuration, and the randomness held fixed. Writing $\vartheta_{c,0}^{(r)} = \theta^{(r)}$ for the local start and $\vartheta_{c,T}^{(r)} = \vartheta_T(\mathcal{D}_c^{(r)})$ for client c's local endpoint, the target is the local LOO effect (1) and its SOURCE estimate is $\tau_c^{(r)}(z_q, z_j)$. This is attribution with respect to c's local model, not directly with respect to the post-FedAvg global model; the conditioning on the Level-1-selected pair is what lets the two levels compose.

Removal surrogate. Rather than retrain for each sample, the effect of excluding z_j is approximated by perturbing its loss weight by $1 + \epsilon$, $F_{c,\epsilon}(\vartheta) = \frac{1}{n_c} \sum_{z \in \mathcal{D}_c^{(r)}} \ell(z; \vartheta) + \frac{\epsilon}{n_c} \ell(z_j; \vartheta)$, where $\epsilon = 0$ recovers the unperturbed local objective $F_c^{(r)}$ and $\epsilon = -1$ suppresses z_j. This is the local instance of SOURCE's downweighting response function Bae et al. (2024): z_j carries weight $1 + \epsilon$, and the first-order effect of $\epsilon = -1$ approximates, rather than reproduces, retraining on $\mathcal{D}_c^{(r)}$ without z_j. The remaining equations instantiate SOURCE's approximation of this effect, replacing repeated retraining with a segmented, stationary approximation of the local training trajectory.

Segmented trajectory and curvature. We partition the T local steps into L segments $0 = T_0 < T_1 < \cdots < T_L = T$ with lengths $K_\ell = T_\ell - T_{\ell-1}$, recording C_ℓ checkpoints $\{\vartheta_{\ell,a}\}$ for $a = 1, \ldots, C_\ell$ in segment ℓ (a checkpoint observes the trajectory; a segment is a range of steps). Within segment ℓ we form the checkpoint-averaged sample gradient, mean learning rate, and segment curvature, $\bar{g}_\ell = \frac{1}{C_\ell} \sum_{a=1}^{C_\ell} \nabla_\vartheta \ell(z_j; \vartheta_{\ell,a}), \bar{\eta}_\ell, \bar{H}_\ell$. Here $\bar{g}_\ell$ is the segment-averaged gradient of the example z_j being attributed, not of the full client objective. Following SOURCE Bae et al. (2024), $\bar{H}_\ell$ is built with EK-FAC George et al. (2018), the eigenvalue-corrected refinement of K-FAC Martens & Grosse (2015), which approximates the Hessian by the generalized Gauss-Newton (GN) matrix Martens (2020). This GN form is positive-semidefinite when the loss is convex in the model output, avoiding the unstable exponential growth that negative-curvature directions of the raw Hessian would introduce during propagation.

Sample-level score. SOURCE Bae et al. (2024) turns these segment statistics into an attribution score. Removing z_j injects, within each segment, the curvature-corrected effect

$$\bar{r}_\ell = \frac{1}{N} F_r(\bar{H}_\ell) \bar{g}_\ell, \quad F_r(\sigma) = \frac{1 - \exp(-\bar{\eta}_\ell K_\ell \sigma)}{\sigma}. \tag{2}$$

This is applied to $\bar{H}_\ell$ as a matrix function through its eigendecomposition, with the analytic limit $F_r(0) = \bar{\eta}_\ell K_\ell$; in high-curvature directions F_r acts as an inverse-curvature adjustment, in flat directions as a plain gradient step. Each segment's effect is then propagated to the end of training by the segment operator $\bar{S}_\ell = \exp(-\bar{\eta}_\ell K_\ell \bar{H}_\ell)$. This suppresses high-curvature directions while preserving flatter ones; earlier segment effects are propagated through more subsequent operators, while the final segment needs no further propagation. Summed over segments and aligned with the query gradient at the final parameters, this yields the SOURCE score

$$\tau_{\mathrm{SOURCE}}(z_q, z_m, \mathcal{D}; \lambda) := \nabla_\theta f(z_q, \theta_T)^\top \left[\sum_{\ell=1}^{L} \left(\prod_{\ell'=L}^{\ell+1} \bar{S}_{\ell'} \right) \bar{r}_\ell \right]. \tag{3}$$

Here, each segment effect $\bar{r}_\ell$ is propagated forward through the later segments (the product runs over $\ell + 1, \ldots, L$ and is empty, hence the identity, for $\ell = L$) and then projected onto the query gradient. We apply this to the routed pair (c, r): the dataset is $\mathcal{D}_c^{(r)}$ so $N = n_c$, the final parameters θ_T are the endpoint of client c's local trajectory (the ϑ_T of the Preliminaries), and z_m ranges over the client's local examples z_j; we write the resulting routed-pair score as $\tau_c^{(r)}(z_q, z_j)$. It estimates how removing z_j would change the query measurement at that endpoint, approximating $\bar{\tau}_{\mathrm{LOO}}$.

Sign convention. The score above estimates the change in f caused by removing z_j. When $f = \ell(z_q; \cdot)$ is the query loss, $\tau_c^{(r)} > 0$ means removal is estimated to increase the query loss, so z_j is a proponent; $\tau_c^{(r)} < 0$ denotes an opponent. This matches the Level-1 convention, where positive $s_i^{(r)}$ denotes a helpful client contribution. For measurements where larger is better, such as prediction margin, the interpretation reverses: $\tau_c^{(r)} < 0$ denotes a proponent and $\tau_c^{(r)} > 0$ an opponent. If a unified positive-means-proponent convention is desired, we report $-\tau_c^{(r)}$ for such larger-is-better measurements. Algorithm 1 summarizing the overall FedHIT workflow and extended details of the proposed methodology are provided in the Appendix.

4 Experimental Results and Discussion

Datasets and federated partitions. We evaluate FedHIT on HAM10000 Tschandl et al. (2018), Matek-19 Matek et al. (2019), and BreakHis Spanhol et al. (2016), representing dermoscopic, peripheral-blood-smear, and breast-histopathology imaging with 7, 15, and 8 categories, respectively. Each dataset uses a fixed 80:20 train–test split, with patient or lesion groups kept intact when identifiers are available. The same training pool is then partitioned into 10, 20, or 50 clients. To model institutional heterogeneity, 40% of clients receive an approximately 80:20 dominant-to-nondominant class mixture, 20% receive a 90:10 mixture, and 40% are mixed clients, yielding type counts of $(4, 2, 4)$, $(8, 4, 8)$, and $(20, 10, 20)$, respectively. Each example belongs to exactly one client, with patient or lesion groups preserved whenever possible.

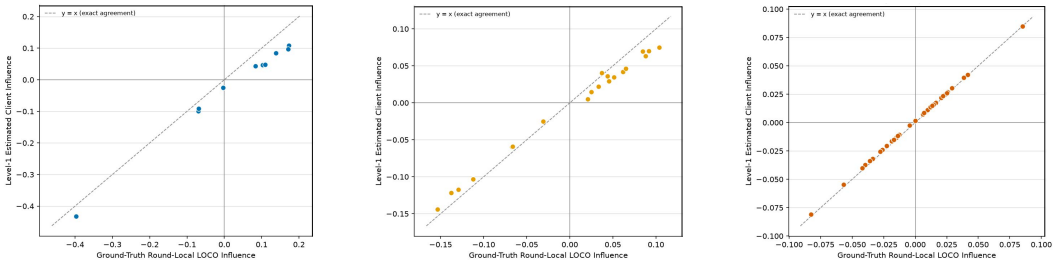
Model, queries, and system setup. We use ResNet-18 He et al. (2015) with full-participation FedAvg McMahan et al. (2017) for 200 communication rounds and 3 local epochs per client per round. We select 50 fixed, approximately class-balanced query images from the held-out test split; these are never used for training and are reused across all rounds, client-count settings, and attribution methods. The primary attribution metrics are Level 1 Spearman correlation, Linear Datamodeling Score (LDS), counterfactual changes in query behavior, and attribution GFLOPs. All experiments are run on an NVIDIA RTX A6000 GPU under identical conditions for fairness, with extended details provided in the Appendix.

4.1 Level 1 Client Attribution

We evaluate whether FedHIT Level 1 recovers query-specific client influence using a round-local leave-one-client-out (LOCO) counterfactual. For each query and aggregation round, LOCO removes one participating client's weighted update while keeping the other client updates in that round fixed, and measures the resulting change in query behavior. Since LOCO directly evaluates this query-round removal intervention, whereas FedHIT estimates the same effect from the communicated update and query gradient, we use LOCO as the ground-truth reference for this round-local setting. This reference does not measure influence over the complete FL training trajectory. We measure ranking agreement between FedHIT and LOCO using Spearman's correlation, computed separately for each query and round. The evaluation is conducted on Matek-19 Matek et al. (2019), HAM10000 Tschandl et al. (2018), and BreakHis Spanhol et al. (2016), while varying the number of clients and keeping the training pool fixed to isolate the effect of federation size. Figure 2 shows the Matek-19 results, where each point pairs a FedHIT score with its corresponding LOCO counterfactual. FedHIT achieves strong rank agreement across federation sizes, with mean **Spearman correlations** of **0.936**, **0.960**, and **0.988** for $K = 10$, 20, and 50 clients, respectively. These results indicate that Level 1 reliably recovers the client rankings induced by query-specific, round-local counterfactuals. Extended results on HAM10000 and BreakHis are provided in the Appendix.

