zkFU: Verifiable Federated Unlearning via Trace-Consistent Subtraction and Zero-Knowledge Proofs

Anonymous authors
Paper under double-blind review

Abstract

Federated unlearning needs both effective client removal and verifiable evidence of the removal computation. Existing methods largely optimize forgetting and utility, while direct cryptographic replay scales with the training history. We introduce **zkFU**, which combines compact zero-knowledge certification with retained-aware recovery. Its key idea is to use linear aggregation to turn committed client updates into a compact deletion relation. Trace-consistent gradient subtraction (TCGS) computes a retained-normalized seed for weighted client requests. Homomorphic commitment aggregation then reduces certification to an $O(d)$ affine statement after public aggregation, compared with $O(NTd)$ direct replay over N clients, T rounds, and d parameters. TraceTrust recovers utility through sparse trust filtering, scale calibration, and forgetting-direction-shielded repair. We prove certificate soundness, zero knowledge, and completeness, and bound the seed's mismatch with retained-only retraining. The construction supports cumulative requests and public-coefficient linear aggregation rules, including FedAvg and its common variants. Across three datasets and 11 baselines, zkFU alone ranks in the top two for both retained accuracy and KL divergence to retraining. Twenty-seed tests establish equivalence to retraining within ± 1 percentage point in mean membership-inference gap under a shadow-model attack on Fashion-MNIST and CIFAR-10. The Halo2 prototype verifies an 11M-parameter ResNet-18 certificate in 29 s. A matched Fashion-MNIST comparison projects about $930\times$ lower proving time than the implemented aggregation-only replay baseline.

1 Introduction

Federated learning (FL) (McMahan et al., 2017) trains a shared model while clients keep their raw data local. When a client requests removal, the system must compute a useful replacement model and provide evidence that the removal computation used that client's actual training contributions. Federated unlearning (FU) addresses the first problem by approximating a model trained without the client (Liu et al., 2021; Cao & Yang, 2015; Bourtoule et al., 2021). Replay, gradient correction, distillation, and repair improve the forgetting–utility trade-off (Liu et al., 2021; Halimi et al., 2022; Wu et al., 2022; Zhang et al., 2023; Chundawat et al., 2024; Pan et al., 2025). Evidence of execution remains harder to obtain: membership-inference tests, markers, and accuracy gaps assess behavior, but adaptive servers can pass them without performing the intended computation (Gao et al., 2024; Zhang et al., 2024). A verifiable removal procedure must therefore connect the deletion request to the recorded training contributions.

Zero-knowledge (ZK) proofs can establish this connection, but the choice of computation to certify determines their cost. Existing systems certify centralized unlearning, including historical-update subtraction (Eisenhofer et al., 2025), or personalized edge models (Maheri et al., 2025). Directly replaying FL aggregation creates an $O(NTd)$ proof target for N clients, T rounds, and d parameters. Historical subtraction offers an efficient starting point (Wu et al., 2022), yet updates accumulated on the original trajectory need not produce a useful model after removal. The design challenge is to obtain a compact relation tied to the training history and recover utility from the resulting correction.

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.

We propose **zkFU** by making the aggregation relation itself the certification target. With fixed FedAvg weights, the final model is the initial model plus a weighted sum of submitted updates. Client commitments bind this history during training. Trace-consistent gradient subtraction (TCGS) removes the requested clients' weighted contribution and renormalizes the retained mass, yielding a seed determined by the committed trace. Homomorphic aggregation lets the verifier combine the relevant commitments before proof verification; the circuit checks an aggregate opening and the affine seed relation. Thus, after public aggregation, certification scales with d independently of N and T. This interface supports cumulative requests and public-coefficient linear rules, including FedProx (Li et al., 2020), FedAvgM (Hsu et al., 2019), and FedNova (Wang et al., 2020).

The same trace that makes certification compact also guides recovery. TCGS reuses updates from the original trajectory, whereas retained-only retraining follows a different trajectory. We bound this mismatch and introduce TRACETRUST to adapt the trace displacement to retained data. Sparse trust filtering selects coordinates with strong trace signal relative to retained sensitivity; calibration chooses the displacement scale; shielded repair restores utility while attenuating gradients opposing that displacement. zkFU thereby combines a verifiable deletion computation with recovery evaluated for both utility and forgetting. Its gains are joint: across three datasets, it ranks in the top two for both retained accuracy and KL to retraining among the evaluated non-retraining methods.

Our contributions are:

(1) **Trace-consistent client removal.** TCGS formulates removal for weighted client sets and public-coefficient linear aggregation rules as a relation over committed training contributions.
(2) **Compact deletion certificates.** A commit-and-prove protocol uses homomorphic aggregation to certify the TCGS relation with an $O(d)$ statement after public aggregation, including cumulative requests.
(3) **Trace-guided recovery.** TraceTrust uses retained sensitivity and the trace displacement to select, calibrate, and repair the removal update.
(4) **Theory and empirical evidence.** We prove certificate security, bound trajectory mismatch, and demonstrate strong utility and retraining fidelity across three datasets, with 20-seed attack-specific equivalence tests and certificate benchmarks up to 11M parameters.

2 RELATED WORK

Federated unlearning and empirical verification. Federated unlearning removes target-client influence without full retraining. Existing methods approximate removal through update replay (Liu et al., 2021), gradient- or Newton-style erasure (Halimi et al., 2022), distillation and retained-client recovery (Wu et al., 2022; Guo et al., 2023; Zhang et al., 2023), contribution dampening or auxiliary unlearning (Chundawat et al., 2024; Fraboni et al., 2024; Gu et al., 2024), and conflict- or adapter-based repair (Deng et al., 2024; Pan et al., 2025; Zhong et al., 2025). In particular, Wu et al. (2022) derive the uniform single-client normalization underlying TCGS and use unnormalized subtraction followed by distillation. zkFU generalizes the trace relation to weighted requests and couples commitment-based certification, mismatch analysis, and retained-aware recovery. These FU methods improve the forgetting–utility trade-off but do not certify the deletion computation. VeriFi/uS2U introduces verification based on membership-inference attacks (MIAs), markers, and accuracy (Gao et al., 2024), but such behavioral tests can be gamed by adaptive servers (Zhang et al., 2024), and VERIFUL (Nguyen et al., 2026) therefore identifies verification by zero-knowledge proofs (ZKPs) as an open direction.

Certified removal and ZK proofs for machine learning. Certified data removal studies when a deletion output is close to the model trained without the removed data (Guo et al., 2020; Sekhari et al., 2021). zkFU targets client-level contributions accumulated along a FedAvg trajectory and certifies the corresponding trace-consistent subtraction. ZK proofs have been used to certify FL aggregation correctness (Wang et al., 2024), centralized retraining, amnesiac, and optimization-based unlearning (Eisenhofer et al., 2025), and personalized edge-model unlearning via curvature-aware pruning (Maheri et al., 2025). These systems certify different objects: round aggregation, centralized training/unlearning state, or a personalized pruning procedure. In contrast, zkFU uses the linearity of federated aggregation to express client removal as an affine subtraction relation over homomorphically

aggregated commitments, avoiding Hessians, architecture-specific pruning, and full training replay. Appendix I compares these systems in detail.

3 PRELIMINARIES

FedAvg trace. Let θ_0 and θ_T be the initial and final models of the original FL run. Under full participation with fixed FedAvg weights $\{a_i\}_{i=1}^N$ and $\sum_i a_i = 1$, the signed client updates Δ_i^t satisfy

$$\theta_T = \theta_0 + \sum_{t=1}^{T} \sum_{i=1}^{N} a_i \Delta_i^t. \tag{1}$$

TCGS uses this trace identity to isolate the forgotten-client contribution. Since Δ_i^t denotes the signed update submitted by client i, the identity holds for any model, loss, and local optimizer, including FedProx objectives. TCGS needs only a server rule that is linear in the submitted updates with public coefficients. Round-wise partial participation (Appendix C), FedAvgM, and FedNova have this form, and TCGS and its certificate carry over with the rule's coefficients (Appendix H.1). Assumption 1 enters only the seed-quality bound of Theorem 1, and nonlinear robust aggregators and shard retraining such as SISA (Bourtoule et al., 2021) need different relations.

Commitments and ZK proofs. Each update vector is deterministically encoded into elements of a prime field $\mathbb{F}_p$ and bound to the public audit trace with a vector Pedersen commitment:

$$\mathsf{Com}(v; r) = \sum_{j=1}^{d} v_j \mathsf{G}_j + r\mathsf{H}. \tag{2}$$

Here $v \in \mathbb{F}_p^d$ is the encoded model update vector, $r \in \mathbb{F}_p$ is commitment randomness, and $\{\mathsf{G}_j\}_{j=1}^d, \mathsf{H}$ are public group generators. Its hiding, binding, and additive homomorphism let the verifier aggregate forgotten-client update commitments into a single commitment to their trace contribution without revealing individual updates. We use Halo2 (Zcash Foundation, 2022) as the proof system and defer formal definitions to Appendix A.

4 PROBLEM FORMULATION

zkFU combines a verifiable trace-consistent deletion step with retained-aware model recovery. For a deletion request over client set $\mathcal{F}$ with $0 < a_{\mathcal{F}} < 1$, the server proves that the published seed θ_{tcgs} is the unique trace-consistent result of subtracting the forgotten clients' committed FedAvg contribution. TraceTrust produces the deployed model θ_{out} from this seed. The certificate verifies the seed relation; Theorem 1 bounds its trajectory mismatch, and experiments evaluate the recovered model's utility and forgetting.

We consider clients, a server, and an external verifier. Clients post identity-bound update commitments on an append-only board before aggregation. A training-consistency certificate binds $\theta_T - \theta_0$ to their weighted sum (Appendix B). The server may report an incorrect training endpoint or deletion seed; the verifier checks these algebraic relations against the trace. The trajectory-mismatch guarantee additionally assumes an honestly executed or independently authenticated FedAvg trajectory, with honest client updates from the corresponding shared round states. Per-round aggregation and state checks can establish this premise; the endpoint check alone certifies the sum.

zkFU preserves the standard FL privacy boundary, as the server never accesses retained clients' raw data. The TCGS certificate uses only public training states and the committed update trace, while retained-aware recovery uses locally computed retained-client summaries that can be protected with secure aggregation or differential privacy.

Formally, a seed-verifiable scheme outputs a seed and a proof $(\theta_{\mathrm{tcgs}}, \pi_{\mathrm{tcgs}})$ and satisfies completeness, soundness, zero knowledge, and seed quality. Soundness binds any accepted seed to the trace-consistent TCGS output derived in Section 6. Seed quality requires that $\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le \epsilon$, where $\theta^*_{\setminus\mathcal{F}}$ is the retained-only counterfactual model. The parameter ϵ is instantiated explicitly in Theorem 1 under Assumption 1, which bounds the trace-mismatch error as $\epsilon = O(L_\Delta \eta G T^2 a_{\mathcal{F}})$ in the stable

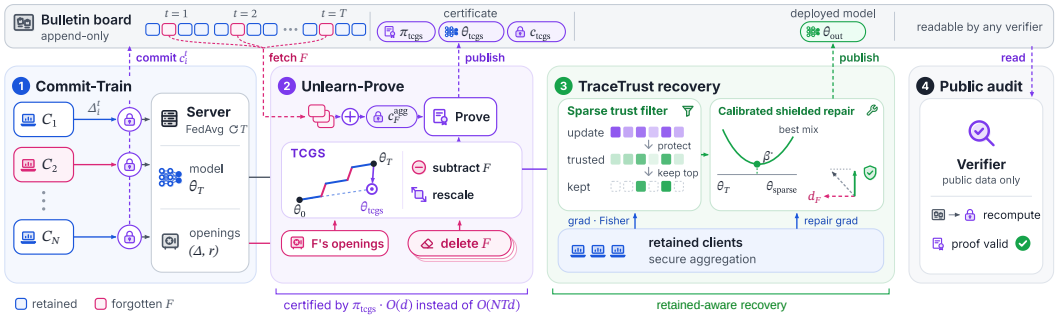

Figure 1: Overview of zkFU. (1) In Commit-Train, clients post commitments c_i^t to their updates on an append-only board, and the server aggregates the updates and keeps the openings. (2) In Unlearn-Prove, for a request $\mathcal{F}$, the server aggregates the forgotten clients' commitments into $c_{\mathcal{F}}^{\text{agg}}$, subtracts their contribution and rescales to obtain θ_{tcgs}, and proves this $O(d)$ relation with π_{tcgs}. (3) TraceTrust turns the seed into the deployed model θ_{out} using retained statistics. (4) Any verifier recomputes $c_{\mathcal{F}}^{\text{agg}}$ from the board and checks π_{tcgs}. The dashed $d_F = \theta_{\text{tcgs}} - \theta_T$ in (3) is the trace displacement used to attenuate opposing repair gradients.

local-update regime. Completeness, soundness, and zero knowledge do not depend on ϵ or on Assumption 1.

5 zkFU Protocol

zkFU follows the two-phase workflow in Figure 1. During the original FL run, each client posts a commitment $c_i^t = \mathsf{Com}(\Delta_i^t; r_i^t)$ to its signed FedAvg update on the public audit board before aggregation, and the server checks each submitted update against its commitment. This trace does not reveal client updates, but it fixes the historical contribution that can later be audited.

When a deletion request for client set $\mathcal{F}$ arrives, the server aggregates the forgotten-client commitments homomorphically, $c_{\mathcal{F}}^{\text{agg}} = \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} a_i c_i^t$, and uses the archived openings of the forgotten clients to compute the trace-consistent seed θ_{tcgs}. It then generates a proof π_{tcgs} for the affine TCGS relation and publishes $(\theta_{\text{out}}, \theta_{\text{tcgs}}, \pi_{\text{tcgs}}, c_{\text{tcgs}}, c_{\mathcal{F}}^{\text{agg}})$, where c_{tcgs} commits to θ_{tcgs}. Any verifier recomputes $c_{\mathcal{F}}^{\text{agg}}$ from the public audit trace, forms the public statement x_{tcgs}, and accepts the deletion certificate only if $\mathsf{Verify}(\mathsf{vk}, x_{\text{tcgs}}, \pi_{\text{tcgs}}) = 1$ for the verification key vk.

Appendix B gives protocol steps, commitment storage, and training-time overhead.

Sequential and selected-round requests. The k-th request is certified against the immutable original trace with the cumulative set $\mathcal{F}(k) = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$, so the circuit shape, proof size, and proof-verification cost do not depend on k. Selected-round deletion follows from the round-wise relation of Appendix C.

6 TCGS Seed and TraceTrust Recovery

The protocol uses the training trace in two ways: TCGS defines the certified removal seed, and TraceTrust adapts the associated displacement for recovery. Stage 1 computes θ_{tcgs}; Stages 2 and 3 filter, calibrate, and repair it to produce θ_{out} (Algorithm 2).

6.1 Stage 1: ZK-Verifiable Trace-Consistent Gradient Subtraction

Let $\mathcal{F}$ be the set of clients requesting removal, and in the setting of equation 1 define

$$a_{\mathcal{F}} = \sum_{i \in \mathcal{F}} a_i, \qquad W_{\mathcal{F}} = \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} a_i \Delta_i^t. \tag{3}$$

Using the FedAvg trace identity, TCGS constructs

$$\Delta_{\text{tcgs}} = \frac{a_{\mathcal{F}}(\theta_T - \theta_0) - W_{\mathcal{F}}}{1 - a_{\mathcal{F}}}, \qquad \theta_{\text{tcgs}} = \theta_T + \Delta_{\text{tcgs}}. \tag{4}$$

Here $W_{\mathcal{F}}$ removes the forgotten clients' recorded contribution, while $a_{\mathcal{F}}(\theta_T - \theta_0)$ corrects the normalization shift from all-client FedAvg weights to retained-client mass $1 - a_{\mathcal{F}}$. Thus, θ_{tcgs} is the retained-only counterfactual computed on the observed trace. Its approximation error comes from trajectory mismatch, since exact retraining would evaluate retained-client updates on a trajectory where $\mathcal{F}$ never participated.

TCGS is also compactly verifiable. During training, each update is committed as $c_i^t = \mathsf{Com}(\Delta_i^t; r_i^t)$. For a deletion request, the verifier aggregates the forgotten-client commitments as

$$c_{\mathcal{F}}^{\mathsf{agg}} = \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} a_i c_i^t, \qquad r_{\mathcal{F}}^{\mathsf{agg}} = \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} a_i r_i^t, \tag{5}$$

which opens to $\mathsf{Com}(W_{\mathcal{F}}; r_{\mathcal{F}}^{\mathsf{agg}}) = c_{\mathcal{F}}^{\mathsf{agg}}$ by homomorphism. The prover commits to the seed as $c_{\text{tcgs}} = \mathsf{Com}(\theta_{\text{tcgs}}; r_{\text{tcgs}})$ and proves that the aggregate opening and equation 4 hold. Section 7 gives the resulting $O(d)$ statement.

Assumption 1 (Smooth bounded client-update maps)**.** *For each retained client $j \notin \mathcal{F}$ and round t, the local update map is Lipschitz in the model state: $\|\Delta_j^t(\theta) - \Delta_j^t(\theta')\|_2 \le L_\Delta \|\theta - \theta'\|_2$. All client updates are bounded as $\|\Delta_i^t(\theta)\|_2 \le \eta G$ where ηG bounds the complete local update, including local-step and optimizer effects. For single-step local gradient descent with an L-smooth loss, one may take $L_\Delta = \eta L$.*

Theorem 1 (TCGS trajectory-mismatch bound)**.** *Let $\theta^*_{\setminus \mathcal{F}}$ denote the retained-only counterfactual model obtained by replaying FedAvg without clients in $\mathcal{F}$, using normalized retained weights $\bar{a}_j = a_j/(1 - a_{\mathcal{F}})$ for $j \notin \mathcal{F}$, the same initialization, and matched local randomness. Under Assumption 1, the TCGS seed satisfies $\|\theta_{\text{tcgs}} - \theta^*_{\setminus \mathcal{F}}\|_2 = O\big(L_\Delta \eta G T^2 a_{\mathcal{F}}\big)$ in the stable local-update regime $L_\Delta T = O(1)$. The error comes from trajectory mismatch, since TCGS evaluates retained-client updates on the observed trajectory, whereas exact retraining evaluates them on the retained-only trajectory.*

For single-step local gradient descent, $L_\Delta = \eta L$. For one forgotten client under uniform weights, $a_{\mathcal{F}} = 1/N$, and with $\eta = O(1/(LT))$ the bound recovers the $O(G/(LN))$ scaling. The proof is given in Appendix F.1. Thus, holding the update-map bounds fixed, the single-client mismatch bound decreases with federation size.

Beyond the stable regime. The general bound (Appendix F.1) carries a factor $e^{L_\Delta T}$, which the stable regime $L_\Delta T = O(1)$ keeps $O(1)$. Corollaries 3 and 4 remove the factor under non-expansive retained updates and make the bound polynomial in T under inverse-time stepsizes. Empirically the bound is conservative. On Fashion-MNIST the raw-seed distance falls faster than $1/N$, and trajectory probes estimate retained-map amplification of only 5 to 12 over 100 rounds on Fashion-MNIST and by less than 1 on CIFAR-10 with ResNet-18, against more than e^{30} and e^{85} in the worst case (Figure 2b,c). Appendix F.3 adds sweeps over T and η.

6.2 Stage 2: Sparse Trust Filtering

TCGS determines the historical correction, while sparse trust filtering adapts it to retained-data sensitivity. Theorem 1 controls its parameter distance to retraining, yet even a modest parameter error can affect accuracy on retained-sensitive coordinates (Table 3a). Stage 2 converts it into a retained-aware sparse update using aggregated retained statistics, including a retained-gradient estimate $g_R \approx \nabla_\theta \mathcal{L}_R(\theta_T)$ of the retained loss $\mathcal{L}_R$ and a diagonal retained-Fisher proxy $F_R = \{F_{R,k}\}_{k=1}^d$. Retained clients compute these summaries locally and aggregate them before server-side filtering, optionally through secure aggregation.

We motivate sparse trust filtering with

$$\begin{aligned} \min_{\Delta} \quad & \frac{1}{2}\|\Delta - \Delta_{\text{tcgs}}\|_2^2 + \frac{\lambda_F}{2} \sum_{k=1}^{d} F_{R,k} \Delta_k^2 \\ \text{s.t.} \quad & \langle \Delta, g_R \rangle \le 0, \qquad \|\Delta\|_0 \le \rho d. \end{aligned} \tag{6}$$

With Fisher damping weight λ_F and sparsity ratio ρ, the objective expresses the desired balance between trace fidelity, first-order retained-loss control, sensitivity damping, and sparsity.

We use a closed-form three-step surrogate. First, retained-conflict projection (RCP) attenuates the component associated with a first-order retained-loss increase:

$$\Delta_{\mathrm{proj}} = \Delta_{\mathrm{tcgs}} - \max\left(\frac{\langle \Delta_{\mathrm{tcgs}}, g_R\rangle}{\|g_R\|_2^2 + \varepsilon_g}, 0\right) g_R. \tag{7}$$

Second, Fisher trust damping (FTD) attenuates retained-sensitive coordinates:

$$\Delta_{\mathrm{trust},k} = \frac{\Delta_{\mathrm{proj},k}}{1 + \lambda_F F_{R,k}}. \tag{8}$$

Finally, Fisher-scaled trust masking (FSTM) targets the top fraction ρ of coordinates according to

$$s_k = \frac{|\Delta_{\mathrm{trust},k}|}{\varepsilon_F + F_{R,k}}, \qquad m_k = \mathbf{1}[s_k \geq \tau_\rho], \qquad \Delta_{\mathrm{sparse},k} = m_k \Delta_{\mathrm{trust},k}. \tag{9}$$

Here τ_ρ is the threshold for the target fraction, with ties retained, and $\varepsilon_g, \varepsilon_F, \varepsilon_d > 0$ are small stabilizing constants. The sparse candidate is $\theta_{\mathrm{sparse}} = \theta_T + \Delta_{\mathrm{sparse}}$. The mask favors strong trace signal relative to retained sensitivity.

6.3 Stage 3: Calibrated Shielded Repair

Stage 3 calibrates the sparse displacement and repairs retained utility while discouraging movement against the trace displacement. Trust-calibrated interpolation (TCI) searches along the path $\theta(\beta) = \theta_T + \beta(\theta_{\mathrm{sparse}} - \theta_T)$ for $\beta \in [\beta_{\min}, \beta_{\max}]$ from the original model to the sparse candidate. The scale is selected using retained or public validation signals:

$$J(\beta) = \mathrm{CE}_R(\theta(\beta)) + \lambda_{\mathrm{kd}} \mathrm{KD}_T(\theta(\beta)) + \lambda_{\mathrm{pull}}(1-\beta)^2, \tag{10}$$

and $\theta_{\mathrm{cal}} = \theta_T + \beta^\star(\theta_{\mathrm{sparse}} - \theta_T)$ with $\beta^\star = \arg\min_{\beta \in \mathcal{B}} J(\beta)$ over a finite grid $\mathcal{B} \subset [\beta_{\min}, \beta_{\max}]$. Here, with weights λ_{kd} and λ_{pull}, the retained cross-entropy term preserves retained predictive performance, the knowledge-distillation (KD) term preserves retained or public-input behavior of the teacher θ_T, and the anti-pullback term discourages collapse back to θ_T.

Starting from θ_{cal}, retained clients perform a short repair phase, forgetting-direction-shielded anchored distillation repair (FDS-ADR). Each client computes a local repair gradient on its data $\mathcal{D}_j$ from

$$\mathcal{L}^{(j)}_{\mathrm{repair}}(\theta) = \mathrm{CE}_{\mathcal{D}_j}(\theta) + \lambda_{\mathrm{kd}} \mathrm{KD}^{(j)}_T(\theta) + \lambda_{\mathrm{prox}} \|\theta - \theta_{\mathrm{cal}}\|_2^2, \tag{11}$$

with proximal weight λ_{prox}, and the server aggregates normalized retained gradients:

$$g_{\mathrm{repair}} = \sum_{j \notin \mathcal{F}} \bar{a}_j \nabla_\theta \mathcal{L}^{(j)}_{\mathrm{repair}}(\theta), \qquad \bar{a}_j = \frac{a_j}{1 - a_{\mathcal{F}}}. \tag{12}$$

To attenuate repair steps that oppose the trace displacement, we shield the repair gradient along $d_F = \theta_{\mathrm{tcgs}} - \theta_T$:

$$g_{\mathrm{shield}} = g_{\mathrm{repair}} - \lambda_{\mathrm{sh}} \max\left(\frac{\langle g_{\mathrm{repair}}, d_F\rangle}{\|d_F\|_2^2 + \varepsilon_d}, 0\right) d_F. \tag{13}$$

Here λ_{sh} is the shielding strength, and g_{shield} is supplied to the repair optimizer with stepsize η_r (Appendix G.1). Thus, Stage 3 combines retained-data repair with directional regularization toward the TCGS displacement.

7 ZK Certificate and Security

The TCGS certificate proves that the released seed is the unique trace-consistent result of subtracting the forgotten clients' committed FedAvg contribution. Given $\mathcal{F}$, the verifier recomputes $a_{\mathcal{F}}$ and $c^{\mathsf{agg}}_{\mathcal{F}}$ from the public audit trace, and the prover proves knowledge of openings $\mathsf{Com}(W_{\mathcal{F}}; r^{\mathsf{agg}}_{\mathcal{F}}) = c^{\mathsf{agg}}_{\mathcal{F}}$ and $c_{\mathrm{tcgs}} = \mathsf{Com}(\theta_{\mathrm{tcgs}}; r_{\mathrm{tcgs}})$ that satisfy equation 4. After public aggregation, the circuit contains one aggregate opening check, one seed commitment check, and coordinate-wise affine TCGS constraints.

Its size is $O(d)$ and independent of N and T, whereas direct aggregation replay has an $O(NTd)$ proof target. With one fixed-point product per retained client, round, and coordinate against two per TCGS coordinate, our aggregation-only replay baseline takes about $(N-1)T/2$ times the proving time on the same Halo2 stack (Figure 2d). The full statement, witness, constraints, and Halo2 circuit are given in Appendix A.

Theorem 2 (TCGS certificate security)**.** *Assume vector Pedersen commitments are hiding and binding under the discrete-log assumption and Halo2 is complete, knowledge-sound, and zero-knowledge. Then the TCGS certificate satisfies completeness, soundness for the trace-consistent seed, and zero knowledge. An honest prover is accepted except with negligible probability. No probabilistic polynomial-time (PPT) adversary can produce an accepting proof for an encoded seed different from the TCGS output prescribed by equation 4 in the specified arithmetic except with negligible probability. The proof reveals no retained-client openings or per-round forgotten-client openings beyond the aggregate effect implied by the public seed.*

Soundness holds because the verifier recomputes $c_{\mathcal{F}}^{\mathsf{agg}}$ from the public trace, so an accepted proof opens the trace-defined aggregate, which binding fixes, and the affine constraints determine the seed. Zero knowledge follows from Halo2 and commitment hiding (Appendix E). The public seed does determine the forgotten clients' aggregate contribution $W_{\mathcal{F}}$ through equation 4, which for a single client is its weighted sum of updates over all rounds and is released only upon that client's own deletion request. Zero knowledge ensures that the proof reveals no further witness information beyond the public statement.

Assumptions and attack surface. With authenticated identity-bound submissions, Sybil-resistant enrollment, and an append-only board, the certificate detects incorrect subtraction seeds, ex-post alteration of the trace, and reassignment of entries across identities, and injecting an uncommitted canceling pair cannot preserve the requested aggregate opening, which uses only entries indexed by $\mathcal{F}$. Proofs of training, identity enrollment, and runtime attestation are complementary layers for local computation, data provenance, and serving (Appendices E.3 and J).

8 Experiments

8.1 Experimental Setup

Datasets, models, and FL configuration. We evaluate zkFU on Fashion-MNIST (FMNIST), CIFAR-10, and CIFAR-100. Fashion-MNIST uses a lightweight convolutional neural network (CNN) with approximately 4.4×10^4 parameters, while CIFAR-10/100 use a CIFAR-adapted ResNet-18 with approximately 11M parameters. Following common client-level FU protocols (Chundawat et al., 2024; Zhang et al., 2023; Gu et al., 2024; Halimi et al., 2022; Deng et al., 2024; Gao et al., 2024; Pan et al., 2025), we evaluate single-client deletion under full-participation FedAvg, and to test generality Appendix H.1 repeats the evaluation under FedProx, FedAvgM, and FedNova. Unless otherwise stated, experiments use a Dirichlet non-IID (not independent and identically distributed) split with $\alpha = 0.5$ and $N = 20$ clients. We train for $T = 100$ rounds with $E = 5$ local epochs per round and batch size $B = 256$. Local updates use stochastic gradient descent (SGD) with learning rate 0.03, momentum 0.9, and weight decay 5×10^{-4} (Appendix G.1).

Metrics. We report test accuracy (TA), retained-client accuracy Acc_r and forgotten-client accuracy Acc_f on the clients' training data, forgotten-client gap Δ_f, MIA Gap, Kullback–Leibler (KL) divergence, and unlearning time. Federated Retrain is the reference, so $\Delta_f = |\mathrm{Acc}_f - \mathrm{Acc}_f^{\mathrm{Retrain}}|$ and KL is measured against it. MIA Gap is reported in percentage points (pp) as $100 \times |\mathrm{MIAAcc} - 0.5|$ for the accuracy MIAAcc of a shadow-model attack, with 0.5 denoting random guessing (Shokri et al., 2017; Carlini et al., 2022). Unlearning time measures computation of θ_{out} in the simulation, including calibration and repair but excluding final metric evaluation, proving, and distributed communication. Results are mean $\pm$ std over 5 trials unless stated otherwise.

Baselines. We compare against retraining references, simple trace-based variants, and representative client-level FU methods: Federated Retrain, Direct Accumulation, FedEraser (Liu et al., 2021), PGD (Halimi et al., 2022), FedRecovery (Zhang et al., 2023), ConDa (Chundawat et al., 2024), SIFU (Fraboni et al., 2024), FedAU (Gu et al., 2024), FCU (Deng et al., 2024), FedOSD (Pan et al., 2025), FUSED (Zhong et al., 2025), and uS2U (Gao et al., 2024). Direct Accumulation subtracts

Table 1: Main results at Dirichlet $\alpha = 0.5$, mean $\pm$ std over 5 trials. Acc_r is in percent; Δ_f and MIA Gap are in pp. Federated Retrain is the reference. CIFAR-10 unlearning time is normalized by zkFU. Best and second-best non-retraining means are in **red** and **light red**. denotes trace-based methods.

Methods	FMNIST				CIFAR-10					CIFAR-100			
	$\text{Acc}_r \uparrow$	$\Delta_f \downarrow$	MIA Gap $\downarrow$	KL $\downarrow$	$\text{Acc}_r \uparrow$	$\Delta_f \downarrow$	MIA Gap $\downarrow$	KL $\downarrow$	Unl. Time $\downarrow$	$\text{Acc}_r \uparrow$	$\Delta_f \downarrow$	MIA Gap $\downarrow$	KL $\downarrow$
Federated Retrain	91.83 $\pm$0.34	0.00 $\pm$0.00	1.00 $\pm$0.50	0.000 $\pm$0.000	80.20 $\pm$3.34	0.00 $\pm$0.00	0.68 $\pm$0.47	0.000 $\pm$0.000	157.61×	83.31 $\pm$2.07	0.00 $\pm$0.00	0.62 $\pm$0.14	0.000 $\pm$0.000
PGD	92.43 $\pm$0.27	2.99 $\pm$1.80	0.65 $\pm$0.48	0.046 $\pm$0.010	**93.70** $\pm$0.79	15.35 $\pm$8.19	1.02 $\pm$0.79	0.513 $\pm$0.173	3.54×	89.29 $\pm$0.79	27.00 $\pm$3.04	3.67 $\pm$0.88	0.492 $\pm$0.055
FCU	92.40 $\pm$0.19	2.28 $\pm$1.62	0.98 $\pm$0.98	0.051 $\pm$0.013	85.92 $\pm$7.06	**7.22** $\pm$6.68	0.92 $\pm$0.28	**0.261** $\pm$0.134	32.70×	86.86 $\pm$1.70	16.48 $\pm$3.12	1.02 $\pm$0.80	0.456 $\pm$0.035
FedAU	63.63 $\pm$10.66	66.69 $\pm$6.60	4.90 $\pm$3.55	1.589 $\pm$0.505	65.38 $\pm$10.60	46.67 $\pm$6.07	0.90 $\pm$0.60	0.751 $\pm$0.124	99.65×	79.83 $\pm$2.01	**4.17** $\pm$4.22	1.61 $\pm$0.87	0.595 $\pm$0.025
FedOSD	91.55 $\pm$0.42	**1.27** $\pm$1.36	0.71 $\pm$0.60	0.056 $\pm$0.024	90.50 $\pm$0.83	10.25 $\pm$8.33	1.00 $\pm$1.07	0.606 $\pm$0.175	86.30×	76.43 $\pm$1.60	9.57 $\pm$3.32	1.04 $\pm$0.56	0.686 $\pm$0.042
FUSED	92.20 $\pm$0.26	3.46 $\pm$2.25	1.45 $\pm$0.52	0.045 $\pm$0.009	91.97 $\pm$1.35	14.74 $\pm$7.59	0.56 $\pm$0.40	0.490 $\pm$0.143	2.29×	84.42 $\pm$1.82	25.29 $\pm$2.70	2.78 $\pm$1.51	0.442 $\pm$0.030
Direct Accumulation	38.41 $\pm$15.90	81.95 $\pm$7.06	1.86 $\pm$2.01	9.351 $\pm$4.937	23.84 $\pm$6.43	76.98 $\pm$8.68	1.00 $\pm$0.69	11.311 $\pm$5.380	2.50×	11.07 $\pm$2.46	57.96 $\pm$3.13	0.99 $\pm$1.14	10.738 $\pm$0.993
FedEraser	82.53 $\pm$2.23	5.01 $\pm$2.72	0.80 $\pm$0.40	0.266 $\pm$0.046	52.26 $\pm$28.78	22.00 $\pm$22.93	0.79 $\pm$0.57	2.774 $\pm$4.312	42.75×	36.35 $\pm$2.80	26.18 $\pm$3.76	0.75 $\pm$0.51	1.447 $\pm$0.134
FedRecovery	87.27 $\pm$1.46	4.21 $\pm$4.18	0.36 $\pm$0.36	0.180 $\pm$0.065	29.42 $\pm$8.77	42.76 $\pm$16.44	0.77 $\pm$0.77	2.242 $\pm$0.692	3.78×	25.42 $\pm$2.14	32.01 $\pm$3.44	**0.10** $\pm$0.12	2.208 $\pm$0.116
ConDa	91.68 $\pm$0.45	3.37 $\pm$1.49	1.17 $\pm$0.74	**0.040** $\pm$0.011	66.63 $\pm$36.92	25.81 $\pm$38.79	**0.52** $\pm$0.48	6.586 $\pm$12.704	2.59×	82.59 $\pm$1.78	26.18 $\pm$2.68	5.78 $\pm$0.37	**0.367** $\pm$0.019
SIFU	80.31 $\pm$0.89	7.17 $\pm$3.55	0.65 $\pm$0.60	0.304 $\pm$0.020	65.33 $\pm$1.53	14.53 $\pm$6.66	0.55 $\pm$0.43	0.626 $\pm$0.020	85.81×	29.57 $\pm$0.33	32.46 $\pm$2.26	1.02 $\pm$0.26	1.694 $\pm$0.061
uS2U	89.95 $\pm$0.39	1.98 $\pm$1.12	0.84 $\pm$0.88	0.055 $\pm$0.007	82.90 $\pm$2.13	10.11 $\pm$6.57	0.87 $\pm$0.46	0.576 $\pm$0.129	4.09×	61.68 $\pm$1.77	5.33 $\pm$2.16	1.35 $\pm$0.85	0.697 $\pm$0.050
zkFU-TraceTrust (Ours)	**92.52** $\pm$0.38	2.01 $\pm$1.45	0.51 $\pm$0.50	0.043 $\pm$0.008	93.27 $\pm$1.47	10.84 $\pm$6.31	1.31 $\pm$0.97	0.277 $\pm$0.105	**1.00×**	**93.48** $\pm$0.73	20.75 $\pm$2.47	1.90 $\pm$1.58	0.412 $\pm$0.024

Table 2: Halo2 prototype costs with sequential proof blocks; model computation excludes proving. Replay uses a separate 2^{17}-row configuration at $N = 20$, $T = 100$: one measured round extrapolated to T for Fashion-MNIST, and projected costs for ResNet-18 (Appendix H.8). Matched Fashion-MNIST zkFU proving takes 74.6 s, giving the 930× ratio.

Dataset / Model	Params d	Blocks × rows	Unl. Time	Prove Time	Peak Mem.	Proof Size	Verify Time	Replay Prove Time
FMNIST / CNN	44,426	3×2^{18}	2.08 s	113.7 s	21.91 GiB	36 KiB	0.12 s	19.3 h
CIFAR-10 / ResNet-18	11,183,562	683×2^{19}	4.54 s	8.30 h	14.96 GiB	4.84 MiB	29.2 s	$\approx$ 4,655 h
CIFAR-100 / ResNet-18	11,229,732	686×2^{19}	4.58 s	8.63 h	9.83 GiB	4.86 MiB	29.4 s	$\approx$ 4,674 h

the stored forgotten-client trace from θ_T without retained-mass correction or recovery. Baseline descriptions are given in Appendix G.2.

ZK proof and implementation. We benchmark a block-wise Halo2 prototype separately from unlearning on an AMD Ryzen 9 7950X workstation with 64 GB RAM and two NVIDIA RTX 4090 GPUs.