4.2 Training Sample Attribution

We evaluate Level 2 independently of routing to avoid confounding sample-attribution quality with client selection. Using identical ResNet-18 models, preprocessing, trainable parameters, and queries, we compare FedHIT's SOURCE-based Level 2 estimator Bae et al. (2024) with TRAK Park et al. (2023), TracIn Pruthi et al. (2020), and classical influence functions Grosse et al. (2023); Koh & Liang (2017). We measure performance using the Linear Datamodeling Score (LDS) Park et al. (2023), which evaluates how well additive attribution scores predict query behavior after subset



(a) 10 clients (b) 20 clients (c) 50 clients

Figure 2: Agreement between FedHIT Level-1 client influence estimates and the ground-truth round-local leave-one-client-out (LOCO) influence for (a) 10, (b) 20, and (c) 50 clients for Matek-19 dataset. The dashed diagonal indicates exact agreement.

retraining. For subset fraction $\alpha \in \{0.3, 0.5\}$, we sample $M = 100$ subsets, and compute the Spearman correlation between predicted measurements, obtained by summing per-example scores, and observed measurements across subsets, averaged over queries. Figure 3 shows that FedHIT achieves the strongest and consistent agreement across all settings. Its modest LDS values (0.078-0.162) indicate consistent improvement rather than accurate reconstruction, highlighting the difficulty of neural training-data attribution.

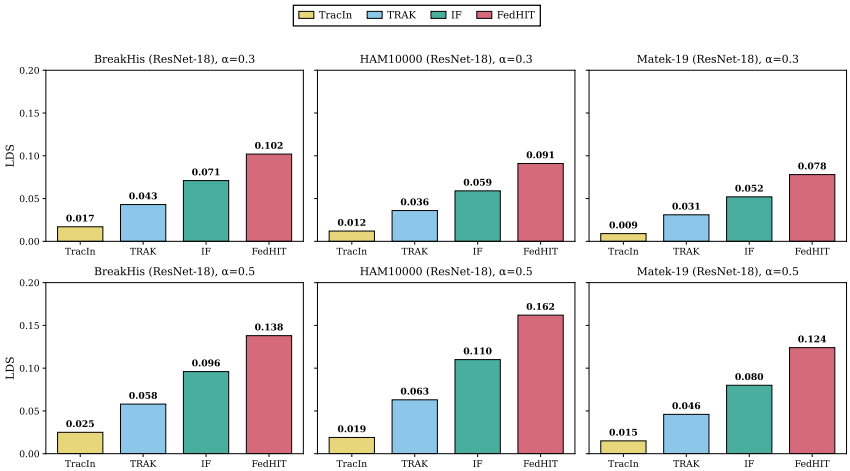


Figure 3: Linear Datamodeling Score (LDS) for training-sample attribution for all three datasets. Each value is the mean over query samples; higher is better.

4.3 Counterfactual removal evaluation

We test whether signed attribution scores predict removal effects at both levels of the hierarchy.

Client removal. We use client removal as a round-local counterfactual test of Level 1 attribution. For each query and round, clients are ranked by their Level 1 scores, and we remove the top 10%, 20%, or 30% under three rules: largest positive scores, most negative scores, or uniform random selection. Helpful (positively influential) clients are those whose updates support the queried prediction, so removing them should decrease the true-class margin; harmful (negatively influential) clients weaken the queried prediction, so removing them should increase the margin. Random clients serve as an unstructured baseline with smaller or less systematic changes. Removal is performed without retraining by excluding the selected trained client updates, renormalizing the remaining FedAvg weights, and recomputing the aggregate, thereby isolating the selected updates without adding local-training noise. Table 1 shows a consistent signed pattern across datasets and federation sizes: removing helpful clients lowers the margin, removing harmful clients raises it, and random removal remains comparatively close to M_0. For example, on Matek-19 with $K = 10$,

Table 1: Level-1 counterfactual client-removal results. Here, K: number of clients and M_0: mean query margin before client removal. The remaining entries report the mean margin after removing helpful, randomly selected, or harmful clients for Matek-19 dataset.

Dataset	K	M_0	**Helpful removed** (↓)			**Random removed** ($\approx M_0$)			**Harmful removed** (↑)		
			10%	20%	30%	10%	20%	30%	10%	20%	30%
HAM10000	10	7.2185	6.7790	6.2738	5.7195	7.1287	7.2909	7.3257	7.6241	8.0962	**8.7040**
	20	6.5484	6.1514	5.7488	5.2843	6.5579	6.5210	6.4828	6.8664	7.2614	7.7618
	50	5.4422	5.1209	4.7467	4.3563	5.4366	5.4552	5.5207	5.7806	6.0944	6.4152
Matek-19	10	1.593	0.911	0.345	-0.281	1.638	1.679	1.683	2.286	2.802	3.399
	20	3.069	2.705	2.465	2.248	3.120	3.095	3.145	3.394	3.627	3.842
	50	1.320	0.910	0.636	0.428	1.330	1.288	1.231	1.663	1.889	2.099
BreakHis	10	4.533	4.070	3.572	2.916	4.559	4.428	4.239	5.033	5.553	6.017
	20	3.365	2.914	2.479	2.069	3.370	3.397	3.266	3.992	4.427	4.915
	50	2.408	2.054	1.707	1.395	2.397	2.356	2.347	2.656	2.888	3.151

Table 2: Overall computational cost and runtime of FedHIT and exhaustive leave-one-out attribution. Runtime speedup is computed as the ratio of exhaustive runtime to FedHIT runtime.

Dataset	**Clients**	**Attribution cost (GFLOPs)** (↓)			**Runtime (s)** (↓)		
		FedHIT	**Exhaustive**	**Reduction (%)** (↑)	**FedHIT**	**Exhaustive**	**Speedup** (×) (↑)
HAM10000	10	609,900.7	48,308,580.0	98.7375	81.473	6,099.296	74.86
	50	405,884.2	1,906,697.0	78.7127	37.337	107.133	2.87
Matek-19	10	650,046.6	79,425,440.0	99.1816	33.2473	3,711.9570	111.65
	50	503,650.8	3,088,001.0	83.6901	25.3952	148.1696	5.83
BreakHis	10	557,625.8	19,312,900.0	97.1127	74.8587	2,853.8276	38.12
	50	266,269.3	889,326.5	70.0594	40.9004	128.8898	3.15

removing 30% of helpful clients reduces the margin from 1.593 to -0.281 (-117.6%), random removal changes it only to 1.683 ($+5.6\%$), and removing 30% of harmful clients increases it to 3.399 ($+113.4\%$). This directional behavior supports interpreting Level 1 scores as query-specific indicators of whether a client update helps or hurts the queried prediction. Additional sample-removal results are given in the Appendix.

4.4 Hierarchical routing and computational efficiency

The full-pipeline experiment compares FedHIT with a round-local exhaustive LOO reference. For each query, FedHIT scores all participating clients, routes the top three to its SOURCE-based Level 2 analysis Bae et al. (2024), whereas the reference computes LOCO for every client, selects its top three, and performs LOO retraining for every sample within them. Reported GFLOPs and wall-clock runtime exclude ordinary FL training and measure attribution overhead only; both are reported because GFLOPs omit data-loading, kernel-efficiency, and implementation costs. Table 2 reports the total attribution cost and runtime across datasets and federation sizes. FedHIT substantially reduces both relative to the exhaustive reference by avoiding exhaustive sample-level attribution and restricting the expensive Level 2 analysis to routed clients. Level 1 serves as a low-cost routing mechanism rather than being the main source of computational complexity. Absolute costs decrease as the federation grows because the fixed training pool is divided into smaller local datasets, reducing the number of samples processed within selected clients. This reduction outweighs the modest additional Level 1 cost of scoring more clients. The relative advantage narrows at larger federation sizes because the exhaustive reference also benefits from smaller local datasets. Runtime follows the same overall trend as GFLOPs, although the reductions differ because runtime also reflects data loading, kernel efficiency, and implementation overhead. Table 3 provides the component-wise cost breakdown confirming that Level 1 contributes only a small fraction of FedHIT's total cost, while Level 2 dominates due to local-trajectory reconstruction, curvature estimation, and sample attribution. The exhaustive reference is instead dominated by sample-level LOO retraining. FedHIT's efficiency therefore comes from using an inexpensive client-level screen to route the costly sample-level analysis to a small subset of clients.