8.2 Results

Strong retained utility and retraining fidelity. Table 1 shows that zkFU is the only evaluated method ranked first or second in both Acc_r and KL on all three datasets. It achieves the highest retained accuracy on FMNIST and CIFAR-100, including a 4.19-point gain over PGD on CIFAR-100. Test accuracies are 89.27%, 86.52%, and 60.81%, respectively (Appendix H). The combination matters: FCU's best CIFAR-10 KL comes with 7.35 points lower retained accuracy than zkFU at 32.70× its unlearning time, while FedAU's best CIFAR-100 Δ_f comes with 13.65 points lower retained accuracy. These utility and KL gains coexist with Δ_f values of 10.84 and 20.75 pp on CIFAR-10 and CIFAR-100. Its mean CIFAR-10 MIA Gap is equivalent to retraining within ± 1 pp over 20 seeds (Table 3b).

Retained-aware recovery makes the trace useful. Direct Accumulation yields retained accuracies of 38.41, 23.84, and 11.07 on FMNIST, CIFAR-10, and CIFAR-100, with KL above 9. The TCGS ablation likewise starts at 38.21 Acc_r on FMNIST and reaches 92.52 after TraceTrust (Table 3a). The full pipeline thus converts a low-utility trace seed into an effective recovered model.

Certificate generation scales with model dimension. In Table 2, model computation takes 2 to 5 s in the simulation. Certificate generation takes 114 s for the CNN and 8.3–8.6 h for ResNet-18 per request, with proof verification taking 0.12–29.4 s for 36 KiB–4.9 MiB certificates. After

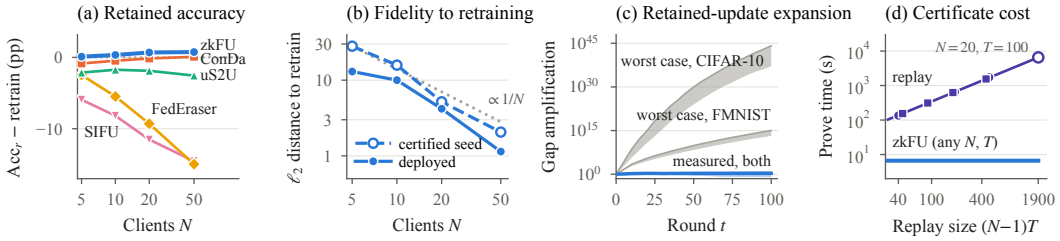

Figure 2: Scaling of zkFU. (a) Retained accuracy of trace-based methods relative to retraining on FMNIST. (b) ℓ_2 distance of the certified seed and the deployed model to retraining on FMNIST, with a dotted $1/N$ reference. (c) Trajectory-gap amplification estimated from trajectory probes (band over 3 FMNIST and 2 CIFAR-10 seeds) and in the worst case of Assumption 1 (Appendix F.3). (d) Prove time at $d = 4{,}096$ against the replay size, sweeping N and T, with the open marker a linear extrapolation to $N = 20$, $T = 100$.

Table 3: (a) FMNIST ablation. (b) MIA Gap (pp): five cumulative requests k, mean over 5 seeds, and one request over 20 seeds. Each p-value compares zkFU with its row: Welch's t-test for a difference from Original, and two one-sided tests (TOST) for equivalence to Retrain within ± 1 pp.

(a) Ablation	Acc$_r$ ↑	TA ↑	KL ↓
TCGS only	$38.21_{\pm 15.73}$	$36.12_{\pm 15.28}$	$10.59_{\pm 5.20}$
+ RCP	$47.69_{\pm 7.84}$	$45.81_{\pm 8.17}$	$4.665_{\pm 4.504}$
+ FTD	$51.59_{\pm 6.71}$	$49.67_{\pm 7.07}$	$2.899_{\pm 1.545}$
+ FSTM	$85.28_{\pm 3.69}$	$82.23_{\pm 4.61}$	$0.287_{\pm 0.208}$
+ TCI	$90.25_{\pm 2.18}$	$87.11_{\pm 2.93}$	$0.084_{\pm 0.075}$
+ FDS-ADR	$92.52_{\pm 0.38}$	$89.27_{\pm 0.52}$	$0.043_{\pm 0.008}$

(b) MIA Gap	Model	k=1	2	3	4	5	20 seeds	p vs. zkFU
FMNIST	Original	1.45	1.77	1.75	1.65	1.97	$1.79_{\pm 0.82}$	2.1×10^{-4}
	Retrain	0.52	0.65	0.25	0.44	0.31	$0.88_{\pm 0.65}$	1.7×10^{-5}
	zkFU	0.59	0.59	0.55	0.62	0.30	$0.83_{\pm 0.64}$	–
CIFAR-10	Original	2.32	2.49	2.63	2.83	2.47	$2.50_{\pm 0.88}$	2.2×10^{-7}
	Retrain	0.57	0.55	0.29	0.64	0.63	$0.61_{\pm 0.56}$	5.0×10^{-4}
	zkFU	1.05	0.47	0.52	0.72	0.53	$0.89_{\pm 0.72}$	–

public aggregation, the circuit cost is set by d and stays flat in N and T (Figure 2d), whereas our aggregation-only replay baseline grows as $(N-1)T$. On the matched Fashion-MNIST configuration, one measured replay round projected to $T = 100$ takes about $930\times$ longer than zkFU at $N = 20$.

zkFU remains close to retraining as the federation grows. FedEraser and SIFU fall about 15 pp below retraining at $N = 50$, while zkFU is the only trace-based method that stays at or above retraining at every N. In this federation-size sweep, the seed approaches retraining faster than a $1/N$ reference, and TraceTrust further reduces the parameter gap (Figure 2a,b).

Sequential requests preserve compact verification and low MIA gaps. In Table 3(b), the MIA Gap of zkFU tracks the retrain oracle at every cumulative request, far below the original model, while the circuit shape and proof size stay fixed across requests (Appendix H.5). Over 20 seeds, Welch's t-test separates zkFU from the original model and TOST establishes equivalence of the mean MIA Gap with retraining within ± 1 pp on both datasets, complementing the 5-trial MIA Gaps in Table 1. Appendix C.2 adds selected-round deletion and Appendix H.4 further attacks.

Ablation study. FSTM raises Acc$_r$ from 51.59 to 85.28 (Table 3a). TCGS defines what to remove, sparse trust filtering selects retained-aware coordinates, calibration decides how far to move, and shielded repair recovers utility while discouraging reversal of the trace displacement.

9 Conclusion

We presented **zkFU**, a framework for verifiable federated unlearning through trace-consistent subtraction. Linear aggregation yields a closed-form deletion relation over committed updates, reducing the proof target from direct $O(NTd)$ replay to an $O(d)$ statement after public aggregation. TraceTrust recovers retained utility through sparse filtering, calibration, and shielded repair. Across three datasets, zkFU alone ranks in the top two for both retained accuracy and KL among 11 baselines; 20-seed tests support mean shadow-attack MIA gaps within ± 1 pp of retraining on Fashion-MNIST and CIFAR-10. The combined method couples compact verification with effective retained-aware recovery.

AI USE STATEMENT

Large language models assisted with language editing and manuscript review. The authors take full responsibility for the paper's content.

ETHICS STATEMENT

This work provides auditable evidence of trace-consistent client-contribution subtraction. All experiments use public benchmark datasets and involve no human subjects.

REPRODUCIBILITY STATEMENT

We provide anonymized source code for zkFU in the supplementary material, including TCGS, the Halo2 certificate backend, TraceTrust, and the scripts for its ablation, sensitivity, and certificate-cost experiments. Section 6 and Appendix D specify the complete TraceTrust procedure, and Appendices A and B give the certificate relation, Halo2 circuit, and protocol steps. Theorem 1, Corollaries 3 and 4, and Theorem 2 are proved in Appendices F and E under the stated assumptions. Appendix G lists datasets, optimizer settings for every stage, baseline descriptions, and the simulation of the recovery stage, and Appendix H states the number of seeds and the reference oracle used for each additional result.

REFERENCES

Kasra Abbaszadeh, Christodoulos Pappas, Jonathan Katz, and Dimitrios Papadopoulos. Zero-knowledge proofs of training for deep neural networks. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 4316–4330, 2024. doi: 10.1145/3658644.3670316.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 141–159, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 463–480, 2015.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 1897–1914, 2022.

Vikram S Chundawat, Pushkar Niroula, Prasanna Dhungana, Stefan Schoepf, Murari Mandal, and Alexandra Brintrup. Conda: Fast federated unlearning with contribution dampening. *arXiv preprint arXiv:2410.04144*, 2024.

Zhipeng Deng, Luyang Luo, and Hao Chen. Enable the right to be forgotten with federated client unlearning in medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 240–250. Springer, 2024.

Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 479–496. IEEE, 2025.

Yann Fraboni, Martin Van Waerebeke, Kevin Scaman, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3457–3465. PMLR, 2024.

Xiangshan Gao, Xingjun Ma, Jingyi Wang, Yisen Sun, Bo Li, Shoufeng Ji, Peng Cheng, and Jiming Chen. VeriFi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5720–5736, 2024.

Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. Unlearning during learning: an efficient federated machine unlearning method. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4035–4043, 2024.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, PMLR, pp. 3832–3842, 2020.

Xintong Guo, Pengfei Wang, Sen Qiu, Wei Song, Qiang Zhang, Xiaopeng Wei, and Dongsheng Zhou. Fast: Adopting federated unlearning to eliminating malicious terminals at server side. *IEEE Transactions on Network Science and Engineering*, 11(2):2289–2302, 2023.

Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521*, 2022.

Feng Hao. Schnorr Non-interactive Zero-Knowledge Proof. Technical Report 8235, RFC Editor, 2017. URL `https://www.rfc-editor.org/rfc/rfc8235.html`.

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

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, volume 119 of *PMLR*, pp. 5132–5143, 2020.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 2, pp. 429–450, 2020.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th international symposium on quality of service (IWQOS)*, pp. 1–10. IEEE, 2021.

Mohammad M Maheri, Sunil Cotterill, Alex Davidson, and Hamed Haddadi. ZK-APEX: Zero-knowledge approximate personalized unlearning with EXecutable proofs. *arXiv preprint arXiv:2512.09953*, 2025.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *PMLR*, pp. 1273–1282, 2017.

Thanh Linh Nguyen, Marcela Tuler de Oliveira, An Braeken, Aaron Yi Ding, and Quoc-Viet Pham. Towards verifiable federated unlearning: Framework, challenges, and the road ahead. *IEEE Internet Computing*, 2026.

Zibin Pan, Zhichao Wang, Chi Li, Kaiyan Zheng, Boqi Wang, Xiaoying Tang, and Junhua Zhao. Federated unlearning with gradient descent and conflict mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19804–19812, 2025.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 18075–18086, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 7611–7623, 2020.

Zhipeng Wang, Nanqing Dong, Jiahao Sun, William Knottenbelt, and Yike Guo. zkFL: Zero-knowledge proof-based gradient aggregation for federated learning. *IEEE Transactions on Big Data*, 2024.

Cheng Wu, Sencun Zhu, and Prasenjit Mitra. Federated unlearning with knowledge distillation. In *arXiv preprint arXiv:2201.09441*, 2022.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, pp. 268–282, 2018.

Zcash Foundation. The Halo2 book, 2022. URL `https://zcash.github.io/halo2/`.

Binchi Zhang, Zihan Chen, Cong Shen, and Jundong Li. Verification of machine unlearning is fragile. In *International Conference on Machine Learning*, pp. 58717–58738. PMLR, 2024.

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18:4732–4746, 2023.

Zhengyi Zhong, Weidong Bao, Ji Wang, Shuai Zhang, Jingxuan Zhou, Lingjuan Lyu, and Wei Yang Bryan Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30661–30670, 2025.

A Cryptographic Details and TCGS Certificate

A.1 Fixed-Point Encoding

Floating-point updates are encoded with deterministic fixed-point arithmetic. For scaling factor Q, a coordinate x is encoded as

$$\widetilde{x} = \mathrm{round}(Qx)$$

and embedded into $\mathbb{F}_p$, with negative values represented modulo p. The same encoding is used for updates, aggregation weights, and TCGS arithmetic. Quantization error is separate from the trajectory-mismatch error analyzed in Appendix F.

A.2 TCGS Certificate Relation

For a deletion request over $\mathcal{F}$, the verifier recomputes $a_{\mathcal{F}}$ and $c_{\mathcal{F}}^{\mathsf{agg}}$ from the public audit trace by equations 3 and 5. The public statement is

$$x_{\mathrm{tcgs}} = \big(\theta_0, \theta_T, \theta_{\mathrm{tcgs}}, \mathcal{F}, \{a_i\}_{i\in\mathcal{F}}, a_{\mathcal{F}}, c_{\mathcal{F}}^{\mathsf{agg}}, c_{\mathrm{tcgs}}\big), \tag{14}$$

and the witness is

$$w_{\mathrm{tcgs}} = \big(W_{\mathcal{F}}, r_{\mathcal{F}}^{\mathsf{agg}}, r_{\mathrm{tcgs}}\big). \tag{15}$$

Including θ_{tcgs} in the public statement binds the proof to the released seed.

All certificate equalities use the specified deterministic encoding and arithmetic. The relation $R_{\mathrm{tcgs}}(x_{\mathrm{tcgs}}, w_{\mathrm{tcgs}}) = 1$ holds iff

$$\mathsf{Com}(W_{\mathcal{F}}; r_{\mathcal{F}}^{\mathsf{agg}}) = c_{\mathcal{F}}^{\mathsf{agg}}, \tag{C1}$$

$$\Delta_{\mathrm{tcgs}} = \frac{a_{\mathcal{F}}(\theta_T - \theta_0) - W_{\mathcal{F}}}{1 - a_{\mathcal{F}}}, \tag{C2}$$

$$\theta_{\mathrm{tcgs}} = \theta_T + \Delta_{\mathrm{tcgs}}, \tag{C3}$$

$$\mathsf{Com}(\theta_{\mathrm{tcgs}}; r_{\mathrm{tcgs}}) = c_{\mathrm{tcgs}}. \tag{C4}$$

Constraint C1 proves that the witness opens the aggregate forgotten-client commitment. Constraints C2 and C3 enforce trace-consistent subtraction. Constraint C4 binds the proof to the committed seed. After public aggregation, the in-circuit statement is linear in d and independent of N and T.

A.3 Proof-System Properties

We use standard properties of zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs). Completeness gives

$$\Pr[\mathsf{Verify}(\mathsf{vk}, x, \mathsf{Prove}(\mathsf{pk}, x, w)) = 1] \geq 1 - \mathsf{negl}(\lambda)$$

for every valid (x, w). Knowledge soundness means that any efficient prover producing an accepted proof must know a witness satisfying the relation, except with negligible probability. Zero knowledge means that there exists a simulator Sim such that

$$\{\mathsf{Sim}(\mathsf{vk}, x)\} \overset{c}{\approx} \{\mathsf{Prove}(\mathsf{pk}, x, w)\}.$$

A.4 Halo2 Circuit

Circuit construction. The circuit construction combines commitment-opening checks for constraints C1 and C4 with fixed-point coordinate gates for constraints C2 and C3. A vector Pedersen opening check evaluates a fixed-base multi-scalar multiplication over the block's coordinates and its randomness term. The affine gates reconstruct the seed from the public model states and aggregation coefficients and the witness $W_{\mathcal{F}}$. The released seed, commitments, coefficients, and block index must be bound to the public statement; signed-range constraints and deterministic rounding specify the fixed-point relation. Each block checks its coordinate range, and acceptance requires all blocks. The block configurations and measured Halo2 prototype costs are reported in Appendix H.8. Blocks can be proved sequentially to bound peak memory or concurrently within the available memory budget.

Algorithm 1 ZK Proof Generation and Verification

Require: $\theta_0, \theta_T, \mathcal{F}, \{a_i\}$; public audit trace $\{c_i^t\}$; forgotten openings $\{\Delta_i^t, r_i^t\}_{i\in\mathcal{F},t=1}^{T}$; pk, vk
Ensure: $\theta_{\mathrm{tcgs}}, \pi_{\mathrm{tcgs}}, c_{\mathrm{tcgs}}, c_{\mathcal{F}}^{\mathsf{agg}}$

[Server]
Compute $a_{\mathcal{F}} \leftarrow \sum_{i\in\mathcal{F}} a_i$
Compute $W_{\mathcal{F}} \leftarrow \sum_{t=1}^{T}\sum_{i\in\mathcal{F}} a_i\Delta_i^t$
Compute $r_{\mathcal{F}}^{\mathsf{agg}} \leftarrow \sum_{t=1}^{T}\sum_{i\in\mathcal{F}} a_i r_i^t$ and $c_{\mathcal{F}}^{\mathsf{agg}} \leftarrow \sum_{t=1}^{T}\sum_{i\in\mathcal{F}} a_i c_i^t$
Assert $\mathsf{Com}(W_{\mathcal{F}}; r_{\mathcal{F}}^{\mathsf{agg}}) = c_{\mathcal{F}}^{\mathsf{agg}}$
Compute θ_{tcgs} by equation 4
Sample r_{tcgs} and set $c_{\mathrm{tcgs}} \leftarrow \mathsf{Com}(\theta_{\mathrm{tcgs}}; r_{\mathrm{tcgs}})$
Form $(x_{\mathrm{tcgs}}, w_{\mathrm{tcgs}})$ by equations 14–15
$\pi_{\mathrm{tcgs}} \leftarrow \mathsf{Prove}(\mathsf{pk}, x_{\mathrm{tcgs}}, w_{\mathrm{tcgs}})$
Publish $(\theta_{\mathrm{tcgs}}, \pi_{\mathrm{tcgs}}, c_{\mathrm{tcgs}}, c_{\mathcal{F}}^{\mathsf{agg}})$, together with θ_{out} from Algorithm 2

[Any Verifier]
Recompute $a_{\mathcal{F}} \leftarrow \sum_{i\in\mathcal{F}} a_i$
Recompute $c_{\mathcal{F}}^{\mathsf{agg}} \leftarrow \sum_{t=1}^{T}\sum_{i\in\mathcal{F}} a_i c_i^t$ from the public audit trace
Form x_{tcgs} using the published θ_{tcgs}, c_{tcgs}, and recomputed $c_{\mathcal{F}}^{\mathsf{agg}}$
return $\mathsf{Verify}(\mathsf{vk}, x_{\mathrm{tcgs}}, \pi_{\mathrm{tcgs}})$

B Protocol and Implementation Details

Block commitments. For large models, we instantiate vector commitments block-wise. With block size B_c, the flattened update is split into $\lceil d/B_c \rceil$ blocks, and client C_i publishes

$$c_{i,b}^t = \mathsf{Com}(\Delta_{i,b}^t; r_{i,b}^t) = \sum_{j\in b} \widetilde{\Delta}_{i,j}^t \mathsf{G}_j + r_{i,b}^t \mathsf{H}.$$

All aggregate commitments are computed block-wise. The main text writes c_i^t and $c_{\mathcal{F}}^{\mathsf{agg}}$ as single vector commitments for readability.

Storage. With 32-byte compressed group elements, the public audit trace takes $NT\lceil d/B_c \rceil \times 32$ bytes, and a Merkle root over it can be published to bind it compactly (Wang et al., 2024). Commitment generation is incurred during training, whereas proof generation is triggered only when deletion is requested. The server-side witness material is the set of openings. Per-round retention stores dT coordinates per client, about 4.5 GB per ResNet-18 client at $T = 100$ and 32-bit precision. The TCGS witness in equation 15 needs only the aggregate pair $(W_{\mathcal{F}}, r_{\mathcal{F}}^{\mathsf{agg}})$, so for fixed-weight whole-client deletion the server can maintain one running pair per client, $S_i = \sum_t a_i\Delta_i^t$ and $R_i = \sum_t a_i r_i^t$, update both online during training, and discard per-round openings. Storage falls from $O(NTd)$ to $O(Nd)$. Measured system-wide opening storage falls from 373.18 to 3.73 MiB on Fashion-MNIST and from $93{,}894$ to 939 MiB on CIFAR-10, and the ResNet-18 per-client figure falls to about 45 MB. Public commitment aggregation takes $O(|\mathcal{F}|T\lceil d/B_c \rceil)$ group operations before proof verification. Verification is unchanged because verifiers recompute $c_{\mathcal{F}}^{\mathsf{agg}}$ from the public per-round commitments. Round-wise partial participation and selected-round deletion (Appendix C) require per-round openings for the rounds that may be corrected.

Training-consistency certificate. The aggregate-opening machinery also certifies the released training output. The public commitments of all clients aggregate to $c_{\mathrm{all}}^{\mathsf{agg}} = \sum_t \sum_i a_i c_i^t$, which by homomorphism commits to $\sum_t \sum_i a_i \Delta_i^t$. A one-time $O(d)$ proof that $c_{\mathrm{all}}^{\mathsf{agg}}$ opens to $\theta_T - \theta_0$ establishes that the released model is consistent with the committed trace, so a server cannot later present a trace whose aggregate is inconsistent with θ_T. The seed-quality analysis also requires updates computed from the shared round states; these states can be checked by the per-round form of the certificate. The relation is the same aggregate opening and coordinate-wise affine check as the TCGS certificate with $\mathcal{F}$ replaced by the full client set and the affine map replaced by the identity. Because the identity map needs no fixed-point product, this certificate is far cheaper than the deletion certificate. Measured as in Appendix H.8, it takes 1.9 s for Fashion-MNIST and 296 s for ResNet-18,

whose 86 proofs of 2^{17} rows are limited by the instance column, and verifies in 0.11 s and 27 s, providing a separate check of the training endpoint. The per-round version costs T times as much. On our traces, equation 1 holds up to floating-point accumulation, with ℓ_2 residual 1.8×10^{-5} on Fashion-MNIST and 1.0×10^{-4} on CIFAR-10 before quantization.

Cumulative requests. For the k-th deletion request the server and every verifier use the cumulative set $\mathcal{F}(k) = \bigcup_{j \le k} \mathcal{F}_j$ in Algorithm 1. The original trace is never modified, and the previously recovered model is not the input to the new request. Proof size and verification cost are independent of k, and off-circuit aggregation over the board grows with $|\mathcal{F}(k)|$. Because the public statement in equation 14 contains $\mathcal{F}(k)$ and its weights, a certificate issued for one cumulative set does not verify for another, and a request index can be added to the statement when freshness is also required.

C Partial Participation

The main text presents TCGS under full participation with fixed aggregation weights. Under partial participation, absent clients have $a_i^t = 0$, and the forgotten mass becomes

$$a_{\mathcal{F}}^t = \sum_{i \in \mathcal{F}} a_i^t.$$

Assuming $1 - a_{\mathcal{F}}^t > 0$ for every corrected round, the retained-normalized trace seed is

$$\theta_{\mathrm{tcgs}} = \theta_0 + \sum_{t=1}^{T} \frac{(\theta_t - \theta_{t-1}) - W_{\mathcal{F}}^t}{1 - a_{\mathcal{F}}^t}, \qquad W_{\mathcal{F}}^t = \sum_{i \in \mathcal{F}} a_i^t \Delta_i^t. \tag{16}$$

The corresponding ZK relation remains affine. The verifier aggregates commitments using the public round-wise weights, and the circuit checks the same type of aggregate opening and coordinate-wise affine reconstruction.

Under full participation $a_{\mathcal{F}}^t = a_{\mathcal{F}}$ for every t, and equation 16 reduces algebraically to equation 4.

Selected-round deletion. The round-wise relation also removes a client's committed contribution over a selected set of rounds $S \subseteq \{1, \dots, T\}$. Setting the client's weight to zero on S and leaving it unchanged elsewhere gives the same affine statement with $W_{\mathcal{F}}^t$ and $a_{\mathcal{F}}^t$ nonzero only for $t \in S$, so the certificate stays $O(d)$ after public aggregation. The certified object is exact subtraction of the committed window contribution.

C.1 Evaluation Under Client Sampling

Table 4 evaluates the deployed zkFU model under client sampling, with TraceTrust applied to the round-wise seed of equation 16.

Table 4: Deployed zkFU model under client sampling, built from the round-wise seed of equation 16. Each cell reports test accuracy and Δ_f against a retrain oracle trained with the same round budget at the same sampling rate. Fashion-MNIST over 5 seeds and CIFAR-10 over 2 seeds. The CIFAR-10 row at rate 1.0 is the full-participation run of Table 15 ($k = 1$, 5 seeds).

Sampling rate	Fashion-MNIST / CNN		CIFAR-10 / ResNet-18	
	Same-round oracle	zkFU	Same-round oracle	zkFU
0.5	84.91, 0.00	88.50, 7.71	66.19, 0.00	84.91, 19.27
0.75	87.56, 0.00	88.91, 4.97	75.22, 0.00	86.78, 13.12
1.0	88.90, 0.00	89.35, 1.34	77.71, 0.00	85.89, 12.37

KL of zkFU to the same-round oracle at rates 0.5 / 0.75 / 1.0: Fashion-MNIST 0.273 / 0.149 / 0.045, CIFAR-10 0.872 / 0.359 / 0.212.

At sampling rate 0.5, a retrain oracle with the same round budget is unstable, reaching 84.91 ± 6.02 test accuracy on Fashion-MNIST and 66.19 ± 3.70 on CIFAR-10, because it sees only half of the local updates of the full run. Table 5 therefore also gives a retrain oracle with equal total local-update compute, obtained by training for T/rate rounds. On CIFAR-10 the equal-compute oracle reaches

82.75 and 80.31 test accuracy at rates 0.5 and 0.75, and the forgotten-client gap of zkFU relative to it falls from 19.27 to 5.64 pp and from 13.12 to 4.68 pp. Behavioral distance under sampling therefore depends materially on the counterfactual training budget, while the affine certificate relation is exact at every rate. Table 4 uses the same-round oracle; Table 5 adds the equal-compute comparison.

Table 5: Same-round against equal-compute retrain oracle under client sampling on CIFAR-10 with ResNet-18 over 2 seeds. The equal-compute oracle trains for T/rate rounds rounded to an integer, to approximately match the full-participation local-update budget.

Sampling rate	Oracle	Rounds	Oracle test acc.	zkFU Δ_f vs. oracle (pp)
0.5	same rounds	100	$66.19_{\pm 3.70}$	19.27
0.5	equal compute	200	$82.75_{\pm 5.40}$	5.64
0.75	same rounds	100	$75.22_{\pm 4.78}$	13.12
0.75	equal compute	133	$80.31_{\pm 2.51}$	4.68

C.2 Evaluation of Selected-Round Deletion

Table 6 removes the forgotten client's committed contribution for rounds 51 to 100 only. The matched counterfactual is a windowed-retrain oracle that includes the client in rounds 1 to 50 and excludes it in rounds 51 to 100 with the same initialization, seed, and round budget. Plain retraining with the client absent everywhere is shown for contrast. On Fashion-MNIST, selected-round zkFU differs from the windowed oracle by 0.37 pp in test accuracy and 1.74 pp in forgotten-client accuracy, and its MIA Gap of 0.50 is below the oracle's 0.88 and the original model's 1.86. On CIFAR-10 both MIA Gaps lie below the original model's 2.06. Against the windowed oracle, selected-round zkFU is about as close as full retraining, with ℓ_2/KL of 4.49/0.054 against 4.81/0.059 on Fashion-MNIST and 15.24/0.022 against 15.89/0.017 on CIFAR-10. An independent five-seed Fashion-MNIST run reproduces the result with 89.36 ± 0.43 test accuracy, 91.78 ± 2.79 forgotten-client accuracy, and MIA Gap 0.53 ± 0.27.

Table 6: Selected-round deletion of rounds 51 to 100 over 3 seeds. The windowed-retrain oracle is the matched counterfactual for a window claim. ℓ_2 and KL are measured against the windowed oracle. MIA Gap in pp.

Model	Fashion-MNIST / CNN				CIFAR-10 / ResNet-18			
	Test acc.	Forgotten acc.	MIA Gap	ℓ_2 / KL	Test acc.	Forgotten acc.	MIA Gap	ℓ_2 / KL
Original model θ_T	$88.71_{\pm 0.53}$	$93.78_{\pm 1.65}$	$1.86_{\pm 0.59}$	–	$80.51_{\pm 1.58}$	$88.40_{\pm 4.57}$	$2.06_{\pm 1.42}$	–
Windowed-retrain oracle	$88.78_{\pm 0.34}$	$90.59_{\pm 1.18}$	$0.88_{\pm 0.37}$	0 / 0	$83.19_{\pm 3.28}$	$87.23_{\pm 2.17}$	$0.15_{\pm 0.09}$	0 / 0
Full retrain (contrast)	$88.74_{\pm 0.48}$	$90.85_{\pm 1.65}$	$0.46_{\pm 0.26}$	4.81 / 0.059	$74.67_{\pm 8.32}$	$74.88_{\pm 13.67}$	$0.39_{\pm 0.50}$	15.89 / 0.017
Selected-round zkFU	$89.15_{\pm 0.31}$	$92.33_{\pm 1.16}$	$0.50_{\pm 0.35}$	4.49 / 0.054	$86.72_{\pm 0.28}$	$90.03_{\pm 1.89}$	$0.80_{\pm 0.33}$	15.24 / 0.022

D Full Unlearning Algorithm

This section expands the three stages of Section 6, the TCGS seed and the TraceTrust recovery, into procedure-level pseudocode. The sparse objective motivates the closed-form filtering surrogate; diagonal damping and masking need not preserve the RCP half-space constraint. FDS regularizes the repair gradient before the optimizer applies momentum and weight decay.

Algorithm 2 zkFU unlearning: TCGS seed and TRACETRUST recovery

Require: $\theta_0, \theta_T, \mathcal{F}, \{a_i\}$; public audit trace $\{c_i^t\}$; forgotten openings $\{\Delta_i^t, r_i^t\}_{i\in\mathcal{F},t=1}^{T}$; retained-statistics oracle $\mathcal{O}_R$; number of repair steps n_r; hyperparameters $\rho, \lambda_F, \mathcal{B}, \lambda_{\mathrm{kd}}, \lambda_{\mathrm{pull}}, \lambda_{\mathrm{prox}}, \lambda_{\mathrm{sh}}, \tau_{\mathrm{kd}}, \eta_r, \varepsilon_g, \varepsilon_F, \varepsilon_d$

Ensure: deployed model θ_{out}, certified seed θ_{tcgs}, proof π_{tcgs}, seed commitment c_{tcgs}, aggregate trace commitment $c_{\mathcal{F}}^{\mathsf{agg}}$

Stage 1. ZK-verifiable trace-consistent subtraction.

Compute $a_{\mathcal{F}}$ and $W_{\mathcal{F}}$ by equation 3
Aggregate $c_{\mathcal{F}}^{\mathsf{agg}}$ and $r_{\mathcal{F}}^{\mathsf{agg}}$ by equation 5
Compute Δ_{tcgs} and θ_{tcgs} by equation 4
Commit to θ_{tcgs} as c_{tcgs}
Generate π_{tcgs} for R_{tcgs}

Stage 2. Retained-aware sparse trust filtering.

Obtain aggregated retained statistics $(g_R, F_R) \leftarrow \mathcal{O}_R(\theta_T)$
Apply RCP by equation 7 to obtain Δ_{proj}
Apply FTD by equation 8 to obtain Δ_{trust}
Apply FSTM by equation 9 to obtain Δ_{sparse}
Set $\theta_{\mathrm{sparse}} = \theta_T + \Delta_{\mathrm{sparse}}$

Stage 3. Trust-calibrated and forgetting-shielded repair.

for each $\beta \in \mathcal{B}$ **do**
    Evaluate $J(\beta)$ along $\theta(\beta) = \theta_T + \beta(\theta_{\mathrm{sparse}} - \theta_T)$
end for
Select $\beta^\star$ and set $\theta_{\mathrm{cal}} = \theta_T + \beta^\star(\theta_{\mathrm{sparse}} - \theta_T)$
Set $d_F = \theta_{\mathrm{tcgs}} - \theta_T$ and initialize $\theta \leftarrow \theta_{\mathrm{cal}}$
for $s = 1, \ldots, n_r$ **do**
    Retained clients compute local repair gradients
    Aggregate g_{repair} by equation 12
    Shield g_{repair} along d_F by equation 13
    Apply the repair optimizer to θ using g_{shield} and stepsize η_r
end for
Set $\theta_{\mathrm{out}} \leftarrow \theta$

return $\theta_{\mathrm{out}}, \theta_{\mathrm{tcgs}}, \pi_{\mathrm{tcgs}}, c_{\mathrm{tcgs}}, c_{\mathcal{F}}^{\mathsf{agg}}$

E SECURITY PROOFS

E.1 SOUNDNESS

Proof. Suppose an adversary outputs θ^*_{tcgs} and π^* such that

$$\mathsf{Verify}(\mathsf{vk}, x^*_{\mathrm{tcgs}}, \pi^*) = 1.$$

The verifier recomputes $a_{\mathcal{F}}$ and $c^{\mathsf{agg}}_{\mathcal{F}}$ from the public audit trace before forming x^*_{tcgs}. By knowledge soundness, an accepting proof implies a witness

$$w^* = (W^*_{\mathcal{F}}, r^{\mathsf{agg},*}_{\mathcal{F}}, r^*_{\mathrm{tcgs}})$$

satisfying equations C1–C4, except with negligible probability. From equation C1,

$$\mathsf{Com}(W^*_{\mathcal{F}}; r^{\mathsf{agg},*}_{\mathcal{F}}) = c^{\mathsf{agg}}_{\mathcal{F}}.$$

The same public commitment is also

$$c^{\mathsf{agg}}_{\mathcal{F}} = \sum_{t=1}^{T} \sum_{i \in \mathcal{F}} a_i c_i^t = \mathsf{Com}(W_{\mathcal{F}}; r^{\mathsf{agg}}_{\mathcal{F}}).$$

By binding, $W^*_{\mathcal{F}} = W_{\mathcal{F}}$ except with negligible probability. Equations C2 and C3 then force

$$\theta^*_{\mathrm{tcgs}} = \theta_T + \frac{a_{\mathcal{F}}(\theta_T - \theta_0) - W_{\mathcal{F}}}{1 - a_{\mathcal{F}}}.$$

Thus, every accepted proof binds the released seed to the unique trace-consistent TCGS output. □

E.2 ZERO KNOWLEDGE

Proof. The TCGS witness is

$$w_{\mathrm{tcgs}} = (W_{\mathcal{F}}, r^{\mathsf{agg}}_{\mathcal{F}}, r_{\mathrm{tcgs}}).$$

It contains only the aggregate forgotten-client trace contribution and commitment randomness. Retained-client openings never appear in the witness. By zero knowledge of the proof system, the proof can be simulated from the public statement and reveals no additional witness information beyond that public statement. □

What is not hidden. Since θ_{tcgs} is public, the aggregate forgotten effect implied by the affine relation is also public:

$$W_{\mathcal{F}} = a_{\mathcal{F}}(\theta_T - \theta_0) - (1 - a_{\mathcal{F}})(\theta_{\mathrm{tcgs}} - \theta_T).$$

For whole-client deletion, this is a weighted sum across rounds. A selected-round request reveals the corresponding aggregate and, for a single client and round, its update. Across cumulative requests, $W_{\mathcal{F}(k)} - W_{\mathcal{F}(k-1)}$ reveals the newly added clients' historical aggregate. Differences between overlapping windows can likewise reveal finer contributions. The ZK guarantee is conditioned on the joint public statements and their implied aggregates over the request history; it does not hide information already determined by those releases.

E.3 SECURITY ASSUMPTIONS AND ATTACK SURFACE

Training-phase checks. Commitments are generated by the clients, bound to authenticated identities, and appended to the board before aggregation, so ex-post rewriting or cross-identity reassignment of the trace is detectable, and each client can verify inclusion of its own commitments. The training-consistency certificate (Appendix B) fixes θ_T to the committed trace, and applied per round it fixes every aggregation step, which is the property zkFL (Wang et al., 2024) proves. Proofs of training (Abbaszadeh et al., 2024) can additionally attest that each update was computed from the claimed data, and identity enrollment determines which identities a request $\mathcal{F}$ covers.

Assumptions. The certificate assumes a prime-order group in which the discrete-log problem is hard and Pedersen generators $\{\mathsf{G}_j\}, \mathsf{H}$ are derived independently, so that the vector commitment is binding and hiding. It assumes completeness, knowledge soundness, and zero knowledge of the Halo2 proof system. It assumes that clients submit commitments under authenticated identities with Sybil-resistant enrollment, and that the audit board is append-only and publicly readable. Theorem 2 and Appendix E.1 reduce soundness of the trace-defined seed to these assumptions. The trajectory-quality bound additionally requires client updates to follow an honest or independently authenticated FedAvg trajectory.