Table 3: Computational cost for FedHIT and exhaustive attribution in GFLOPs (↓). Here, K: number of clients, TR: Total reduction is computed relative to exhaustive attribution.

Dataset	K	FedHIT (Ours)			Exhaustive			TR (%) (↑)
		Level 1 (↓)	Level 2 (↓)	Total (↓)	Level 1 (↓)	Level 2 (↓)	Total (↓)	
HAM10000	10	5.311	609,895.4	609,900.7	41.577	48,308,537.0	48,308,578.0	98.74
	50	5.982	405,878.2	405,884.2	226.945	1,906,470.0	1,906,697.0	78.71
Matek-19	10	5.3108	650,041.3	650,046.6	41.5778	79,425,400.0	79,425,440.0	99.18
	50	5.9825	503,644.8	503,650.8	226.9656	3,087,774.0	3,088,001.0	83.69
BreakHis	10	5.3107	557,620.5	557,625.8	41.5770	19,312,860.0	19,312,900.0	97.11
	50	5.9821	266,263.3	266,269.3	226.9472	889,099.5	889,326.5	70.06

Summary. Overall, the experiments support both FedHIT components. Level 1 closely matches round-local LOCO client rankings across all datasets and federation sizes, while SOURCE achieves the strongest Level 2 LDS among the evaluated sample-attribution methods. Routing each query to three clients also substantially reduces attribution cost relative to exhaustive client- and sample-level LOO. Thus, FedHIT efficiently localizes query-specific global behavior to influential clients and, through a separate local analysis, identifies influential samples within them.

5 Limitations and Potential Solutions

FedHIT provides a query-driven hierarchy for analyzing client-round influence on the global model and sample influence within selected local models, while also highlighting several limitations and directions for future work. (i) Level 2 scores are local to selected client-rounds and are not propagated through later aggregation rounds, so they do not directly measure final global-model influence or causality. Future work could explore truncated unrolling, Jacobian approximations, or explicit causal models, at added computational, storage, and privacy costs. (ii) Both levels are approximate: Level 1 omits higher-order client interactions and nonlinear aggregation effects, while Level 2 inherits SOURCE/EK-FAC stationarity and structured-curvature assumptions. Fidelity could be improved through Hessian-vector products, multi-client removal, adaptive segmentation, alternative curvature estimators, and removal-based validation. (iii) Our evaluation uses synchronous, full-participation FedAvg and assumes access to client updates and selected local attribution statistics. Future work should extend FedHIT to partial or asynchronous participation, heterogeneous computation, alternative aggregation, secure aggregation, differential privacy, client-side ranking, secure inner products, trusted execution, privacy-budget-aware attribution, transformer-based foundation models, and broader medical tasks such as clinical text, electronic health records, multimodal learning, segmentation, detection, and continual learning. Extended details are provided in the Appendix.

6 Conclusion

We introduced Federated Hierarchical Influence Tracing (FedHIT), a query-driven diagnostic framework that attributes global-model behavior to client-round updates at Level 1 and local-model behavior to samples within selected clients at Level 2. Across three medical image-classification benchmarks, using round-local LOCO validation, LDS comparisons, counterfactual removal, and routing analysis, FedHIT localizes query-relevant client-rounds and guides targeted local sample inspection under ResNet-18 with full-participation FedAvg. Its scores remain diagnostic, query-dependent approximations rather than causal explanations. Although FedHIT is broadly applicable to FL, this work focuses on medical FL, where institutional heterogeneity, class imbalance, and decentralized data make influence tracing especially important. Overall, FedHIT provides a practical diagnostic workflow for trustworthy federated learning, offering hierarchical attribution as an interpretable alternative to exhaustive influence propagation and a foundation for future research in explainability, debugging, and data-centric FL. Future directions includes extending FedHIT to cross-round sample propagation, interaction-aware client attribution, alternative curvature approximations, partial-participation FL, and broader medical tasks and modalities.

AI USE STATEMENT

Generative AI tools were used to support the preparation of this work as writing and research assistants, and to support code implementation, debugging, experimental-code review, and manuscript editing for clarity, structure, and readability. They were not used to create or fabricate datasets, results, measurements, or scientific claims at any stage. All experiments were conducted using the datasets, models, code, and procedures described in the paper, and the authors independently verified the reported results. Any AI-assisted code was reviewed, tested, and validated before use. The authors take full responsibility for all technical claims, results, analyses, conclusions, and final content of this work.

ETHICS STATEMENT

This work uses public medical benchmark datasets to study diagnostic attribution in federated learning, with the goal of supporting transparency and data curation; however, FedHIT scores should be interpreted as query-dependent diagnostic evidence rather than clinical decisions or causal judgments about institutions or individual samples.

REPRODUCIBILITY STATEMENT

Section 4 provides the detailed experimental protocol, including the datasets, federated partitions, model architecture, optimization settings and detailed result discussions. The experiments use public benchmark datasets and fixed train-test splits, with patient or lesion grouping preserved where identifiers are available. We report the client-count settings, federated partition design, hardware setup, and attribution procedures used for Level 1 and Level 2 evaluation. Additional implementation details, extended results, and ablation analyses are provided in the Appendix to support reproducibility.

REFERENCES

Aayushya Agarwal, Gauri Joshi, and Larry Pileggi. Fedecado: a dynamical system model of federated learning. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger Grosse. Training data attribution via approximate unrolling. *Advances in Neural Information Processing Systems*, 37:66647–66686, 2024.

Junwei Deng, Weijing Tang, and Jiaqi W. Ma. A versatile influence function for data attribution with non-decomposable loss. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31, 2018.

Bishwamittra Ghosh, Debabrota Basu, Huazhu Fu, Yuan Wang, Renuga Kanagavelu, Jin Peng Jiang, Yong Liu, Rick Siow Mong Goh, and Qingsong Wei. History-aware and dynamic client contribution in federated learning. In *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) Proceedings*, pp. 1374–1381. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2025.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

Wenwen He, Wenke Huang, Bin Yang, Shukan Liu, and Mang Ye. Spmc: Self-purifying federated backdoor defense via margin contribution. In *ICML*, 2025.

Pingbang Hu, Joseph Melkonian, Weijing Tang, Han Zhao, and Jiaqi Ma. Grass: Scalable data attribution with gradient sparsification and sparse projection. *Advances in Neural Information Processing Systems*, 38:43880–43904, 2026.

Xiaoqian Jiang and Jing Zhang. Fedclean: A general robust label noise correction for federated learning. In *Forty-second international conference on machine learning*, 2025.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 06–11 Aug 2017.

K Naveen Kumar, Ranjeet Ranjan Jha, C Krishna Mohan, and Ravindra Babu Tallamraju. Fortifying federated learning towards trustworthiness via auditable data valuation and verifiable client contribution. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4999–5009. IEEE, 2025.

Wenqian Li, Shuran Fu, Fengrui Zhang, and Yan Pang. Data valuation and detections in federated learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12027–12036. IEEE, 2024.

Chuang Liu, Hongyan Xu, Yichao Cao, Xiu Su, Zhe Qu, Tianfa Li, Shan An, and Haogang Zhu. Tinymig: Transferring generalization from vision foundation models to single-domain medical imaging. In *ICML*, 2025a.

Junkang Liu, Yuanyuan Liu, Fanhua Shang, Hongying Liu, Jin Liu, and Wei Feng. Improving generalization in federated learning with highly heterogeneous data via momentum-based stochastic controlled weight averaging. In *Forty-second International Conference on Machine Learning*, 2025b.

Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on intelligent Systems and Technology (TIST)*, 13(4):1–21, 2022.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Christian Matek, Simone Schwarz, Karsten Spiekermann, and Carsten Marr. Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks. *Nature Machine Intelligence*, 1(11):538–544, 2019.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 23–29 Jul 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 33: 19920–19930, 2020.

Ittai Rubinstein and Samuel Hopkins. Rescaled influence functions: Accurate data attribution in high dimension. *Advances in Neural Information Processing Systems*, 38:150915–150944, 2026.

Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63 (7):1455–1462, 2016. doi: 10.1109/TBME.2015.2496264.

Weiwei Sun, Haokun Liu, Nikhil Kandpal, Colin Raffel, and Yiming Yang. Enhancing training data attribution with representational optimization. *Advances in Neural Information Processing Systems*, 38:102274–102311, 2026.

Yue Tan, Guodong Long, Jing Jiang, and Chengqi Zhang. Influence-Oriented Personalized Federated Learning. *arXiv e-prints*, art. arXiv:2410.03315, October 2024. doi: 10.48550/arXiv.2410.03315.