E.4 Completeness

Proof. An honest prover computes $W_{\mathcal{F}}$, $r_{\mathcal{F}}^{\mathsf{agg}}$, and θ_{tcgs} from the committed trace and equation 4, then commits to the seed as c_{tcgs}. Therefore, the witness satisfies equations C1–C4. Completeness of the proof system gives acceptance except with negligible probability. □

F TCGS Approximation and Quantization Error

F.1 Proof of Theorem 1

Let θ_t be the original FedAvg trajectory and $\bar{\theta}_t$ be the retained-only trajectory after removing $\mathcal{F}$ and renormalizing retained weights as $\bar{a}_j = a_j/(1 - a_{\mathcal{F}})$. In this proof, updates are indexed by $t = 0, \ldots, T-1$, with θ_t the state before update t. Then

$$\theta_{t+1} = \theta_t + \sum_{i=1}^{N} a_i \Delta_i^t(\theta_t), \qquad \bar{\theta}_{t+1} = \bar{\theta}_t + \sum_{j \notin \mathcal{F}} \bar{a}_j \Delta_j^t(\bar{\theta}_t). \tag{17}$$

Both trajectories start from θ_0. Let

$$W_R = \sum_{t=0}^{T-1} \sum_{j \notin \mathcal{F}} a_j \Delta_j^t(\theta_t).$$

The FedAvg trace gives $\theta_T - \theta_0 = W_{\mathcal{F}} + W_R$. Substituting this identity into equation 4 yields

$$\theta_{\mathrm{tcgs}} = \theta_0 + \frac{1}{1 - a_{\mathcal{F}}} \sum_{t=0}^{T-1} \sum_{j \notin \mathcal{F}} a_j \Delta_j^t(\theta_t). \tag{18}$$

The retained-only counterfactual is

$$\theta^*_{\setminus \mathcal{F}} = \theta_0 + \frac{1}{1 - a_{\mathcal{F}}} \sum_{t=0}^{T-1} \sum_{j \notin \mathcal{F}} a_j \Delta_j^t(\bar{\theta}_t). \tag{19}$$

Therefore,

$$\theta_{\mathrm{tcgs}} - \theta^*_{\setminus \mathcal{F}} = \frac{1}{1 - a_{\mathcal{F}}} \sum_{t=0}^{T-1} \sum_{j \notin \mathcal{F}} a_j \left(\Delta_j^t(\theta_t) - \Delta_j^t(\bar{\theta}_t) \right). \tag{20}$$

This identity identifies the core approximation source. TCGS uses retained updates evaluated on the observed trajectory, while exact retraining uses retained updates evaluated on a trajectory where $\mathcal{F}$ never participated.

Under Assumption 1,

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus \mathcal{F}}\|_2 \le L_\Delta \sum_{t=0}^{T-1} \|\theta_t - \bar{\theta}_t\|_2. \tag{21}$$

Let $\delta_t = \theta_t - \bar{\theta}_t$. From equation 17,

$$\delta_{t+1} = \delta_t + \sum_{j \notin \mathcal{F}} a_j \left(\Delta_j^t(\theta_t) - \Delta_j^t(\bar{\theta}_t) \right) + \sum_{i \in \mathcal{F}} a_i \Delta_i^t(\theta_t) - \frac{a_{\mathcal{F}}}{1 - a_{\mathcal{F}}} \sum_{j \notin \mathcal{F}} a_j \Delta_j^t(\bar{\theta}_t). \tag{22}$$

Using Lipschitz retained update maps and bounded updates gives

$$\|\delta_{t+1}\|_2 \le (1 + L_\Delta)\|\delta_t\|_2 + 2\eta G a_{\mathcal{F}}. \tag{23}$$

Since $\delta_0 = 0$,

$$\|\delta_t\|_2 \le 2\eta G a_{\mathcal{F}}\, t\, e^{L_\Delta T}. \tag{24}$$

Combining equations 21 and 24 yields

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le \eta G L_\Delta\, T(T+1)\, a_{\mathcal{F}}\, e^{L_\Delta T}. \tag{25}$$

This is the explicit non-asymptotic form of the seed-quality parameter ϵ used in Theorem 1 and in the seed-verifiable unlearning definition in Section 4. The factor $1/(1 - a_{\mathcal{F}})$ of equation 20 cancels against $\sum_{j\notin\mathcal{F}} a_j = 1 - a_{\mathcal{F}}$ in equation 21, so no additional reciprocal retained-mass factor appears. When the local-update map is stable with $L_\Delta T = O(1)$, it implies

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le O\big(L_\Delta \eta G T^2 a_{\mathcal{F}}\big)\,.$$

Role of the trace correction. The original model and the seed obey

$$\theta_T - \theta^*_{\setminus\mathcal{F}} = -\Delta_{\mathrm{tcgs}} + (\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}).$$

TCGS removes the recorded trace discrepancy; Theorem 1 bounds the residual caused by evaluating retained updates on the original trajectory. If update maps are independent of the model state, that residual is zero and TCGS exactly matches retained-only training. For state-dependent maps, $1/N$ scaling alone does not establish that the seed is always closer than the original model: equation 24 also bounds the original-model discrepancy. The analysis identifies the residual approximation source; the experiments assess its magnitude.

F.2 Corollaries Beyond the Stable Regime

Corollary 3 (Non-expansive retained updates)**.** *Consider single-step local gradient descent with local objectives f_i satisfying $\|\nabla f_i\|_2 \le G$, and let $\mathcal{L}_R = \sum_{j\notin\mathcal{F}} \bar{a}_j f_j$ with $\bar{a}_j = a_j/(1 - a_{\mathcal{F}})$. If $\mathcal{L}_R$ is convex and L-smooth and $\eta(1 - a_{\mathcal{F}}) \le 2/L$, then the retained component $S_R(\theta) = \theta - \eta(1 - a_{\mathcal{F}})\nabla\mathcal{L}_R(\theta)$ of the observed update map is non-expansive, the trajectory gap satisfies $\|\delta_t\|_2 \le 2\eta G a_{\mathcal{F}}\, t$, and*

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le L_\Delta\, \eta G\, a_{\mathcal{F}}\, T(T+1) = O\big(L\eta^2 G a_{\mathcal{F}} T^2\big)$$

for every T, with no exponential factor.

Corollary 4 (Inverse-time stepsizes)**.** *Let round t use stepsize $\eta_t = \eta_0/(t+1)$ with $L_{\Delta,t} \le L\eta_t$ and $\|\Delta_i^t(\theta)\|_2 \le \eta_t G$. Then $\prod_{t<T}(1 + L\eta_t) \le e^{L\eta_0} T^{L\eta_0}$, and*

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 = O\big(a_{\mathcal{F}}\, G\, L\, \eta_0^2\, T^{L\eta_0} \log^2 T\big)\,,$$

which is polynomial rather than exponential in T.

Both corollaries replace the step bound in equation 23 and keep the rest of the argument. The factor $e^{L_\Delta T}$ in equation 24 enters only through that step bound, which uses the Lipschitz condition alone.

Proof of Corollary 3. With single-step local gradient descent, $\Delta_j^t(\theta) = -\eta\nabla f_j(\theta)$, and the two trajectories in equation 17 read

$$\theta_{t+1} = \theta_t - \eta(1 - a_{\mathcal{F}})\nabla\mathcal{L}_R(\theta_t) - \eta\sum_{i\in\mathcal{F}} a_i \nabla f_i(\theta_t), \qquad \bar{\theta}_{t+1} = \bar{\theta}_t - \eta\nabla\mathcal{L}_R(\bar{\theta}_t).$$

Writing $S_R(\theta) = \theta - \eta(1 - a_{\mathcal{F}})\nabla\mathcal{L}_R(\theta)$, the gap $\delta_t = \theta_t - \bar{\theta}_t$ satisfies

$$\delta_{t+1} = S_R(\theta_t) - S_R(\bar{\theta}_t) + \eta a_{\mathcal{F}}\nabla\mathcal{L}_R(\bar{\theta}_t) - \eta\sum_{i\in\mathcal{F}} a_i \nabla f_i(\theta_t).$$

If $\mathcal{L}_R$ is convex and L-smooth and $\eta(1 - a_{\mathcal{F}}) \le 2/L$, the gradient step S_R is non-expansive, so $\|S_R(\theta_t) - S_R(\bar{\theta}_t)\|_2 \le \|\delta_t\|_2$. The two remaining terms are each bounded by $\eta a_{\mathcal{F}} G$, since

$\|\nabla \mathcal{L}_R\|_2 \le \sum_j \bar{a}_j \|\nabla f_j\|_2 \le G$ and $\|\sum_{i\in\mathcal{F}} a_i \nabla f_i\|_2 \le a_{\mathcal{F}} G$. Hence $\|\delta_{t+1}\|_2 \le \|\delta_t\|_2 + 2\eta G a_{\mathcal{F}}$, and with $\delta_0 = 0$ we obtain $\|\delta_t\|_2 \le 2\eta G a_{\mathcal{F}}\, t$. Substituting into equation 21,

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le L_\Delta \sum_{t=0}^{T-1} 2\eta G a_{\mathcal{F}}\, t \le L_\Delta\, \eta G a_{\mathcal{F}}\, T(T+1).$$

For single-step gradient descent $L_\Delta = \eta L$, which gives $O(L\eta^2 G a_{\mathcal{F}} T^2)$ for every T. □

Proof of Corollary 4. With round-dependent stepsizes η_t and Lipschitz constants $L_{\Delta,t} \le L\eta_t$, the step bound in equation 23 becomes $\|\delta_{t+1}\|_2 \le (1 + L\eta_t)\|\delta_t\|_2 + 2\eta_t G a_{\mathcal{F}}$. Unrolling from $\delta_0 = 0$,

$$\|\delta_t\|_2 \le 2G a_{\mathcal{F}} \sum_{s<t} \eta_s \prod_{s<r<t} (1 + L\eta_r) \le 2G a_{\mathcal{F}} \Big(\sum_{s<t} \eta_s \Big) \prod_{r<T} (1 + L\eta_r).$$

For $\eta_t = \eta_0/(t+1)$ and $1 \le t < T$, $\sum_{s<t} \eta_s \le \eta_0(1 + \ln t)$ and, using $1 + x \le e^x$ and the harmonic bound $\sum_{r<T} 1/(r+1) \le 1 + \ln T$,

$$\prod_{r<T} (1 + L\eta_r) \le \exp\Big(L\eta_0 \sum_{r<T} \tfrac{1}{r+1} \Big) \le e^{L\eta_0}\, T^{L\eta_0}.$$

The error bound in equation 21 with round-dependent constants reads $\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le \sum_t L_{\Delta,t} \|\delta_t\|_2$. Substituting the two displays and using $\delta_0 = 0$ and bounding $\sum_{t=1}^{T-1} \eta_t(1 + \ln t) \le \eta_0(1 + \ln T)^2$ gives

$$\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2 \le 2\, a_{\mathcal{F}}\, G\, L\, \eta_0^2\, e^{L\eta_0}\, T^{L\eta_0} (1 + \ln T)^2,$$

which is $O(a_{\mathcal{F}} G L \eta_0^2 T^{L\eta_0} \log^2 T)$. For $L\eta_0 < 1$ the growth in T is sublinear up to logarithmic factors. □

F.3 Empirical Behavior of the Trajectory Mismatch

We measure the distance $\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2$ of the raw TCGS seed to seed-matched retraining, sweeping one quantity of Theorem 1 at a time from the default configuration (Table 7, Figure 3, and Figure 2b for N). On Fashion-MNIST the fitted exponents in T and η lie below those of the stable-regime bound and the exponent in N is close to the $a_{\mathcal{F}} \approx 1/N$ scaling, consistent with these reference scaling laws over the measured ranges. On CIFAR-10 with ResNet-18, where parameter-space distance is dominated by seed-to-seed variation, the function-space measure shows the same trend. Raw-seed KL to retraining falls from 15.51 to 4.57 as N grows, and deployed KL reaches 0.145 at $N = 50$. At $T = 100$ on Fashion-MNIST, TraceTrust reduces raw-seed ℓ_2 from 5.18 to 4.18 and Δ_f from 80.95 to 1.67 pp (Table 10).

Table 8 extends the round sweep to $T \in \{100, 300, 1000\}$. The inverse-time schedule $\eta_t = \eta_0/(t+1)$ of Corollary 4 makes the raw seed about 7 to 12 times closer to retraining at every T, and its growth is consistent with the polynomial bound. Under the constant schedule the deployed model stays within 0.4 pp of retrain test accuracy at every T.

Table 7: Fitted log-log slope of the raw TCGS-seed distance $\|\theta_{\mathrm{tcgs}} - \theta^*_{\setminus\mathcal{F}}\|_2$ to seed-matched retraining, 3 seeds per point (2 at $T = 1000$, Table 8). The last column gives the stable-regime exponent of Theorem 1, except the inverse-time row, which uses Corollary 4 up to logarithmic factors.

Sweep	Dataset	Values	Fitted slope	Bound exponent
Rounds T	Fashion-MNIST	25 / 50 / 75 / 100	1.00	2
Clients N	Fashion-MNIST	5 / 10 / 20 / 50	-1.18	-1
Stepsize η	Fashion-MNIST	0.01 / 0.03 / 0.1	0.87	2
Rounds T (constant η)	Fashion-MNIST	100 / 300 / 1000	0.805	2
Rounds T (inverse-time η_t)	Fashion-MNIST	100 / 300 / 1000	0.558	$L\eta_0$

Table 8: Long-horizon trajectory mismatch on Fashion-MNIST under a constant stepsize and under the inverse-time schedule $\eta_t = \eta_0/(t+1)$. Raw seed is the TCGS seed before recovery and deployed is the TraceTrust output. ℓ_2 and KL are measured against seed-matched retraining. Seeds are 3, 3, and 2 at $T = 100$, 300, and 1000, drawn independently of the shorter-horizon round sweep in Table 7.

T	Constant stepsize			Inverse-time stepsize		
	Raw-seed ℓ_2	Deployed ℓ_2 / KL	Test acc. deployed / retrain	Raw-seed ℓ_2	Deployed ℓ_2 / KL	Test acc. deployed / retrain
100	$5.18_{\pm 0.26}$	4.18 / 0.032	89.15 / 88.77	$0.784_{\pm 0.091}$	4.53 / 0.076	78.59 / 80.77
300	$13.03_{\pm 0.35}$	8.22 / 0.094	89.07 / 88.90	$1.436_{\pm 0.144}$	7.05 / 0.058	81.51 / 83.03
1000	$33.10_{\pm 0.94}$	11.38 / 0.148	89.02 / 89.03	$2.834_{\pm 0.253}$	4.93 / 0.043	82.32 / 85.10

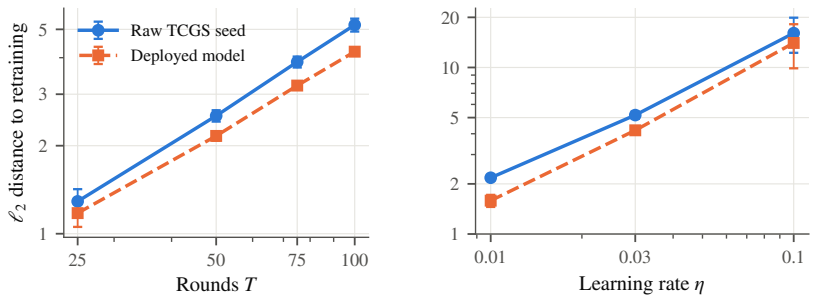

Figure 3: Raw TCGS-seed and deployed-model ℓ_2 distance to seed-matched retraining on Fashion-MNIST against rounds T (left) and stepsize η (right), log-log axes. Figure 2b gives the sweep over N.

Measured expansion of the retained update. Assumption 1 bounds each client separately, and the worst case compounds a factor $1 + L_\Delta$ per round. The trajectory gap is instead governed by the retained update map $\theta \mapsto \theta + \sum_{j \notin \mathcal{F}} a_j \Delta_j^t(\theta)$. We train the original and the retained-only runs on Fashion-MNIST and CIFAR-10 from the same initialization and, every fifth round, evaluate each retained client's update on both states with identical randomness, so the measurement is deterministic. Figure 4 reports the expansion ratio $\kappa_t = \|(\theta_t - \bar{\theta}_t) + \sum_{j \notin \mathcal{F}} a_j(\Delta_j^t(\theta_t) - \Delta_j^t(\bar{\theta}_t))\|_2 / \|\theta_t - \bar{\theta}_t\|_2$ over 3 Fashion-MNIST seeds and 2 CIFAR-10 seeds. On Fashion-MNIST, the median per-client ratio is 0.2 to 0.8, so the worst-case factor $e^{L_\Delta T}$ would exceed e^{30}, yet the aggregated retained map is almost non-expansive, with κ_t falling from at most 1.10 early in training to 1.006 by round 95 with probe-based amplification estimates of 5 to 12 over 100 rounds. The observed trajectory gap grows approximately linearly in t, consistent with the milder accumulation illustrated by Corollary 3. On CIFAR-10 with ResNet-18 (2 seeds) the per-client ratio is larger, 0.4 to 2.0, so the worst-case factor would exceed e^{85}, yet κ_t stays within $[0.90, 1.04]$ and gives a probe-based amplification estimate below 1 over 100 rounds, while the observed trajectory gap remains bounded. It stays between 13 and 18 for one seed and halves from 49 to 24 for the other.

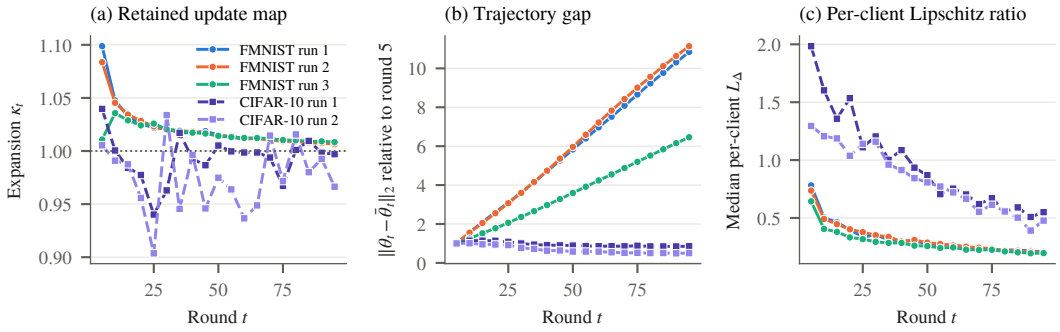

Figure 4: Retained FedAvg update measured along real trajectories (five local epochs, momentum 0.9): Fashion-MNIST, 3 seeds, and CIFAR-10 with ResNet-18, 2 seeds. (a) Expansion ratio κ_t of the retained update map; the dotted line marks non-expansiveness. (b) Gap between the original and the retained-only trajectories, relative to round 5. (c) Median over retained clients of $\|\Delta_j^t(\theta_t) - \Delta_j^t(\bar{\theta}_t)\|_2 / \|\theta_t - \bar{\theta}_t\|_2$.

F.4 Quantization Error

Let $\widehat{\Delta}_i^t$ be the fixed-point quantized update. If each coordinate has resolution δ_q, then

$$\|\Delta_i^t - \widehat{\Delta}_i^t\|_2 \le \frac{\sqrt{d}\,\delta_q}{2}.$$

For fixed aggregation weights and exact weighted accumulation, the forgotten-trace quantization error satisfies

$$\|W_{\mathcal{F}} - \widehat{W}_{\mathcal{F}}\|_2 \le \frac{\sqrt{d}\,\delta_q T a_{\mathcal{F}}}{2}.$$

Holding the endpoints and weights fixed, TCGS depends on $W_{\mathcal{F}}$ through a factor $(1 - a_{\mathcal{F}})^{-1}$, so

$$\|\theta_{\text{tcgs}} - \widehat{\theta}_{\text{tcgs}}\|_2 \le \frac{\sqrt{d}\,\delta_q T a_{\mathcal{F}}}{2(1 - a_{\mathcal{F}})}.$$

For a b-bit signed fixed-point encoding over coordinate range M_∞, $\delta_q = 2M_\infty/2^b$, giving

$$\|\theta_{\text{tcgs}} - \widehat{\theta}_{\text{tcgs}}\|_2 = O\left(\frac{\sqrt{d}\,M_\infty T a_{\mathcal{F}}}{(1 - a_{\mathcal{F}})2^b}\right).$$

This term is additive to the trajectory-mismatch bound. If endpoint encoding errors have norms e_0, e_T, their additional contribution is at most $(e_T + a_{\mathcal{F}} e_0)/(1 - a_{\mathcal{F}})$; coefficient quantization and arithmetic rounding must also be included when used. At 16 bits the measured seed error is 1.0×10^{-4} on Fashion-MNIST and 0.10 on CIFAR-10, within the corresponding bounds of 1.7×10^{-2} and 0.30.

G Experimental Details

G.1 Optimizer Hyperparameters Across zkFU Stages

All optimization stages of zkFU use SGD in the reported experiments. Table 9 lists the exact settings of every stage used for the main results. Identical values are used for Fashion-MNIST with the CNN and CIFAR-10/100 with ResNet-18 unless stated otherwise. TCI selects the scale of the sparse displacement; the subsequent repair uses a lower learning rate with the optimizer settings below.

Repair learning rate. We use $\eta_r = 5 \times 10^{-3}$, one sixth of the training learning rate, for a single retained-data repair epoch, which together with the proximal anchor $\lambda_{\text{prox}} = 0.1$ and FDS-ADR shielding strength $\lambda_{\text{sh}} = 1.0$ keeps repair local to the calibrated seed.

Table 9: Optimizer settings for each SGD-driven loop in zkFU. Symbols are η for learning rate, μ for momentum, λ_{wd} for ℓ_2 weight decay, B for batch size, E for epochs, τ_{kd} for KD temperature, λ_{kd} for KD weight, λ_{pull} for the anti-pullback weight, λ_{prox} for proximal-anchor weight, and λ_{sh} for the FDS-ADR shielding strength of equation 13. B_{RCP} and B_F are the numbers of retained batches used to estimate g_R and F_R.

Stage	Opt.	η	μ	λ_{wd}	Batch / Epochs	Extra terms
FedAvg training (client local updates)	SGD	0.03	0.9	5×10^{-4}	B=256, E=5	none
RCP / FTD	none	–	–	–	–	closed-form projection and diagonal-Fisher shrinkage, B_{RCP}=2, B_F=4, λ_F=10
FSTM	none	–	–	–	–	top-ρ selection over $\lvert\Delta_{\mathrm{trust}}\rvert/(\varepsilon_F + F_R)$ with ρ=0.20
TCI	none	–	–	–	–	grid search over 7 values of β in $[0.25, 1.00]$ with τ_{kd}=2.0, λ_{kd}=0.5, and λ_{pull}=0.05
FDS-ADR repair	SGD	5×10^{-3}	0.9	5×10^{-4}	B=256, E=1	τ_{kd}=2.0, λ_{kd}=0.5, λ_{prox}=0.1, λ_{sh}=1.0

Evaluation protocol. We use Dirichlet $\alpha = 0.5$ as a moderate non-IID regime, $N = 20$ clients and single-client deletion for comparability with prior client-level FU protocols, and full participation to isolate deletion quality. The budget $T = 100$, $E = 5$, $B = 256$ keeps matched retraining feasible for every configuration. Partial participation, selected-round deletion, and cumulative deletion are evaluated separately in Appendices C and H.

G.2 BASELINES

We compare zkFU-TraceTrust with retraining references, simple trace-based variants, representative client-level federated unlearning methods, and an empirical verification method.

Retraining references and simple variants. *Federated Retrain* retrains the FL model from scratch using only retained clients and serves as the federated unlearning reference. *Direct Accumulation* directly subtracts the stored forgotten-client trace from θ_T without retained-mass correction or recovery. It isolates subtraction and does not reproduce the complete subtraction-plus-distillation method of Wu et al. (2022).

Federated unlearning methods. *FedEraser* reconstructs an unlearned model from historical updates (Liu et al., 2021). *PGD* treats unlearning as a constrained optimization problem (Halimi et al., 2022). *FedRecovery* performs recovery using weighted residual information with privacy protection (Zhang et al., 2023). *ConDa* dampens parameters associated with the forgotten client (Chundawat et al., 2024). *SIFU* performs sequential informed unlearning (Fraboni et al., 2024). *FedAU* uses an auxiliary unlearning module (Gu et al., 2024). *FCU* uses contrastive client unlearning with memory preservation (Deng et al., 2024). *FedOSD* mitigates conflicts between forgetting and retaining objectives (Pan et al., 2025). *FUSED* overwrites forgotten knowledge through sparse adapters (Zhong et al., 2025). *uS2U* is the unlearning method of the VeriFi verification framework (Gao et al., 2024).

G.3 IMPLEMENTATION DETAILS

Retained-client work in recovery. TraceTrust uses six retained mini-batches for the gradient and diagonal-Fisher summaries, evaluates seven candidate scales during calibration, and runs one repair epoch at the reduced learning rate in Table 9. Calibration evaluations are additional to the statistics and repair budget. Unlearning time in Table 1 measures the full recovery computation in the simulation. Stage 1 certificate correctness is independent of recovery participation. Appendix H.6 evaluates recovery-cohort and dropout sensitivity.

Simulation of the recovery stage. The recovery protocol of Section 6.3 is federated. Retained clients compute the summaries (g_R, F_R) and the repair gradients on their own data, and the server aggregates them with the normalized weights $\bar{a}_j$ of equation 12. As is common in federated learning simulations, we simulate the aggregated retained objective in a single process. We compute the summaries and repair updates from pooled retained mini-batches, sampling clients in proportion to local dataset size. One repair epoch visits each client's data once. The privacy boundary stated in Section 4 is a property of the protocol, and the simulation changes no quantity that enters the certificate, since Stage 1 is computed from the committed trace alone.

H ADDITIONAL RESULTS

Results below use the reference models and seed counts stated for each experiment; comparisons are made within each table.

H.1 TRAINING RULES BEYOND FEDAVG

TCGS depends only on the server rule. Let the rule be linear in the submitted updates with public coefficients ω_i^t determined by the aggregation weights and public training states, so that $\theta_T = \theta_0 + \sum_t \sum_i \omega_i^t \Delta_i^t$, and let retraining without $\mathcal{F}$ multiply every retained coefficient by one public scalar ν. With per-client aggregates $A_i = \sum_t \omega_i^t \Delta_i^t$, the retained-only rule evaluated on the observed trajectory is

$$\theta_{\mathrm{tcgs}} = \theta_0 + \nu \sum_{j \notin \mathcal{F}} A_j = \theta_0 + \nu\big(\theta_T - \theta_0 - W_{\mathcal{F}}\big), \qquad W_{\mathcal{F}} = \sum_{i \in \mathcal{F}} A_i, \tag{26}$$

which is equation 4 with $a_{\mathcal{F}}$ replaced by $\tilde{a}_{\mathcal{F}} = 1 - 1/\nu$. The verifier aggregates the forgotten-client commitments with ω_i^t in place of a_i, so the circuit and proof size are unchanged, and the error decomposition equation 20 holds with $\nu\omega_j^t$ in place of $\bar{a}_j$, so the only approximation is again trajectory

mismatch. FedProx (Li et al., 2020) changes only the client objective, so $\omega_i^t = a_i$ and $\nu = 1/(1-a_{\mathcal{F}})$. Server momentum (FedAvgM (Hsu et al., 2019)) with buffer $v \leftarrow mv + (1-m)\sum_i a_i \Delta_i^t$ and step $\theta \leftarrow \theta + v$ gives $\omega_i^t = a_i(1-m^{T-t+1})$ and again $\nu = 1/(1-a_{\mathcal{F}})$, and splitting the buffer by client lets the server keep one running A_i per client. FedNova (Wang et al., 2020) normalizes each update by the client's effective number of local steps τ_i, so $\omega_i^t = a_i \tau_{\text{eff}}/\tau_i$ with $\tau_{\text{eff}} = \sum_i a_i \tau_i$ and $\nu = \sum_{j \notin \mathcal{F}} a_j \tau_j / \big((1-a_{\mathcal{F}})^2 \tau_{\text{eff}}\big)$. Both rules act on the trainable parameters, as a server optimizer and the FedNova optimizer do, and average batch-normalization statistics as in FedAvg, whose ν is $1/(1-a_{\mathcal{F}})$. TCGS then applies equation 26 per block, and committing the statistics in their own blocks lets the verifier use each block's public coefficients. When client updates do not depend on the model state, equation 26 equals retained-only training exactly under all three rules.

We rerun Fashion-MNIST and CIFAR-10 with FedProx local training ($\mu = 0.01$), keeping server aggregation, TCGS, and TraceTrust unchanged (Table 10). On Fashion-MNIST, a FedAvg control under the same 3 seeds shows that the deployed model has similar performance under both local objectives: retained accuracy, test accuracy, Δ_f, MIA Gap, and KL to the matching retrain oracle agree within one standard deviation, and the raw seed is slightly closer to retraining under FedProx, consistent with the proximal term limiting client drift and hence the trajectory mismatch of Theorem 1. On CIFAR-10, FedProx stabilizes the retrain oracle across seeds, raising its retained accuracy from 79.39 ± 10.86 to 88.40 ± 2.72, while zkFU keeps 93.36 (92.81 under FedAvg) and moves closer to its oracle, with KL 0.142 against 0.254 and Δ_f 4.78 against 14.03 pp. Under FedAvgM ($m = 0.9$) and FedNova, TCGS uses the coefficients above while TraceTrust is unchanged. On Fashion-MNIST, zkFU again reaches its retrain oracle, with retained accuracy 91.05 against 90.51 under FedAvgM and 92.29 against 90.91 under FedNova at KL at most 0.040, while the raw seed stays far from retraining with KL 6.73 and 8.47, the same pattern as under FedAvg. On CIFAR-10, the retrain oracle is again the weaker model, with retained accuracy 78.67 against 86.23 for zkFU under FedAvgM and 80.57 against 92.87 under FedNova, at KL at most 0.286, while the raw seed has KL 11.51 and 12.97.

Table 10: FedAvg, FedProx ($\mu = 0.01$), FedAvgM ($m = 0.9$), and FedNova training, mean $\pm$ std over the same 3 seeds. Each row is compared with the retrain oracle trained under the same rule. Accuracies are in percent; Δ_f and MIA Gap are in percentage points.

Training	Model	Acc_r	Test acc.	Δ_f	MIA Gap	KL	ℓ_2
Fashion-MNIST / CNN							
	Retrain	$91.79_{\pm 0.53}$	$88.77_{\pm 0.39}$	0	$0.57_{\pm 0.35}$	0	0
FedAvg	TCGS seed	$36.54_{\pm 13.63}$	$34.68_{\pm 13.19}$	$80.95_{\pm 5.62}$	$1.05_{\pm 0.54}$	$9.19_{\pm 5.91}$	$5.18_{\pm 0.26}$
	zkFU	$92.44_{\pm 0.60}$	$89.13_{\pm 0.40}$	$1.67_{\pm 0.65}$	$0.61_{\pm 0.29}$	$0.032_{\pm 0.003}$	$4.18_{\pm 0.28}$
	Retrain	$91.54_{\pm 0.70}$	$88.74_{\pm 0.41}$	0	$1.22_{\pm 1.26}$	0	0
FedProx	TCGS seed	$39.92_{\pm 14.50}$	$38.36_{\pm 14.07}$	$80.33_{\pm 6.47}$	$1.64_{\pm 0.79}$	$8.19_{\pm 4.66}$	$4.90_{\pm 0.81}$
	zkFU	$92.40_{\pm 0.55}$	$89.16_{\pm 0.19}$	$1.76_{\pm 1.04}$	$0.87_{\pm 0.52}$	$0.030_{\pm 0.008}$	$3.96_{\pm 0.78}$
	Retrain	$90.51_{\pm 0.54}$	$88.52_{\pm 0.17}$	0	$0.37_{\pm 0.16}$	0	0
FedAvgM	TCGS seed	$39.69_{\pm 4.07}$	$38.05_{\pm 4.55}$	$79.40_{\pm 3.53}$	$1.88_{\pm 2.21}$	$6.73_{\pm 2.63}$	$4.30_{\pm 0.10}$
	zkFU	$91.05_{\pm 0.53}$	$88.79_{\pm 0.36}$	$1.32_{\pm 0.46}$	$0.62_{\pm 0.60}$	$0.020_{\pm 0.003}$	$3.37_{\pm 0.06}$
	Retrain	$90.91_{\pm 0.88}$	$88.04_{\pm 0.93}$	0	$0.15_{\pm 0.14}$	0	0
FedNova	TCGS seed	$37.52_{\pm 11.75}$	$35.86_{\pm 10.95}$	$81.81_{\pm 6.06}$	$1.22_{\pm 0.58}$	$8.47_{\pm 3.51}$	$5.73_{\pm 0.64}$
	zkFU	$92.29_{\pm 0.24}$	$89.11_{\pm 0.27}$	$1.66_{\pm 0.66}$	$0.74_{\pm 0.63}$	$0.040_{\pm 0.015}$	$4.58_{\pm 0.59}$
CIFAR-10 / ResNet-18							
	Retrain	$79.39_{\pm 10.86}$	$74.99_{\pm 10.23}$	0	$0.96_{\pm 0.62}$	0	0
FedAvg	TCGS seed	$23.42_{\pm 11.25}$	$22.33_{\pm 11.05}$	$74.02_{\pm 14.36}$	$2.65_{\pm 2.55}$	$11.88_{\pm 4.45}$	$61.18_{\pm 21.07}$
	zkFU	$92.81_{\pm 1.04}$	$85.72_{\pm 1.00}$	$14.03_{\pm 13.91}$	$1.76_{\pm 0.90}$	$0.254_{\pm 0.163}$	$19.08_{\pm 5.44}$
	Retrain	$88.40_{\pm 2.72}$	$83.42_{\pm 2.30}$	0	$0.20_{\pm 0.28}$	0	0
FedProx	TCGS seed	$23.88_{\pm 10.52}$	$22.51_{\pm 10.22}$	$84.64_{\pm 1.27}$	$0.78_{\pm 0.80}$	$11.81_{\pm 5.16}$	$56.81_{\pm 18.66}$
	zkFU	$93.36_{\pm 1.01}$	$86.74_{\pm 0.72}$	$4.78_{\pm 2.69}$	$1.25_{\pm 0.96}$	$0.142_{\pm 0.042}$	$18.02_{\pm 5.17}$
	Retrain	$78.67_{\pm 5.86}$	$75.94_{\pm 5.47}$	0	$1.60_{\pm 0.44}$	0	0
FedAvgM	TCGS seed	$23.96_{\pm 10.57}$	$23.22_{\pm 10.36}$	$76.00_{\pm 7.65}$	$0.85_{\pm 0.83}$	$11.51_{\pm 5.50}$	$75.47_{\pm 51.13}$
	zkFU	$86.23_{\pm 3.42}$	$82.05_{\pm 2.75}$	$8.24_{\pm 2.79}$	$0.61_{\pm 0.34}$	$0.113_{\pm 0.027}$	$21.04_{\pm 6.10}$
	Retrain	$80.57_{\pm 3.01}$	$76.11_{\pm 2.47}$	0	$0.89_{\pm 0.84}$	0	0
FedNova	TCGS seed	$22.98_{\pm 10.17}$	$21.42_{\pm 9.84}$	$76.75_{\pm 6.62}$	$0.90_{\pm 0.46}$	$12.97_{\pm 5.14}$	$56.43_{\pm 18.93}$
	zkFU	$92.87_{\pm 0.69}$	$85.43_{\pm 0.99}$	$11.91_{\pm 10.02}$	$0.53_{\pm 0.17}$	$0.286_{\pm 0.156}$	$18.11_{\pm 5.61}$

H.2 Federation-Size Scaling

Table 11 reports single-client deletion on Fashion-MNIST for $N \in \{5, 10, 50\}$, complementing the $N = 20$ setting of Table 1. Figure 2(a) plots the retained accuracy of the trace-based methods. At $N = 50$, zkFU-TraceTrust nearly matches the best retained accuracy, 89.73 versus 89.78, while achieving the best $\Delta_f = 0.83$. The forgotten-client gap of zkFU-TraceTrust decreases from 5.40 at $N = 5$ to 0.83 at $N = 50$.

Table 11: Scaling with federation size under single-client deletion on Fashion-MNIST. Federated Retrain is the unranked reference. Best and second-best distinct displayed means are in red and light red, with ties sharing a rank.