Guojun Tang, Jiayu Zhou, Mohammad Mamun, and Steve Drew. Lightweight and robust federated data valuation. In *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 2320–2329. IEEE, 2025.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

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 (CVPR) Workshops*, pp. 3369–3378, June 2026.

Jiachen Tianhao Wang, Dawn Song, James Y Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal dependence of training data influence. In *International conference on learning representations*, volume 2025, pp. 11236–11273, 2025.

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. *Advances in Neural Information Processing Systems*, 34:16104–16117, 2021.

Yihao Xue, Chaoyue Niu, Zhenzhe Zheng, Shaojie Tang, Chengfei Lyu, Fan Wu, and Guihai Chen. Toward understanding the influence of individual clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10560–10567, 2021.

Ziyuan Yang, Yingyu Chen, Zhiwen Wang, Hongming Shan, Yang Chen, and Yi Zhang. Patient-level anatomy meets scanning-level physics: Personalized federated low-dose ct denoising empowered by large language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5154–5163. IEEE, 2025.

Dewen Zeng, Wenlong Tian, Haozhao Wang, Jianfeng Lu, Weijun Xiao, and Zhiyong Xu. Ripple shapley: Data influence attribution in one federated training run. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28085–28093, 2026.

Lin Zhang, Lixin Fan, Yong Luo, and Ling-Yu Duan. Intrinsic performance influence-based participant contribution estimation for horizontal federated learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(6):1–24, 2022.

Xiangtao Zhang, Sheng Li, Ao Li, Yipeng Liu, Fan Zhang, Ce Zhu, and Le Zhang. Subspace constraint and contribution estimation for heterogeneous federated learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20632–20642. IEEE, 2025.

Tongtian Zhu, Wenhao Li, Can Wang, and Fengxiang He. Dice: Data influence cascade in decentralized learning. In *International Conference on Learning Representations*, volume 2025, pp. 69007–69042, 2025.

APPENDIX

This appendix provides additional methodological details, an extended discussion of related work, and supplementary experimental results omitted from the main paper due to space constraints.

A RELATED WORK

Client contribution and data valuation in FL. A large body of work studies how to quantify the contribution of clients or datasets in FL. Shapley-style methods, including GTG-Shapley and Ripple Shapely, estimate participant value by comparing the utility of different client coalitions or reconstructed models Liu et al. (2022); Zeng et al. (2026). Other approaches estimate client utility from optimization signals, such as gradients, validation performance, or update geometry. FedIF estimates client contributions using client updates and validation gradients Tang et al. (2025), while data-free gradient-entropy methods use statistics of client gradients to evaluate contribution without accessing raw data Ukaye et al. (2026). History-aware and dynamic contribution methods further account for the temporal variation of client value across rounds Ghosh et al. (2025), and influence-oriented or intrinsic-performance methods study how individual clients affect the learned model or personalized FL behavior Xue et al. (2021); Zhang et al. (2022); Tan et al. (2024). These methods are valuable for participant assessment, reward allocation, robustness, or aggregation control. However, they mainly operate at the client or dataset level and do not identify which individual samples within an influential client shaped the corresponding local model behavior.

Training-data attribution for individual samples. Training-data attribution methods aim to estimate how individual training examples affect a model's predictions or losses. Classical influence functions estimate the effect of upweighting or removing a training point through local sensitivity at an optimum Koh & Liang (2017), but their assumptions can be restrictive for modern neural networks trained with nonconvex optimization and early stopping. TracIn traces gradient similarity along saved checkpoints to estimate influence accumulated during training Pruthi et al. (2020), while TRAK scales attribution to large models using randomized projections and surrogate linearization Park et al. (2023). More recent work improves attribution through representational optimization or accumulative SGD-based influence estimation Sun et al. (2026). These methods provide sample-level explanations, but they are generally formulated for centralized training and do not explicitly model the FL hierarchy in which samples first shape local client models and client updates then affect the global model through aggregation.

Temporal influence and approximate unrolling. Several methods address the fact that influence depends on the training trajectory rather than only on the final model. Trajectory-based differentiation and related methods can capture how a training example affects later optimization states, but exact unrolling can be expensive in storage and computation Pruthi et al. (2020); Wang et al. (2025). SOURCE reduces this cost by approximating the unrolled training dynamics with segment-level gradient and curvature statistics, providing a practical estimator for finite-horizon training influence beyond fully converged models Bae et al. (2024). This is closely related to the Level 2 component of FedHIT. However, SOURCE is not designed as a federated diagnostic framework: it does not first localize query-relevant client-round updates at the global level, nor does it distinguish client-update attribution from sample attribution within a selected local trajectory.

Federated influence propagation. Recent work has begun to study data influence across federated training dynamics. Ripple Shapley explicitly models how sample influence can propagate across FL rounds Zeng et al. (2026), while DICE studies influence cascades in decentralized learning Zhu et al. (2025). These approaches are important for understanding end-to-end propagation, where the goal is to estimate how data effects travel through multiple communication rounds or decentralized interactions. FedHIT takes a different diagnostic stance. Rather than propagating a sample's effect through all subsequent rounds to estimate its influence on the final global model, FedHIT separates the attribution targets: Level 1 explains query-specific global behavior using client-round updates, and Level 2 explains the selected client's local-model behavior using its training samples. This avoids treating local sample scores as direct global influence estimates and makes the resulting interpretation more localized and inspection-oriented.

Why FedHIT does not focus on Shapley valuation. Shapley-based methods are well suited for estimating the contribution of clients, data sources, or samples to an overall model utility, often

by averaging marginal contributions across coalitions Liu et al. (2022); Zeng et al. (2026). This is useful for valuation, reward allocation, and fairness. Our goal is different. FedHIT is designed for query-specific and round-specific diagnosis: given a particular query, it asks which client-round updates are positively or negatively associated with the queried global-model behavior, and then which local samples within those selected clients support or oppose the corresponding local-model behavior. This requires signed, gradient-based attribution at two distinct levels rather than a single utility-based valuation score. Consequently, FedHIT emphasizes *per-query, per-round* influence tracing and targeted sample inspection instead of estimating a Shapley value over all data sources or propagating sample values end-to-end through the federation.

Positioning of FedHIT. FedHIT is complementary to prior work but addresses a distinct setting. Client-valuation methods such as FedIF, GTG-Shapley, and gradient-based reward methods assess which clients are valuable to the federation Tang et al. (2025); Liu et al. (2022); Xu et al. (2021). Sample-attribution methods such as influence functions, TRAK, TracIn, and SOURCE estimate how individual examples affect model behavior Koh & Liang (2017); Park et al. (2023); Pruthi et al. (2020); Bae et al. (2024). End-to-end federated influence methods such as Ripple Shapley and DICE study how data effects propagate across rounds or decentralized interactions Zeng et al. (2026); Zhu et al. (2025). In contrast, FedHIT introduces a query-driven hierarchical workflow:

global query $\rightarrow$ *influential client-round update* $\rightarrow$ *influential samples within the selected client's local trajectory*.

This formulation combines client localization and local sample inspection while deliberately preserving their distinct attribution targets.

B FedHIT Algorithm

Algorithm 1 depicts the overall proposed worksflow of FedHIT. Top-k client selection is performed independently for each communication round. Level-2 attribution is performed independently within each selected client-round pair; samples from different clients are not jointly ranked or compared. Ties in Top-k selection are broken by client index.

Interpretation of the two levels. FedHIT provides a hierarchical attribution view of a query by separating client-level localization from local sample-level inspection:

- **Level 1: Where did the influence come from?** Level 1 evaluates the participating clients for the given query and identifies the client-round updates most associated with the queried global-model behavior.
- **Level 2: Which local data contributed to that influence?** For each selected client-round pair, the same query is passed to the corresponding client, where Level 2 identifies the local training samples whose estimated contributions to the client's training trajectory support or oppose the queried behavior.

FedHIT connects global client-level attribution with local sample-level attribution without merging them into a single end-to-end influence score. At Level 1, the server uses the federated updates to identify influential client-round pairs for the query. At Level 2, each selected client performs sample-level attribution along its local training trajectory using the approximate unrolling formulation of SOURCE. Thus, Level 1 determines *where* the query-relevant influence arises, while Level 2 determines *which local samples* shape the selected client's contribution. The Level 2 attribution remains specific to the selected client-round pair and does not imply a globally comparable ranking of samples across different clients or rounds.