Method	$N = 5$			$N = 10$			$N = 50$		
	$\mathrm{Acc}_r \uparrow$	$\Delta_f \downarrow$	KL $\downarrow$	$\mathrm{Acc}_r \uparrow$	$\Delta_f \downarrow$	KL $\downarrow$	$\mathrm{Acc}_r \uparrow$	$\Delta_f \downarrow$	KL $\downarrow$
Federated Retrain	$94.57_{\pm 1.82}$	$0.00_{\pm 0.00}$	$0.0000_{\pm 0.0000}$	$93.63_{\pm 1.33}$	$0.00_{\pm 0.00}$	$0.0000_{\pm 0.0000}$	$88.98_{\pm 0.29}$	$0.00_{\pm 0.00}$	$0.0000_{\pm 0.0000}$
PGD (ICML Workshop'22)	$94.62_{\pm 1.72}$	$3.66_{\pm 2.52}$	$0.2244_{\pm 0.0284}$	$93.85_{\pm 1.42}$	$4.04_{\pm 2.53}$	$0.1139_{\pm 0.0473}$	$89.57_{\pm 0.19}$	$1.35_{\pm 1.64}$	$0.0119_{\pm 0.0062}$
FCU (MICCAI'24)	$94.42_{\pm 1.68}$	$2.02_{\pm 1.21}$	$0.2426_{\pm 0.0333}$	$93.60_{\pm 1.91}$	$2.26_{\pm 1.69}$	$0.1217_{\pm 0.0432}$	$89.78_{\pm 0.19}$	$1.37_{\pm 0.85}$	$0.0175_{\pm 0.0127}$
FedAU (IJCAI'24)	$49.83_{\pm 13.32}$	$80.02_{\pm 2.64}$	$3.5546_{\pm 0.7179}$	$51.16_{\pm 5.34}$	$78.28_{\pm 5.10}$	$2.6289_{\pm 0.8001}$	$67.25_{\pm 3.26}$	$66.88_{\pm 10.81}$	$1.0658_{\pm 0.3925}$
FedOSD (AAAI'25)	$89.69_{\pm 3.69}$	$44.32_{\pm 23.58}$	$0.5975_{\pm 0.2103}$	$92.89_{\pm 1.54}$	$3.32_{\pm 1.40}$	$0.1521_{\pm 0.0590}$	$89.08_{\pm 0.19}$	$0.98_{\pm 0.48}$	$0.0045_{\pm 0.0021}$
FUSED (CVPR'25)	$94.66_{\pm 1.68}$	$6.44_{\pm 3.88}$	$0.2589_{\pm 0.0346}$	$93.67_{\pm 1.42}$	$5.17_{\pm 2.76}$	$0.1178_{\pm 0.0510}$	$89.32_{\pm 0.16}$	$1.64_{\pm 0.91}$	$0.0042_{\pm 0.0015}$
Direct Accumulation	$15.10_{\pm 4.39}$	$85.62_{\pm 2.54}$	$23.9288_{\pm 1.1647}$	$16.98_{\pm 9.72}$	$86.74_{\pm 4.53}$	$22.2678_{\pm 4.8159}$	$43.29_{\pm 13.85}$	$73.66_{\pm 10.46}$	$5.7062_{\pm 3.9444}$
FedEraser (IWQoS'21)	$92.11_{\pm 1.69}$	$2.02_{\pm 1.75}$	$0.1410_{\pm 0.0086}$	$88.15_{\pm 1.96}$	$3.20_{\pm 2.89}$	$0.1751_{\pm 0.0235}$	$74.01_{\pm 0.58}$	$14.55_{\pm 5.45}$	$0.3624_{\pm 0.0433}$
FedRecovery (TIFS'23)	$69.39_{\pm 5.85}$	$6.16_{\pm 3.03}$	$0.8982_{\pm 0.1288}$	$85.25_{\pm 2.43}$	$1.36_{\pm 0.80}$	$0.3309_{\pm 0.1102}$	$85.95_{\pm 1.51}$	$1.49_{\pm 1.40}$	$0.0890_{\pm 0.0292}$
ConDa (ArXiv'24)	$93.72_{\pm 1.84}$	$7.45_{\pm 4.46}$	$0.2871_{\pm 0.0495}$	$93.16_{\pm 1.52}$	$5.61_{\pm 2.57}$	$0.1210_{\pm 0.0538}$	$89.06_{\pm 0.20}$	$0.86_{\pm 0.46}$	$0.0030_{\pm 0.0012}$
SIFU (AISTATS'24)	$88.64_{\pm 0.68}$	$3.60_{\pm 3.55}$	$0.2100_{\pm 0.0107}$	$85.45_{\pm 2.43}$	$4.33_{\pm 2.33}$	$0.2385_{\pm 0.0136}$	$74.43_{\pm 0.84}$	$13.98_{\pm 7.16}$	$0.3540_{\pm 0.0104}$
uS2U (TDSC'24)	$92.44_{\pm 2.21}$	$8.15_{\pm 4.18}$	$0.2619_{\pm 0.0541}$	$91.90_{\pm 1.38}$	$4.67_{\pm 2.57}$	$0.1016_{\pm 0.0392}$	$86.43_{\pm 0.42}$	$1.41_{\pm 1.09}$	$0.0509_{\pm 0.0046}$
zkFU-TraceTrust (Ours)	$94.68_{\pm 1.70}$	$5.40_{\pm 3.39}$	$0.2616_{\pm 0.0346}$	$93.97_{\pm 1.27}$	$3.36_{\pm 1.89}$	$0.1139_{\pm 0.0500}$	$89.73_{\pm 0.13}$	$0.83_{\pm 0.46}$	$0.0096_{\pm 0.0031}$

H.3 Auxiliary Diagnostics

Additional diagnostics beyond the main metrics. Table 12 shows that the advantage of zkFU-TraceTrust is not tied to a particular choice of main metrics. On FMNIST, zkFU-TraceTrust has the smallest ℓ_2 distance to Federated Retrain (4.91), closely followed by PGD, FedOSD, FUSED, and ConDa (4.94–4.95) and has the second-best Jensen–Shannon (JS) divergence, while staying in the top TA group. Thus, TraceTrust does more than recover label accuracy after TCGS. It also keeps the recovered model close to the retraining reference in parameter space and predictive distribution. On CIFAR-10, PGD attains the highest TA and FCU attains the lowest JS, but zkFU-TraceTrust remains competitive on both while being tied to a compact TCGS certificate. On CIFAR-100, zkFU-TraceTrust achieves the second-best TA and JS, indicating that the recovery pipeline remains effective when the label space is larger and the retained-only counterfactual is harder to approximate.

Trace reuse needs retained-aware correction. Direct Accumulation follows the stored trace most literally, but its TA collapses and its JS divergence is the largest on all datasets. FedEraser, FedRecovery, and SIFU further show that historical update reuse does not automatically produce a retraining-consistent model, especially on CIFAR-10 and CIFAR-100 where their ℓ_2 and JS distances become large. TraceTrust combines this historical signal with retained-data sensitivity and repair to recover a useful model.

Federation-size diagnostics. Table 13 shows that zkFU-TraceTrust remains in the top TA group across $N = 5$, $N = 10$, and $N = 50$. At $N = 50$, it obtains the best TA and the second-best MIA Gap, while several trace reuse baselines lose substantial utility.

Table 12: Auxiliary diagnostics under Dirichlet $\alpha = 0.5$, complementing Table 1: test accuracy TA (%), parameter distance ℓ_2, and predictive Jensen–Shannon divergence JS to Federated Retrain on the test set. Retrain is the unranked reference; best and second-best distinct displayed means (ties shared) in **red** and **light red**.

Method	FMNIST ($N = 20$)			CIFAR-10 ($N = 20$)			CIFAR-100 ($N = 20$)		
	TA ↑	ℓ_2 ↓	JS ↓	TA ↑	ℓ_2 ↓	JS ↓	TA ↑	ℓ_2 ↓	JS ↓
Federated Retrain	$88.89_{\pm 0.46}$	$0.00_{\pm 0.00}$	$0.000_{\pm 0.000}$	$76.42_{\pm 3.17}$	$0.00_{\pm 0.00}$	$0.000_{\pm 0.000}$	$58.39_{\pm 0.55}$	$0.00_{\pm 0.00}$	$0.000_{\pm 0.000}$
PGD (ICML Workshop'22)	**$89.35_{\pm 0.52}$**	*$4.94_{\pm 0.90}$*	*$0.010_{\pm 0.002}$*	**$87.98_{\pm 0.50}$**	*$23.12_{\pm 2.72}$*	$0.076_{\pm 0.019}$	**$61.06_{\pm 0.89}$**	$29.61_{\pm 1.94}$	$0.091_{\pm 0.007}$
FCU (MICCAI'24)	$89.07_{\pm 0.35}$	$5.38_{\pm 0.93}$	$0.011_{\pm 0.002}$	$81.22_{\pm 6.56}$	$25.11_{\pm 2.48}$	**$0.048_{\pm 0.020}$**	$59.95_{\pm 0.81}$	$32.15_{\pm 1.96}$	$0.089_{\pm 0.005}$
FedAU (IJCAI'24)	$59.43_{\pm 10.09}$	$7.52_{\pm 0.66}$	$0.230_{\pm 0.061}$	$61.12_{\pm 8.59}$	$26.26_{\pm 2.22}$	$0.148_{\pm 0.038}$	$54.67_{\pm 0.68}$	$31.98_{\pm 1.72}$	$0.120_{\pm 0.005}$
FedOSD (AAAI'25)	$88.45_{\pm 0.38}$	$4.95_{\pm 0.90}$	$0.013_{\pm 0.006}$	$85.27_{\pm 0.53}$	$23.26_{\pm 2.71}$	$0.090_{\pm 0.015}$	$54.92_{\pm 0.62}$	$29.75_{\pm 1.92}$	$0.127_{\pm 0.005}$
FUSED (CVPR'25)	*$89.29_{\pm 0.49}$*	$4.95_{\pm 0.89}$	*$0.010_{\pm 0.002}$*	*$87.15_{\pm 0.99}$*	**$22.74_{\pm 2.56}$**	$0.070_{\pm 0.014}$	$59.71_{\pm 0.17}$	**$28.92_{\pm 1.92}$**	$0.084_{\pm 0.005}$
Direct Accumulation	$36.32_{\pm 15.47}$	$5.97_{\pm 1.04}$	$0.405_{\pm 0.123}$	$22.89_{\pm 6.24}$	$55.28_{\pm 19.62}$	$0.465_{\pm 0.089}$	$10.05_{\pm 2.11}$	$40.31_{\pm 3.92}$	$0.559_{\pm 0.023}$
FedEraser (IWQoS'21)	$82.52_{\pm 2.45}$	$10.81_{\pm 0.36}$	$0.072_{\pm 0.011}$	$51.70_{\pm 27.55}$	$163.57_{\pm 9.32}$	$0.263_{\pm 0.211}$	$32.38_{\pm 2.28}$	$78.23_{\pm 0.94}$	$0.302_{\pm 0.023}$
FedRecovery (TIFS'23)	$85.12_{\pm 1.94}$	$7.27_{\pm 0.59}$	$0.042_{\pm 0.015}$	$29.20_{\pm 8.69}$	$89.86_{\pm 0.63}$	$0.405_{\pm 0.083}$	$22.31_{\pm 2.10}$	$91.30_{\pm 0.49}$	$0.436_{\pm 0.017}$
ConDa (ArXiv'24)	$88.80_{\pm 0.57}$	$4.95_{\pm 0.90}$	**$0.009_{\pm 0.002}$**	$63.11_{\pm 35.13}$	$23.45_{\pm 2.88}$	$0.194_{\pm 0.301}$	$58.98_{\pm 0.65}$	*$29.44_{\pm 1.89}$*	**$0.076_{\pm 0.003}$**
SIFU (AISTATS'24)	$80.04_{\pm 1.14}$	$12.05_{\pm 0.29}$	$0.080_{\pm 0.005}$	$63.69_{\pm 1.59}$	$88.24_{\pm 1.19}$	$0.144_{\pm 0.005}$	$27.39_{\pm 0.62}$	$92.50_{\pm 0.60}$	$0.342_{\pm 0.011}$
uS2U (TDSC'24)	$88.51_{\pm 0.42}$	$5.77_{\pm 0.52}$	$0.013_{\pm 0.002}$	$80.38_{\pm 1.69}$	$61.62_{\pm 4.84}$	$0.104_{\pm 0.012}$	$50.04_{\pm 0.92}$	$45.54_{\pm 0.92}$	$0.141_{\pm 0.009}$
zkFU-TraceTrust (Ours)	$89.27_{\pm 0.52}$	*$4.91_{\pm 0.90}$*	*$0.010_{\pm 0.002}$*	$86.52_{\pm 1.59}$	$24.15_{\pm 2.45}$	*$0.051_{\pm 0.012}$*	*$60.81_{\pm 0.69}$*	$29.98_{\pm 1.90}$	*$0.082_{\pm 0.003}$*

Table 13: Fashion-MNIST federation-size diagnostics complementing Table 11: test accuracy TA in percent and MIA Gap in percentage points. Retrain is the unranked reference; best and second-best in **red** and **light red**. At $N = 5$ one run whose retrain reference did not converge is excluded.

Method	$N = 5$		$N = 10$		$N = 50$	
	TA ↑	MIA Gap ↓	TA ↑	MIA Gap ↓	TA ↑	MIA Gap ↓
Federated Retrain	$87.48_{\pm 0.87}$	$0.55_{\pm 0.35}$	$89.01_{\pm 0.31}$	$0.64_{\pm 0.80}$	$87.93_{\pm 0.14}$	$0.74_{\pm 0.60}$
PGD (ICML Workshop'22)	$88.22_{\pm 0.62}$	$0.72_{\pm 0.89}$	$89.29_{\pm 0.34}$	$0.94_{\pm 0.39}$	$88.40_{\pm 0.31}$	**$0.44_{\pm 0.14}$**
FCU (MICCAI'24)	$87.89_{\pm 0.66}$	*$0.27_{\pm 0.32}$*	$89.05_{\pm 0.81}$	$0.40_{\pm 0.24}$	*$88.42_{\pm 0.28}$*	$0.90_{\pm 0.81}$
FedAU (IJCAI'24)	$38.93_{\pm 11.69}$	$3.29_{\pm 3.11}$	$44.64_{\pm 4.86}$	$3.46_{\pm 1.38}$	$65.55_{\pm 3.03}$	$1.18_{\pm 1.01}$
FedOSD (AAAI'25)	$75.12_{\pm 6.77}$	$1.45_{\pm 0.60}$	$87.99_{\pm 0.47}$	*$0.35_{\pm 0.17}$*	$87.95_{\pm 0.22}$	$0.95_{\pm 0.57}$
FUSED (CVPR'25)	**$88.83_{\pm 0.48}$**	$0.65_{\pm 0.13}$	**$89.40_{\pm 0.46}$**	$0.67_{\pm 0.70}$	$88.29_{\pm 0.28}$	$1.59_{\pm 0.75}$
Direct Accumulation	$12.14_{\pm 3.91}$	**$0.05_{\pm 0.07}$**	$15.38_{\pm 9.43}$	**$0.16_{\pm 0.34}$**	$42.07_{\pm 13.47}$	$2.46_{\pm 1.58}$
FedEraser (IWQoS'21)	$88.23_{\pm 0.90}$	$0.73_{\pm 0.35}$	$87.03_{\pm 1.13}$	$0.92_{\pm 0.74}$	$74.08_{\pm 0.50}$	$2.91_{\pm 2.65}$
FedRecovery (TIFS'23)	$69.43_{\pm 4.14}$	$1.52_{\pm 2.28}$	$83.06_{\pm 1.92}$	$0.89_{\pm 0.35}$	$85.15_{\pm 1.35}$	$1.33_{\pm 1.16}$
ConDa (ArXiv'24)	$88.23_{\pm 0.98}$	$1.12_{\pm 0.20}$	$89.24_{\pm 0.46}$	$1.15_{\pm 0.88}$	$88.01_{\pm 0.16}$	$1.57_{\pm 1.04}$
SIFU (AISTATS'24)	$86.29_{\pm 1.18}$	$1.76_{\pm 1.15}$	$84.89_{\pm 1.50}$	$2.49_{\pm 1.60}$	$74.40_{\pm 1.05}$	$1.83_{\pm 1.37}$
uS2U (TDSC'24)	$88.49_{\pm 0.91}$	$0.42_{\pm 0.38}$	$89.19_{\pm 0.54}$	$0.79_{\pm 0.57}$	$85.93_{\pm 0.44}$	$1.34_{\pm 0.55}$
zkFU-TraceTrust (Ours)	*$88.71_{\pm 0.68}$*	$0.62_{\pm 0.24}$	*$89.34_{\pm 0.38}$*	$0.61_{\pm 0.53}$	**$88.47_{\pm 0.33}$**	*$0.82_{\pm 0.37}$*

H.4 Statistical Membership-Inference Evaluation

Table 3(b) reports the shadow-attacker MIA Gap over 20 seeds with Welch's test against the no-unlearning model and TOST equivalence against Federated Retrain. On CIFAR-10 zkFU differs from retrain by less than 0.3 pp, about half the oracle's own standard deviation. Under the loss-threshold attack of Yeom et al. (2018), over the same 20 seeds on Fashion-MNIST, the original model, retraining, and zkFU differ by at most 1.0 pp in MIA Gap against seed standard deviations of 3.6 to 3.9 pp, so that attack does not resolve them. The shadow-attacker gap, with seed standard deviations below 1 pp, is the measure with dynamic range here, and it gives separation from no unlearning together with equivalence to retraining on both datasets.

Per-example LiRA. We run the offline variant of the likelihood-ratio attack (LiRA) (Carlini et al., 2022). Each example is scored by the z-score of the target model's confidence $\phi(x) = \log(p_y/(1-p_y))$, where p_y is the predicted probability of the true label y, against a per-example Gaussian fitted on seed-matched retrain reference models that were never trained on the forgotten client's data or on test data, with identical calibration for members and non-members. Table 14 reports the area under the receiver operating characteristic curve (AUC) and the true-positive rate at 0.1% and 1% false-positive rate, where chance equals the false-positive rate. A client-level paired t-test over 19 seeds on Fashion-MNIST and 20 on CIFAR-10 gives $t = 0.66$ and $t = -0.33$ for zkFU against retrain, indicating no statistically resolved difference under this attack. Every AUC, including that of the original model, lies within 0.01 of chance, and no true-positive rate exceeds 1.31 times its false-positive rate: this LiRA configuration provides little separation between original, retrained, and unlearned models.

Table 14: Offline per-example LiRA. AUC 0.5 is chance, and the chance true-positive rate (TPR) equals the false-positive rate (FPR).

Setting	Model	AUC	TPR at 0.1% FPR	TPR at 1% FPR
Fashion-MNIST, CNN	No unlearning	0.504	0.04%	0.30%
	Retrain	0.493	0.05%	0.80%
	zkFU	0.506	0.01%	0.22%
CIFAR-10, ResNet-18	No unlearning	0.500	0.09%	0.74%
	Retrain	0.508	0.02%	1.13%
	zkFU	0.493	0.08%	1.31%

H.5 Sequential Deletion

Table 15 gives all five cumulative requests on both datasets, whose MIA Gap Table 3(b) reports. Request k is certified against the original trace with $\mathcal{F}(k)$ and compared with a retrain oracle trained without $\mathcal{F}(k)$. Proof size and verification time do not depend on k. With the Fashion-MNIST certificate laid out as a single 2^{20}-row block, every request yields a $9{,}632$-byte proof verified in 0.12 s. On CIFAR-10, Δ_f tracks the margin by which zkFU's test accuracy exceeds the oracle while the MIA Gap remains close to retraining across the measured requests, showing that accuracy gaps and this membership attack capture different aspects of recovery. Under the same cumulative-deletion protocol over 3 seeds (Table 16), zkFU has the highest test accuracy at every reported k. SIFU is 0.79 to 1.30 pp lower and closer to the oracle in Δ_f, and FedEraser is 7.55 to 7.98 pp lower with a larger Δ_f.