C Extended Preliminaries

Federated learning setup. We consider a federated learning system with K clients and a coordinating server. At communication round r, the participating clients form the set $\mathcal{P}^{(r)}$. Each client $i \in \mathcal{P}^{(r)}$ holds a local dataset $\mathcal{D}_i^{(r)}$ and trains a local model initialized from the current global model $\theta^{(r)}$. The client returns an update $\Delta\theta_i^{(r)}$, which is aggregated by the server using FedAvg weights

Algorithm 1 FedHIT: Federated Hierarchical Influence Tracing

Require: $\{\theta^{(r)}\}_{r=0}^{R-1}, \{P^{(r)}\}_{r=0}^{R-1}, \{(\alpha_i^{(r)}, \Delta\theta_i^{(r)}) : i \in P^{(r)}\}_{r=0}^{R-1}$ recorded during FL training

Input: query z_q (initially available at the server); routing budget k; segment count L
Output: $\mathrm{Prop}(z_q)$, $\mathrm{Opp}(z_q)$
$\mathrm{Prop}(z_q) \leftarrow \emptyset, \quad \mathrm{Opp}(z_q) \leftarrow \emptyset$
for $r = 0, 1, \ldots, R-1$ **do**
  // Server: Level 1 - client-level routing
  $g_q^{(r)} \leftarrow \nabla_\theta \, \ell(z_q; \theta^{(r)})$
  for each $i \in P^{(r)}$ **do**
    $s_i^{(r)}(z_q) \leftarrow -\alpha_i^{(r)} (\Delta\theta_i^{(r)})^\top g_q^{(r)}$
  end for
  $C_k^{(r)}(z_q) \leftarrow \mathrm{TopK}_{i \in P^{(r)}} \, s_i^{(r)}(z_q)$
  for each $c \in C_k^{(r)}(z_q)$ **do**
    send z_q to c
    // Client: Level 2 -sample-level attribution within c
    retrieve the recorded local trajectory $\vartheta_{c,0}^{(r)} = \theta^{(r)} \to \cdots \to \vartheta_{c,T}^{(r)}$; retrieve $D_c^{(r)}$, set $n_c = |D_c^{(r)}|$
    partition into L segments $0 = T_0 < \cdots < T_L = T$, $K_\ell = T_\ell - T_{\ell-1}$, with checkpoints $\{\vartheta_{\ell,a}\}_{a=1}^{C_\ell}$
    for $\ell = 1, \ldots, L$ **do**
      compute $\bar{\eta}_\ell$ and the EK-FAC curvature $\bar{H}_\ell$ for segment ℓ
      $\bar{S}_\ell \leftarrow \exp(-\bar{\eta}_\ell K_\ell \bar{H}_\ell)$
    end for
    for each $z_j \in D_c^{(r)}$ **do**
      for $\ell = 1, \ldots, L$ **do**

$$\bar{g}_\ell(z_j) \leftarrow \frac{1}{C_\ell} \sum_{a=1}^{C_\ell} \nabla_\vartheta \, \ell(z_j; \vartheta_{\ell,a})$$

$$\bar{r}_\ell(z_j) \leftarrow \frac{1}{n_c} \, F_r(\bar{H}_\ell) \, \bar{g}_\ell(z_j), \quad F_r(\sigma) = \frac{1 - \exp(-\bar{\eta}_\ell K_\ell \sigma)}{\sigma}$$

      end for

$$\tau_c^{(r)}(z_q, z_j) \leftarrow \nabla_\vartheta \ell\Big(z_q, \vartheta_{c,T}^{(r)}\Big)^\top \left[\sum_{\ell=1}^{L} \left(\prod_{\ell'=L}^{\ell+1} \bar{S}_{\ell'}\right) \bar{r}_\ell(z_j)\right]$$

      if $\tau_c^{(r)}(z_q, z_j) > 0$ **then**
        $\mathrm{Prop}(z_q) \leftarrow \mathrm{Prop}(z_q) \cup \{(c, r, z_j, \tau_c^{(r)}(z_q, z_j))\}$
      else if $\tau_c^{(r)}(z_q, z_j) < 0$ **then**
        $\mathrm{Opp}(z_q) \leftarrow \mathrm{Opp}(z_q) \cup \{(c, r, z_j, \tau_c^{(r)}(z_q, z_j))\}$
      end if
    end for
  end for
end for
return $\mathrm{Prop}(z_q), \mathrm{Opp}(z_q)$

$\alpha_i^{(r)}$. Thus, $\alpha_i^{(r)} \Delta\theta_i^{(r)}$ represents the weighted contribution of client i to the global update at round r.

Query-conditioned attribution. FedHIT studies attribution with respect to a query z_q, which may represent an individual test example or a small evaluation subset. The query fixes the model behavior being explained. Level 1 uses the query gradient at the global model to score client-round updates, while Level 2 uses the same query to inspect samples within the selected local client trajectory. This makes all attribution scores query-specific and round-specific.

Two attribution targets. FedHIT separates two related but distinct targets. Level 1 estimates how a client-round update affects query-specific global-model behavior. Level 2 estimates how individual

Table 4: Summary of notation used in FedHIT.

Notation	Description
n	Number of clients in the federation
K	Number of clients used in an experimental federation setting
r	Communication round index
R	Total number of communication rounds
$\mathcal{P}^{(r)}$	Set of clients participating in round r
$\mathcal{D}_i^{(r)}$	Local dataset of client i at round r
$n_i^{(r)}$	Number of training samples in $\mathcal{D} * i^{(r)}$
$\theta^{(r)}$	Global model parameters at the start of round r
$\theta_i^{(r)}$	Locally trained parameters returned by client i in round r
$\Delta\theta_i^{(r)}$	Client update, $\theta_i^{(r)} - \theta^{(r)}$
$\alpha_i^{(r)}$	FedAvg aggregation weight of client i in round r
z_q	Attribution query, e.g., a held-out image or evaluation example
$\ell(z;\theta)$	Loss of model parameters θ on example z
$s_i^{(r)}(z_q)$	Level 1 per-round influence score of client i for query z_q
$S_i(z_q)$	Cumulative Level 1 client score for query z_q across rounds
$\mathcal{C}_k^{(r)}(z_q)$	Top-k routed clients in round r for query z_q
$\mathcal{Q}(z_q)$	Set of routed client-round pairs selected for Level 2 analysis
c	Selected client routed to Level 2
T	Number of local optimization steps within a routed client round
L	Number of local trajectory segments used in Level 2 attribution
K_ℓ	Length of segment ℓ in local optimization steps
$\bar{H}_\ell$	Segment-averaged EK-FAC curvature approximation
$\bar{g}_{\ell,j}$	Segment-averaged gradient of training sample z_j
$\bar{S}_\ell$	Segment propagation operator for perturbations
$\bar{r}_{\ell,j}$	Estimated injected effect of sample z_j in segment ℓ
h_q	Query gradient at the local endpoint
$\tau_c^{(r)}(z_q, z_j)$	Level 2 sample influence score for sample z_j in routed pair (c, r)
M_0	Original query margin before removal

samples affect the selected client's local model along its local training trajectory. Level 2 scores are not propagated through later aggregation rounds and should not be interpreted as direct sample influence on the final global model.

Table 4 provides a summary of the notations used throughout this paper.

D Extended Methodological Details

Level 1 client attribution. For each query and communication round, FedHIT scores participating clients by aligning their FedAvg-weighted updates with the query-loss gradient. Positive scores indicate updates estimated to improve the queried behavior, while negative scores indicate updates estimated to oppose it. The resulting ranking identifies the client-round pairs most relevant to the query and determines which clients are routed to Level 2.

Level 2 sample attribution. For each selected client-round pair, FedHIT applies SOURCE-style approximate unrolling to the client's local training trajectory. The local trajectory is segmented,

and segment-level gradient and curvature statistics are used to approximate the effect of individual samples on the selected client's local endpoint. These scores identify positively and negatively influential samples within the selected client, but they remain local to that client-round pair.

Routing rule. Routing is performed independently for each query and round. Given a routing budget k, FedHIT selects the k participating clients with the highest signed Level 1 scores. The default setting uses $k = 1$, while larger k values widen the search. Routing by signed score prioritizes clients estimated to support the query; analyses focused on harmful behavior can also examine the negative tail or rank by absolute influence.

D.1 Computational Complexity Analysis

Setup. We count the extra computation needed to attribute a single query z_q across all R communication rounds, which is also the setting of the FLOP measurements in Tables 2, 3. Federated training itself is excluded from both pipelines, but everything that attribution triggers is counted, including FedHIT's trajectory replay. Throughout, a multiply-add counts as two FLOPs. The $\Theta(\cdot)$ and $O(\cdot)$ expressions describe how cost scales; the GFLOP values in the tables come from an operation-level counter with its own constants, and the expressions below are not meant to reproduce them term by term. Whenever we vary K, N or $n_i^{(r)}$, we hold E, L, C, $M_{\max}$, the architecture and the per-example costs fixed, and drop terms of lower order in the varied quantity.

Beyond the notation of Table 4, the analysis uses a few extra quantities:

- **Data.** N is the total number of training samples. With full participation every round covers the whole training set, so $\sum_i n_i^{(r)} = N$; with equal shards, $n_i^{(r)} = \bar{n} = N/K$.
- **Per-example costs.** d is the number of attributed parameters, and G and G_f are the FLOPs of one forward-backward pass and one forward pass on a single example. The forward pass touches every attributed weight in at least one multiply-add, so under this FLOP convention $G \geq 2d$.
- **Local training and checkpoints.** E is the number of local epochs, so the T local steps of a round process roughly $E\, n_c^{(r)}$ examples. Level 2 uses C_ℓ checkpoints in segment ℓ and $C = \sum_\ell C_\ell$ in total; these are attribution checkpoints, not optimizer steps. Curvature at each checkpoint is fitted on $m_c = \min(M_{\max}, n_c^{(r)})$ examples.
- **Routing budgets.** FedHIT expands k clients per round, giving the routed pairs $\mathcal{Q}(z_q) = \{(c, r) : c \in \mathcal{C}_k^{(r)}(z_q)\}$, so $|\mathcal{Q}(z_q)| = Rk$ under full participation. The reference expands k_{exh} clients per round, collected in $\mathcal{C}_{\text{exh}}^{(r)}$. All experiments use $k = k_{\text{exh}} = 3$.
- **EK-FAC costs.** For a layer j with input and output dimensions (a_j, b_j) and s_j spatial positions ($s_j = 1$ for fully connected layers), we write

$$\Gamma_{\text{cov}} = \sum_j s_j(a_j^2 + b_j^2), \qquad \Gamma_{\text{rot}} = \sum_j a_j b_j (a_j + b_j), \qquad \Gamma_{\text{eig}} = \sum_j (a_j^3 + b_j^3)$$

  for the per-example cost of accumulating the Kronecker-factor covariances, the cost of rotating one vector into or out of the Kronecker eigenbasis, and the cost of the eigendecompositions, respectively.

In full generality, FedHIT's cost for one query is

$$\sum_{r=1}^{R} \Big[G + |\mathcal{P}^{(r)}|\, d \Big] \; + \sum_{(c,r) \in \mathcal{Q}(z_q)} \Big(n_c^{(r)} C G + E\, n_c^{(r)} G + \Omega_c \Big),$$

where Ω_c is a per-pair overhead defined below. With full participation, a fixed routing budget and equal shards, this reduces to the simpler expressions that follow.

Level 1: client routing. In each round the server computes one query gradient, $\nabla_\theta \ell(z_q; \theta^{(r)})$, at cost G, and scores every client with

$$s_i^{(r)}(z_q) = -\alpha_i^{(r)} \big(\Delta\theta_i^{(r)}\big)^\top \nabla_\theta \ell(z_q; \theta^{(r)}),$$

which takes $2Kd$ FLOPs. It then keeps the top k clients at cost $O(K)$. If routing uses the cumulative scores $S_i(z_q)$ instead, accumulating them adds only $O(RK)$. Hence

$$\mathcal{C}_{L1} = \Theta\big(R\,(G + Kd)\big). \tag{4}$$

Because $s_i^{(r)}(z_q)$ is a first-order estimate of how client i's update moves the query loss, screening a client costs one inner product against an update the server already holds. No model evaluation or training data is involved.

Level 2: sample attribution within routed clients. Level 2 runs only for the pairs $(c, r) \in \mathcal{Q}(z_q)$. It uses the operators $\bar{H}_\ell$, $\bar{S}_\ell$ and F_r from Algorithm 1, all represented in an EK-FAC eigenbasis. On the query side, $h_q = \nabla_\theta f(z_q; \theta_c^{(r)})$ is the gradient of the query measurement f at the local endpoint; on the training side, $\bar{g}_{\ell,j}$ is the checkpoint-averaged gradient of the training loss ℓ on sample z_j. Level 1 thus routes with the query loss, while Level 2 attributes the query measurement. With $\bar{r}_{\ell,j} = \frac{\alpha_c^{(r)}}{n_c^{(r)}} F_r(\bar{H}_\ell)\,\bar{g}_{\ell,j}$, the score is

$$\begin{aligned} \tau_c^{(r)}(z_q, z_j) &= h_q^\top \sum_{\ell=1}^{L} \bar{S}_L \bar{S}_{L-1} \cdots \bar{S}_{\ell+1}\,\bar{r}_{\ell,j} = \sum_{\ell=1}^{L} \big\langle v_\ell,\ \bar{g}_{\ell,j} \big\rangle, \\ v_\ell &= \frac{\alpha_c^{(r)}}{n_c^{(r)}}\, F_r(\bar{H}_\ell)^\top\, \bar{S}_{\ell+1}^\top \cdots \bar{S}_L^\top\, h_q, \end{aligned} \tag{5}$$

where the product is the identity when $\ell = L$. The second equality needs only linearity: each operator moves from the sample side to the query side by transposition, which reverses the order of the product. Under SGD-type local training the operators are symmetric, so the transposes are the operators themselves; with preconditioned optimizers they are not, but under EK-FAC the transposes cost the same to apply.

The vectors v_ℓ are computed once per routed pair by a backward recursion of $2L - 1$ operator applications, so during scoring the curvature operators never touch individual training samples; each sample enters only through its checkpoint gradients. The scalar $\alpha_c^{(r)}/n_c^{(r)}$ is a constant-factor multiplication and does not affect the asymptotic cost.

We count costs per example. Batching changes hardware utilization and wall-clock time, not the operation count, and the implementation counter follows the same convention. A routed pair has five leading costs:

1. **Trajectory replay:** $\Theta(E\,n_c^{(r)} G)$, discussed below.
2. **EK-FAC construction:** a covariance pass and a corrected-eigenvalue pass at every checkpoint, each over m_c examples, plus one eigendecomposition of the checkpoint-averaged factors per segment. As a compact bound, $\Omega_{\mathrm{EK},c} = O\big(C\,m_c\,(G + \Gamma_{\mathrm{cov}} + \Gamma_{\mathrm{rot}}) + L\,\Gamma_{\mathrm{eig}}\big)$.
3. **Training gradients:** C checkpoint gradients per sample, or $n_c^{(r)}\,C\,G$ in total.
4. **Query propagation:** computing h_q and running the recursion in equation 5, at a cost of $G + O(L\,\Gamma_{\mathrm{rot}})$.
5. **Scoring:** one inner product per checkpoint gradient, or $O(n_c^{(r)}\,C\,d) \le O(n_c^{(r)}\,C\,G)$.

The first three items do not depend on the query. If a later query is routed to the same client-round pair, their outputs can be reused, and only Level 1 and items 4-5 are repeated.

Let $\Omega_c = G + L\,\Gamma_{\mathrm{rot}} + \Omega_{\mathrm{EK},c}$. When $n_c^{(r)} \ge M_{\max}$, we have $m_c = M_{\max}$, so Ω_c no longer depends on shard size and we write it simply as Ω. When $n_c^{(r)} < M_{\max}$, the covariance part of Ω_c grows linearly with $n_c^{(r)}$. For $\bar{n} \ge M_{\max}$, with full participation, a fixed routing budget and equal shards, the Level-2 cost with replay (our implementation) and with stored checkpoints is

$$\mathcal{C}_{L2} = \Theta\big(R\,k\,[\,\bar{n}\,C\,G + E\,\bar{n}\,G + \Omega\,]\big), \qquad \mathcal{C}_{L2}^{\mathrm{stored}} = \Theta\big(R\,k\,[\,\bar{n}\,C\,G + \Omega\,]\big), \tag{6}$$

and $\mathcal{C}_{\mathrm{FedHIT}} = \mathcal{C}_{L1} + \mathcal{C}_{L2}$. In both regimes the Level-2 cost is $\Theta(n_c^{(r)})$: the $n_c^{(r)}CG$ term gives a linear lower bound, and no other term grows faster than linearly.

Trajectory replay. Our implementation does not save intermediate checkpoints during FL training. Instead, the routed client rebuilds its round-r trajectory by deterministically replaying local training from $\theta^{(r)}$, and checks that the replayed endpoint matches $\theta_c^{(r)}$ up to numerical tolerance, which confirms that the replay reproduces the submitted update $\Delta\theta_c^{(r)}$. All clients are simulated in a single process in our experiments, so where this computation runs has no effect on the FLOP counts. Because the checkpoints are rebuilt on demand, replay is attribution-time work, and Tables 2, 3 include it. Storing checkpoints would remove the replay term; that would change the coefficient of the linear term, and with it the crossover point below, but not the gap between linear and quadratic scaling.

LOCO + sample-LOO reference. The reference, labelled "Exhaustive" in Tables 2, 3, does not use FedHIT's Level-2 estimator; its second stage retrains the model explicitly for each candidate sample. It first applies round-local LOCO to every participating client. Our implementation builds all K counterfactual aggregates by multiplying the $K \times d$ matrix of client updates by a dense $K \times K$ weighting matrix, at a cost of $2K^2d$ FLOPs, and then runs $K+1$ query forward passes: one on the full aggregate and one on each counterfactual. The construction could be reduced algebraically to $O(Kd)$, but our FLOP counts reflect the dense version that was actually executed, so the tables describe the implemented reference rather than the best possible LOCO. For the k_{exh} clients in $\mathcal{C}_{\text{exh}}^{(r)}$ that LOCO ranks highest, the reference then runs retraining-based sample-level leave-one-out under the same federated training protocol. Each candidate $z_j \in D_c^{(r)}$ requires retraining client c from $\theta^{(r)}$ on $D_c^{(r)} \setminus \{z_j\}$, which costs $\Theta(E\, n_c^{(r)} G)$ to leading order, or $\Theta\big(E\,(n_c^{(r)})^2 G\big)$ over all of the client's samples. In total,

$$\begin{aligned}\mathcal{C}_{\text{Ref}} &= \Theta\Big(\sum_{r=1}^{R}\Big[|\mathcal{P}^{(r)}|^2 d + |\mathcal{P}^{(r)}|\, G_f + E\, G \sum_{c\in\mathcal{C}_{\text{exh}}^{(r)}} (n_c^{(r)})^2\Big]\Big) \\ &= \Theta\Big(R\Big[K^2 d + K\, G_f + k_{\text{exh}}\, E\, G\, \bar{n}^2\Big]\Big) \qquad \text{(fixed } K,\ k_{\text{exh}}\text{; equal shards).}\end{aligned} \tag{7}$$

The per-candidate re-aggregation and query evaluations, $O(n_c^{(r)}(d+G_f))$ per client, are lower order than the retraining term and are dropped.

Comparison with the reference. Both methods expand the same number of clients at Level 2, though not necessarily the same clients. The difference at Level 2 therefore reflects only the attribution estimator, and routing contributes separately at Level 1. At Level 1, FedHIT replaces a counterfactual model evaluation per client with an inner product: $\Theta(G + Kd)$ per round against $\Theta(K^2d + KG_f)$ for the implemented LOCO. At Level 2, with equal shards and ignoring lower-order terms, the leading-order ratio is

$$\frac{\mathcal{C}_{\text{Ref},L2}}{\mathcal{C}_{\text{FedHIT},L2}} = \frac{k_{\text{exh}}\, E\, \bar{n}^2\, G}{k\,\big(\bar{n}\,(C+E)\,G + \Omega\big)} \xrightarrow{\bar{n}\to\infty} \frac{k_{\text{exh}}}{k}\cdot\frac{E\,\bar{n}}{C+E}. \tag{8}$$

The factor $C+E$ is an effective per-sample coefficient that combines the checkpoint gradients with the replay; it is not a number of checkpoints, and it becomes C when checkpoints are stored. In plain terms, the reference retrains the client once for every candidate sample, whereas FedHIT computes C gradients per sample plus a single replay per routed client, and to leading order in $n_c^{(r)}$ that replay costs the same as one of the reference's retrainings. For large shards, the ratio therefore grows linearly with $\bar{n}$.

Within this leading-term model, where one forward-backward pass costs G in both methods, take $k = k_{\text{exh}}$ and $\bar{n} \geq M_{\max}$. FedHIT's Level 2 is then the cheaper one whenever

$$\bar{n} > \bar{n}^\star = \frac{(C+E) + \sqrt{(C+E)^2 + 4E\,\Omega/G}}{2E}. \tag{9}$$

At the operation level, FedHIT's Level 1 costs $G + 2Kd$ per round and the implemented LOCO costs $2K^2d + (K+1)G_f$. Ignoring $O(K)$ scalar work, FedHIT's Level 1 is cheaper exactly when $G - (K+1)\,G_f < 2K(K-1)\,d$. This holds in every configuration we report: 5.3-6.0 GFLOPs for FedHIT against 41.6-227.0 GFLOPs for LOCO (Table 3). When both conditions hold, the full FedHIT pipeline is cheaper than the reference under this cost model.

Consistency with the measured FLOPs. Going from $K = 10$ to $K = 50$ makes every shard five times smaller.

- The reference's Level-2 cost falls by 22-26× across the three datasets, close to the 25× that its quadratic dependence on shard size predicts.
- FedHIT's Level-2 cost falls by only 1.3-2.1×, well short of the 5× a purely linear cost would give. This is consistent with a large share of its cost coming from the shard-independent overhead Ω, and it is why the total reduction is largest at $K = 10$ (97-99%) and smaller, though still substantial, at $K = 50$ (70-84%).
- FedHIT's Level 1 barely moves, because the query gradient G outweighs $2Kd$ for this model.
- LOCO's Level 1 grows 5.5× for a 5× increase in K, because the linear $(K + 1)G_f$ term dominates over this range despite the quadratic $2K^2d$ component. Splitting the measurements into these two parts, the quadratic term accounts for about 4% of LOCO's cost at $K = 10$ and about 18% at $K = 50$.

The coefficient recovered from that split, $2d = 16.78$ MFLOPs, agrees to within 0.1% with the per-client inner-product cost measured independently from FedHIT's Level 1, which supports the decomposition.

Wall-clock time. At $K = 10$, runtime speedups are close to the FLOP ratios: 75× against 79× on HAM10000, and 112× against 122× on Matek-19. At $K = 50$ the gap on HAM10000 is wider, 2.9× against 4.7×, which suggests that costs not captured by FLOP counting make up a larger share of runtime in that setting.

A fully exhaustive baseline. For completeness, consider sample-level LOO on every client, a baseline we do not run. Its cost is

$$\mathcal{C}_{\text{Full}} = \Theta\Big(E\,G\sum_{r=1}^{R}\sum_{i=1}^{K}(n_i^{(r)})^2\Big) \;\geq\; \Theta\Big(\frac{R\,E\,G\,N^2}{K}\Big), \tag{10}$$

with the same lower-order terms dropped as in equation 7. The bound holds round by round: since $\sum_i n_i^{(r)} = N$, Cauchy-Schwarz gives $\sum_i (n_i^{(r)})^2 \geq N^2/K$, with equality for equal shards, so any imbalance between clients makes this baseline more expensive.

Here routing matters directly. Running Level 2 on all K clients instead of k would add at least $R(K - k)(\bar{n}CG + \Omega)$ of work, while routing costs $R(G + 2Kd)$. Since $G \geq 2d$,

$$\frac{\text{routing overhead}}{\text{Level-2 saving}} \;\leq\; \frac{K+1}{(K-k)\,\bar{n}\,C}. \tag{11}$$

For our smallest shards ($\bar{n} \approx 126$ at $K = 50$ on BreakHis), this is below 1% even with a single checkpoint. When per-sample Level-2 work dominates both Ω and Level 1, the routing and estimator factors multiply:

$$\frac{\mathcal{C}_{\text{Full}}}{\mathcal{C}_{\text{FedHIT}}} \approx \underbrace{\frac{K}{k}}_{\text{routing factor}} \cdot \underbrace{\frac{E\,\bar{n}}{C+E}}_{\text{estimator factor}} = \frac{E\,N}{k\,(C+E)}, \tag{12}$$

with $C + E$ replaced by C when checkpoints are stored. Holding K, k, E, C, L, $M_{\max}$ and the architecture fixed, with equal shards, full exhaustive attribution grows as $\Theta(N^2)$, while FedHIT's cost grows linearly in N, plus terms that do not depend on N.

Where the savings come from. The two levels save computation in different ways. Routing screens all K clients with cheap inner products and runs Level 2 on only k of them per round, which controls how many clients are expanded. Inside each routed client, the Level-2 estimator replaces one retraining run per candidate sample with a single replay plus C gradients per sample, which turns a quadratic dependence on shard size into a linear one.

Storage and communication. Counting parameters only:

- **Server.** For post-hoc attribution, the server keeps $\{\alpha_i^{(r)}, \Delta\theta_i^{(r)}\}$ from every round and the round-start models $\theta^{(r)}$. Frozen weights never change, so updates are nonzero only in the

d attributed coordinates and frozen weights need to be stored once. This gives $O(RKd)$ for the updates plus $O(Rd)$ for the attributed part of the round-start models. The reference needs exactly the same records, so this is not a cost specific to FedHIT.

- **Clients.** A routed client needs its local data $D_c^{(r)}$, the round-start model $\theta^{(r)}$, and whatever state makes local training deterministic: data order, augmentation seeds, and any optimizer state carried into the round. Storing checkpoints instead would cost $O(Cd)$ per participating client per round, or $O(RKCd)$ overall, and would remove replay from attribution time.
- **Communication.** For one query, the server sends z_q to the $|\mathcal{Q}(z_q)| \leq Rk$ routed client-round pairs, and each returns $n_c^{(r)}$ attribution scores. Raw training data never leaves the clients. If a client no longer holds $\theta^{(r)}$, sending it again adds $O(d)$ per routed pair for the attributed coordinates; otherwise, attribution adds no model-parameter traffic beyond FL training itself.

E Baselines and References

Removal-based references. We use leave-one-out (LOO) for sample-level removal, where one or more training samples are excluded and the affected local model is retrained. We use leave-one-client-out (LOCO) for client-level removal, where a participating client's weighted update is removed from a given aggregation round, the remaining FedAvg weights are renormalized, and the aggregate model is recomputed while all other client updates in that round are fixed. Thus, LOCO provides a round-local counterfactual reference and does not retrain the full FL trajectory. Exhaustive attribution refers to evaluating all relevant clients or samples by direct removal rather than using FedHIT's routed approximation; it provides a useful reference but is computationally expensive.

Level 1 baseline. For Level 1, LOCO serves as the client-level counterfactual reference. For each query and round, the change in query behavior after removing one client's weighted update gives a direct round-local estimate of that client's effect. We compare FedHIT Level 1 rankings against LOCO rankings using Spearman correlation.

Level 2 baselines. For Level 2, we compare against sample-attribution methods such as influence-function-style estimators Grosse et al. (2023), TRAK Park et al. (2023), and SOURCE-based attribution Bae et al. (2024), which estimate how individual training examples affect model behavior. SOURCE is particularly relevant because FedHIT builds on its approximate unrolling principle for local sample attribution. However, these methods do not by themselves provide a federated hierarchy that first localizes the relevant client-round and then inspects samples within that client. Exhaustive sample removal and retraining provide the most direct sample-level reference, but are costly because each removal requires retraining the affected local model. FedHIT reduces this cost by using Level 1 routing to restrict Level 2 analysis to selected client-rounds.

F Evaluation Objectives and Metrics

Level 1 ranking agreement. The first objective is to test whether FedHIT recovers the same query-specific client rankings as round-local LOCO. We compute Spearman rank correlation between FedHIT Level 1 scores and LOCO counterfactual scores for each query and round, and summarize agreement across the fixed query set.

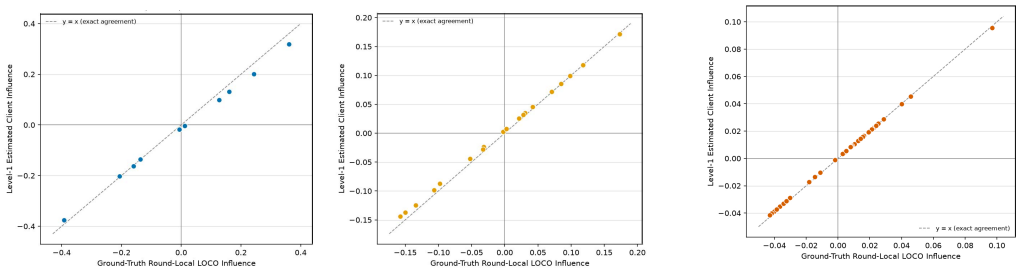
Level 2 sample attribution quality. The second objective is to test whether Level 2 identifies influential local samples within selected clients. We use LDS comparisons and sample-removal counterfactuals to evaluate whether positively and negatively attributed samples produce the expected directional changes in query behavior.

Counterfactual removal behavior. The third objective is to test whether signed attribution scores predict removal effects. At Level 1, removing helpful clients should decrease the true-class query margin, while removing harmful clients should increase it. At Level 2, removing positively attributed samples should reduce the query margin, random removal should have smaller or less systematic effects, and removing negatively attributed samples should move the margin in the opposite direction.

Computational efficiency. The final objective is to evaluate whether hierarchical routing reduces attribution cost relative to exhaustive evaluation. We report attribution GFLOPs and runtime, excluding ordinary FL training. FedHIT's efficiency comes from using a low-cost Level 1 screen to route expensive Level 2 sample attribution to a small subset of client-rounds.

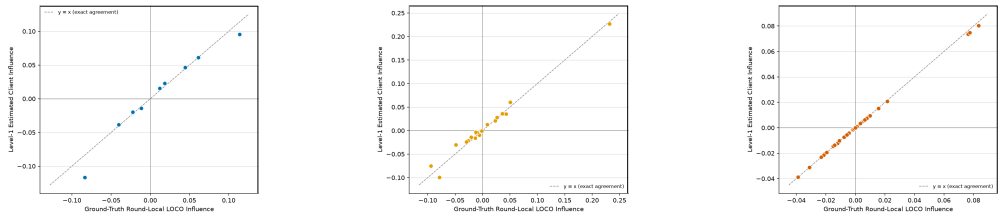
G Extended Experiments and Discussion

G.1 Level 1 Client Attribution



(a) 10 clients (b) 20 clients (c) 50 clients

Figure 4: Agreement between FedHIT Level-1 client influence estimates and the ground-truth round-local leave-one-client-out (LOCO) influence for (a) 10, (b) 20, and (c) 50 clients for Ham10K dataset. The dashed diagonal indicates exact agreement.



(a) 10 clients (b) 20 clients (c) 50 clients

Figure 5: Agreement between FedHIT Level-1 client influence estimates and the ground-truth round-local leave-one-client-out (LOCO) influence for (a) 10, (b) 20, and (c) 50 clients for BreakHis dataset. The dashed diagonal indicates exact agreement.

Figures 4 and 5 show the agreement between FedHIT Level 1 scores and round-local LOCO counterfactual scores on HAM10000 and BreakHis. Spearman correlation measures rank agreement between the FedHIT-induced client ordering and the LOCO-induced client ordering, with values closer to 1 indicating stronger agreement. The results show consistently strong rank agreement across federation sizes. On HAM10000, the mean Spearman correlations are 0.990, 0.991, and 0.992 for $K = 10$, 20, and 50 clients, respectively. On BreakHis, the corresponding correlations are 0.969, 0.972, and 0.996. These high correlations indicate that FedHIT reliably recovers the client rankings induced by query-specific, round-local LOCO across different medical datasets and client-count settings.

G.2 Counterfactual Removal Evaluation

Sample removal. At Level 2, once a relevant client is identified for a given FL round, FedHIT ranks the training samples within that client by their estimated influence on the query. We evaluate these sample-level attributions through removal and retraining. For each query, we remove 3%, 5%, or 10% of samples under three conditions: positively attributed, randomly selected, and negatively attributed samples. After each removal, the local model is retrained and the resulting prediction margin is compared with the original margin. Table 5 illustrates Level-2 counterfactual sample-removal results.

For the qualitative counterfactual illustration, we select a representative query-round example that clearly shows the attribution effect. The observed pattern is consistent across evaluated queries and rounds: removing positively attributed samples decreases the query margin, random removal produces a substantially smaller change, and removing negatively attributed samples shifts the margin in the opposite direction.

Table 5: Level 2 sample-removal counterfactual results. The table reports the original query margin and the resulting margin after removing positively attributed, randomly selected, or negatively attributed samples at 3%, 5%, and 10% removal rates. Here, M_0: Original margin. The results are shown for Matek-19 dataset.

M_0	**Removal Type**	**3% Removal**	**5% Removal**	**10% Removal**
0.388513	Positive	-2.309813	-3.436170	-5.899446
0.388513	Random	0.535441	0.654316	0.434640
0.388513	Negative	2.528747	3.232295	5.382456

H Extended Limitations and Potential Solutions

- Level 2 scores are local to selected client-rounds and are not propagated through later aggregation rounds, so they do not directly measure final global-model influence or causality. Future work could use truncated unrolling, Jacobian approximations, or explicit causal models with defined interventions, although these introduce additional computational, storage, and privacy costs.
- Both levels rely on approximations. Level 1 omits higher-order client interactions and nonlinear aggregation effects, while Level 2 inherits SOURCE/EK-FAC stationarity and structured-curvature assumptions. Fidelity could be improved through Hessian-vector products, multi-client removal, adaptive segmentation, alternative curvature estimators, and removal-based validation, at added cost.
- Routing prioritizes the most helpful clients under signed influence and may miss strongly harmful clients. For data-quality, poisoning, or failure analysis, future studies should examine both tails of the signed influence distribution or rank clients by absolute influence.
- The present evaluation uses synchronous, full-participation FedAvg and assumes access to individual client updates and selected local attribution statistics. Future extensions should consider partial client participation, asynchronous aggregation, heterogeneous local computation, alternative aggregation rules, secure aggregation, differential privacy, client-side ranking, secure inner products, trusted execution, and privacy-budget-aware attribution.
- Additionally, current work focuses on medical image classification with ResNet-18. Future works include evaluating transformer-based foundation models and broader medical tasks, including clinical text, electronic health records, multimodal learning, segmentation, detection, and continual learning.
- FedHIT scores are query-dependent diagnostic approximations to removal counterfactuals, not causal explanations; causal claims require explicit models, defined interventions, and confounder control.