Table 15: Five cumulative single-client deletions, all requests, mean ± std over 5 seeds. The reference at each k is a retrain oracle for the cumulative set $\mathcal{F}(k)$. The no-unlearning row is the original model θ_T evaluated against the same reference. MIA Gap in pp.

k	Model	Fashion-MNIST / CNN				CIFAR-10 / ResNet-18			
		Test acc.	Δ_f	MIA Gap	KL	Test acc.	Δ_f	MIA Gap	KL
1	No unlearning	$89.09_{\pm 0.47}$	$3.21_{\pm 1.30}$	$1.45_{\pm 0.63}$	–	$78.63_{\pm 2.52}$	$8.98_{\pm 5.85}$	$2.32_{\pm 1.55}$	–
	Retrain oracle	$88.90_{\pm 0.33}$	0	$0.52_{\pm 0.32}$	0	$77.71_{\pm 3.89}$	0	$0.57_{\pm 0.69}$	0
	zkFU	$89.37_{\pm 0.44}$	$1.34_{\pm 0.75}$	$0.59_{\pm 0.39}$	$0.045_{\pm 0.017}$	$85.89_{\pm 0.55}$	$12.37_{\pm 10.31}$	$1.05_{\pm 1.09}$	$0.212_{\pm 0.056}$
2	No unlearning	$89.09_{\pm 0.47}$	$3.69_{\pm 1.43}$	$1.77_{\pm 0.69}$	–	$78.63_{\pm 2.52}$	$5.04_{\pm 4.23}$	$2.49_{\pm 0.44}$	–
	Retrain oracle	$88.75_{\pm 0.50}$	0	$0.65_{\pm 0.69}$	0	$79.63_{\pm 2.00}$	0	$0.55_{\pm 0.13}$	0
	zkFU	$89.44_{\pm 0.43}$	$2.50_{\pm 1.28}$	$0.59_{\pm 0.45}$	$0.056_{\pm 0.014}$	$86.00_{\pm 0.70}$	$8.34_{\pm 4.76}$	$0.47_{\pm 0.36}$	$0.169_{\pm 0.067}$
3	No unlearning	$89.09_{\pm 0.47}$	$3.90_{\pm 1.36}$	$1.75_{\pm 0.65}$	–	$78.63_{\pm 2.52}$	$3.37_{\pm 1.87}$	$2.63_{\pm 0.49}$	–
	Retrain oracle	$88.51_{\pm 0.48}$	0	$0.25_{\pm 0.25}$	0	$82.51_{\pm 0.88}$	0	$0.29_{\pm 0.18}$	0
	zkFU	$89.34_{\pm 0.46}$	$2.81_{\pm 0.57}$	$0.55_{\pm 0.19}$	$0.069_{\pm 0.016}$	$86.08_{\pm 0.67}$	$7.70_{\pm 3.62}$	$0.52_{\pm 0.28}$	$0.165_{\pm 0.062}$
4	No unlearning	$89.09_{\pm 0.47}$	$4.22_{\pm 1.27}$	$1.65_{\pm 0.50}$	–	$78.63_{\pm 2.52}$	$2.97_{\pm 2.71}$	$2.83_{\pm 0.55}$	–
	Retrain oracle	$88.38_{\pm 0.59}$	0	$0.44_{\pm 0.36}$	0	$80.87_{\pm 3.15}$	0	$0.64_{\pm 0.42}$	0
	zkFU	$89.32_{\pm 0.47}$	$3.21_{\pm 0.36}$	$0.62_{\pm 0.32}$	$0.084_{\pm 0.015}$	$85.45_{\pm 0.70}$	$7.91_{\pm 2.46}$	$0.72_{\pm 0.26}$	$0.197_{\pm 0.048}$
5	No unlearning	$89.09_{\pm 0.47}$	$4.85_{\pm 1.38}$	$1.97_{\pm 0.51}$	–	$78.63_{\pm 2.52}$	$6.75_{\pm 3.53}$	$2.47_{\pm 0.63}$	–
	Retrain oracle	$88.17_{\pm 0.75}$	0	$0.31_{\pm 0.20}$	0	$76.97_{\pm 3.78}$	0	$0.63_{\pm 0.80}$	0
	zkFU	$89.23_{\pm 0.41}$	$3.67_{\pm 0.70}$	$0.30_{\pm 0.43}$	$0.106_{\pm 0.012}$	$85.56_{\pm 0.80}$	$12.74_{\pm 5.71}$	$0.53_{\pm 0.53}$	$0.249_{\pm 0.076}$

Table 16: Cumulative sequential deletion on Fashion-MNIST against SIFU and FedEraser over 3 seeds. Each cell reports test accuracy and Δ_f against the retrain oracle for $\mathcal{F}(k)$, whose test accuracy is 88.77, 88.23, and 87.81 at $k = 1, 3, 5$.

k	zkFU	SIFU	FedEraser
1	89.15, 1.69	88.36, 0.86	81.17, 4.47
3	89.05, 2.94	87.91, 0.38	81.50, 6.15
5	88.99, 3.66	87.69, 0.25	81.24, 5.12

H.6 Recovery-Stage Participation and Dropout

Table 17 varies the fraction of retained clients invited to the recovery stage and removes 30% of the invited clients either uniformly at random or by largest local dataset. On Fashion-MNIST, a 20% cohort lowers test accuracy by 0.27 pp, 30% random dropout causes no measured loss, and largest-client dropout changes accuracy by 0.05 pp. On CIFAR-10, a 20% cohort and 30% random dropout lower accuracy by 2.71 and 1.39 pp, and the Stage 1 certificate is unchanged across all rows.

Table 17: Deployed-model sensitivity to the recovery cohort and to dropout during the recovery stage, mean ± std over 5 seeds. Cohort is the fraction of retained clients invited to recovery. All rows share the same certified seed.

Dataset	Recovery setting	Test acc.	Δ_f	MIA Gap	KL
Fashion-MNIST	Retrain oracle	$88.90_{\pm 0.33}$	0	$0.52_{\pm 0.32}$	0
	20% cohort	$89.10_{\pm 0.41}$	$1.87_{\pm 0.90}$	$0.60_{\pm 0.63}$	$0.050_{\pm 0.020}$
	50% cohort	$89.28_{\pm 0.44}$	$1.95_{\pm 1.33}$	$0.71_{\pm 0.39}$	$0.045_{\pm 0.016}$
	All invited (default)	$89.37_{\pm 0.44}$	$1.34_{\pm 0.75}$	$0.61_{\pm 0.41}$	$0.045_{\pm 0.017}$
	All invited, 30% random dropout	$89.37_{\pm 0.55}$	$1.30_{\pm 0.57}$	$0.43_{\pm 0.30}$	$0.046_{\pm 0.019}$
	All invited, 30% largest-client dropout	$89.32_{\pm 0.37}$	$1.74_{\pm 0.93}$	$0.73_{\pm 0.42}$	$0.047_{\pm 0.019}$
	50% cohort, 30% largest-client dropout	$89.22_{\pm 0.36}$	$2.05_{\pm 1.33}$	$0.70_{\pm 0.55}$	$0.050_{\pm 0.020}$
CIFAR-10	Retrain oracle	$77.71_{\pm 3.89}$	0	$0.57_{\pm 0.69}$	0
	20% cohort	$83.19_{\pm 1.34}$	$8.07_{\pm 8.55}$	$0.51_{\pm 0.46}$	$0.214_{\pm 0.059}$
	50% cohort	$85.02_{\pm 1.24}$	$11.63_{\pm 10.45}$	$0.89_{\pm 0.78}$	$0.193_{\pm 0.060}$
	All invited (default)	$85.90_{\pm 0.54}$	$12.39_{\pm 10.34}$	$0.92_{\pm 0.83}$	$0.212_{\pm 0.056}$
	All invited, 30% random dropout	$84.51_{\pm 0.86}$	$11.18_{\pm 11.08}$	$0.91_{\pm 0.64}$	$0.197_{\pm 0.064}$

H.7 Hyperparameter and Repair-Budget Sensitivity

Table 18 sweeps the sparsity ratio ρ of equation 9 at $\lambda_F = 10$ and the Fisher damping λ_F of equation 8 at $\rho = 0.2$, and Table 19 the number of FDS-ADR repair epochs. The deployed model is insensitive

to both filter parameters, as test accuracy moves by at most 0.17 pp across ρ and 0.04 pp across two orders of magnitude of λ_F, below the 0.44 pp standard deviation across seeds on Fashion-MNIST. One repair epoch provides the utility gain, raising test accuracy by 2.2 pp on Fashion-MNIST and 3.8 pp on CIFAR-10 and at least halving the KL to retraining. Further epochs increase test accuracy by at most 0.11 pp or decrease it by up to 0.37 pp, and change KL by at most 0.024, while the CIFAR-10 MIA Gap rises from 0.40 after one epoch to 1.43 after five. The one-epoch default therefore captures the utility gain without over-repair.

Table 18: Sensitivity of the deployed model to the sparsity ratio ρ and the Fisher damping λ_F, mean over 5 seeds. Retrain test accuracy is 88.90 on Fashion-MNIST and 77.71 on CIFAR-10. The default is $\rho = 0.2$, $\lambda_F = 10$.

Sweep	Setting	Fashion-MNIST / CNN			CIFAR-10 / ResNet-18		
		Test acc.	Δ_f	KL	Test acc.	Δ_f	KL
ρ ($\lambda_F = 10$)	0.05	89.34	1.62	0.0448	85.92	12.76	0.2120
	0.10	89.36	1.61	0.0449	85.90	12.68	0.2121
	0.20 (default)	89.37	1.34	0.0447	85.90	12.39	0.2121
	0.40	89.36	1.02	0.0443	85.75	11.50	0.2124
λ_F ($\rho = 0.2$)	1	89.33	1.40	0.0447	85.91	12.34	0.2119
	10 (default)	89.37	1.34	0.0447	85.90	12.39	0.2121
	100	89.34	1.38	0.0447	85.90	12.31	0.2122

Table 19: Repair-budget ablation of FDS-ADR, mean $\pm$ std over 5 seeds. Zero epochs is the calibrated model θ_{cal} before repair. The certified seed is identical in every row. Retrain test accuracy is 88.90 on Fashion-MNIST and 74.60 on CIFAR-10 in this experiment.

Repair epochs	Fashion-MNIST / CNN			CIFAR-10 / ResNet-18		
	Test acc.	MIA Gap	KL	Test acc.	MIA Gap	KL
0	$87.17_{\pm 3.42}$	$0.43_{\pm 0.32}$	$0.090_{\pm 0.102}$	$82.54_{\pm 2.90}$	$1.42_{\pm 1.05}$	$0.727_{\pm 0.542}$
1 (default)	$89.37_{\pm 0.44}$	$0.59_{\pm 0.39}$	$0.045_{\pm 0.017}$	$86.29_{\pm 0.52}$	$0.40_{\pm 0.36}$	$0.288_{\pm 0.188}$
2	$89.48_{\pm 0.46}$	$0.57_{\pm 0.67}$	$0.043_{\pm 0.016}$	$86.40_{\pm 0.53}$	$0.68_{\pm 0.67}$	$0.312_{\pm 0.232}$
3	$89.48_{\pm 0.40}$	$0.90_{\pm 0.73}$	$0.043_{\pm 0.017}$	$86.16_{\pm 0.78}$	$0.62_{\pm 0.37}$	$0.290_{\pm 0.209}$
5	$89.45_{\pm 0.37}$	$1.00_{\pm 0.48}$	$0.045_{\pm 0.016}$	$85.92_{\pm 0.43}$	$1.43_{\pm 0.54}$	$0.294_{\pm 0.199}$

H.8 ZK Proof Cost and Scaling

Tables 20 to 23 report certificate costs on the workstation of Section 8, and repeated proving runs of one configuration differ by less than 5%. The Fashion-MNIST certificate holds $14{,}809$ coordinates per block in 2^{18} rows and the ResNet-18 certificates 2^{14} coordinates per block in 2^{19} rows. As N grows from 10 to 100 or T from 25 to 100, the ResNet-18 certificate keeps its size and verification time, and its prove time stays within the run-to-run variation (Table 21). Proof generation is separate from unlearning computation, and the auditor aggregates the public commitments before verifying the proofs (Table 22).

Table 20: TCGS certificate cost. Params is the number of model coordinates in the certified relation and Coords / Block the number per proof block. Prove time and peak memory are prover-side and verification time is verifier-side; times in seconds, memory in GiB, proof size in MiB.

Dataset / Model	Params	Coords / Block	Blocks $\times$ rows	Prove Time	Peak Mem.	Proof Size	Verify Time
FMNIST / CNN	44,426	14,809	3×2^{18}	113.69	21.91	0.035	0.1249
CIFAR-10 / ResNet-18	11,183,562	16,384	683×2^{19}	29,870.54	14.96	4.841	29.1872
CIFAR-100 / ResNet-18	11,229,732	16,384	686×2^{19}	31,058.31	9.83	4.857	29.4339

Measured replay baseline. To measure the $O(NTd)$ side, we implement a replay certificate on the same Halo2 stack. For each round and each slice of coordinates it proves $\theta'_{t+1,k} = \theta'_{t,k} + \sum_{j \notin \mathcal{F}} \bar{a}_j \Delta'^{t}_{j,k}$ over the retained clients' committed updates, using the fixed-point gates of the TCGS circuit and the check of equation C1 for every client. It checks aggregation without proving SGD;

Table 21: Certificate cost when only N or T varies. Forgotten-client commitments are aggregated outside the circuit, so the cost stays flat in N and T. CIFAR-10 / ResNet-18, $16{,}384$ coordinates per block; times in seconds, proof size in MiB. The two rows with $N = 20$ and $T = 100$ are independent runs of the two sweeps.

N	T	zkFU Rows	zkFU Prove Time	zkFU Proof Size	zkFU Verify Time
10	100	683×2^{19}	30,127.95	4.841	29.1740
20	100	683×2^{19}	30,857.72	4.841	29.3098
50	100	683×2^{19}	30,806.07	4.841	29.2698
100	100	683×2^{19}	30,574.85	4.841	29.2236
20	25	683×2^{19}	30,963.74	4.841	29.3950
20	50	683×2^{19}	30,683.16	4.841	29.1882
20	100	683×2^{19}	29,550.73	4.841	29.1522

Table 22: Model-computation and proof costs with and without certification. "zkFU w/o ZKP" releases θ_{out} only, and "zkFU + ZKP" also proves the TCGS certificate for θ_{tcgs}, which the prover can generate asynchronously; public aggregation precedes proof verification. Unlearning Time as in Section 8; times in seconds, proof size in MiB.

Dataset / Model	Setting	Unlearning Time	Prove Time	Verify Time	Proof Size	Auditor: proof check
FMNIST / CNN	zkFU w/o ZKP	2.08	–	–	–	–
	zkFU + ZKP	2.08	113.31	0.1263	0.035	0.1263
CIFAR-10 / ResNet-18	zkFU w/o ZKP	4.54	–	–	–	–
	zkFU + ZKP	4.54	31,008.17	29.2369	4.841	29.2369
CIFAR-100 / ResNet-18	zkFU w/o ZKP	4.58	–	–	–	–
	zkFU + ZKP	4.58	31,057.41	29.5045	4.857	29.5045

Table 23: Effect of the proof block size on the CIFAR-10 / ResNet-18 certificate. Larger blocks need fewer proofs and more memory per proof; at 17.7 rows per coordinate, 2^{12}, 2^{14}, and 2^{16} coordinates per block use 2^{17}, 2^{19}, and 2^{21} rows, and total proving time is nearly constant because the total row count is fixed by d. Times in seconds, memory in GiB.

Coords / Block	#Blocks	Prove Time	Peak Mem.	Verify Time
2^{12}	2731	30,006.48	3.75	36.4522
2^{14}	683	30,771.43	15.19	29.3068
2^{16}	171	31,163.43	29.04	27.3763

it therefore measures the aggregation portion of this direct replay construction. Proving each local step would add a proof of training, reported at about 15 minutes per iteration for a 10M-parameter VGG-11 at batch size 16 (Abbaszadeh et al., 2024). Both certificates use 2^{17}-row proofs with 17 advice columns. A TCGS proof holds $4{,}096$ coordinates, the 2^{12} configuration of Table 23, and a replay proof holds as many coordinate–client terms as fit in the same cells. A replay term needs one fixed-point product where a TCGS coordinate needs two, which predicts a prove-time ratio of $(N-1)T/2$. All $1{,}208$ proofs were generated on one 8-thread prover of the workstation in Section 8, with synthetic inputs, since the cost depends only on the circuit shape.

Table 24 and Figure 5 show that the two certificates scale in different variables. The zkFU prove time grows linearly in d, with log-log slope 0.99, and stays flat in N and T, which do not enter its circuit. The replay prove time grows linearly in $N-1$ and in T, with slopes 1.01 and 1.00, and its ratio to zkFU tracks $(N-1)T/2$ within 7% in every configuration. For Fashion-MNIST with $N = 20$, one complete replay round takes 693.6 s (104 proofs), against 74.6 s for the whole zkFU certificate. Linear projection to $T = 100$ rounds gives about $930\times$ lower proving time for zkFU.

Table 24: Measured prove cost of an aggregation-only replay certificate against the zkFU TCGS certificate on the same Halo2 stack. At $d = 4{,}096$ zkFU needs one proof, with prove time 6.63 s, verification 0.013 s, and 7.25 KiB, averaged over four runs. The last row is one complete round of the Fashion-MNIST setting and is compared with the 11-proof zkFU certificate for $d = 44{,}426$. Times in seconds, proof size in KiB. Ratios are replay over zkFU at the same d.

N	T	d	Replay proofs	Prove	Verify	Size	Prove ratio	Verify ratio	$(N-1)T/2$
5	10	4,096	20	134.6	0.174	144.4	20.3×	13.2×	20.0×
10	10	4,096	50	311.8	0.283	331.9	47.0×	21.5×	45.0×
20	10	4,096	100	648.3	0.476	694.4	97.8×	36.2×	95.0×
50	10	4,096	250	1722.4	1.116	1781.9	259.8×	84.9×	245.0×
10	5	4,096	25	154.2	0.143	165.9	23.3×	10.9×	22.5×
10	20	4,096	100	626.2	0.582	663.8	94.4×	44.3×	90.0×
10	50	4,096	250	1552.9	1.424	1659.4	234.2×	108.3×	225.0×
20	1	44,426	104	693.6	0.518	748.2	9.3×	3.5×	9.5×

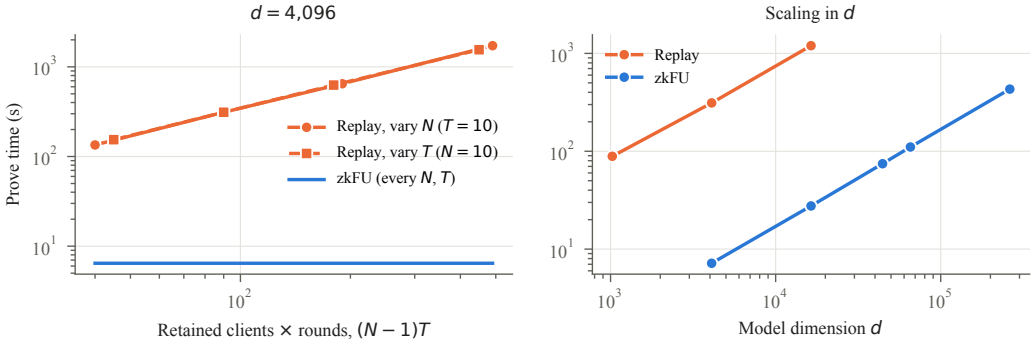

Figure 5: Measured prove time of the zkFU TCGS certificate and of an aggregation-only replay certificate on the same Halo2 stack, on log-log axes. Left: $d = 4{,}096$, varying N at $T = 10$ and T at $N = 10$; zkFU uses one circuit for every N and T. Right: varying d, with replay at $N = 10$ and $T = 10$. Numbers for the left panel are in Table 24.

I Extended Related-Work Comparison

Relation to historical subtraction. For one client i^* under uniform weights, equation 4 becomes

$$\theta_{\text{tcgs}} = \theta_0 + \frac{N}{N-1}(\theta_T - \theta_0) - \frac{1}{N-1}\sum_{t=1}^{T}\Delta_{i^*}^{t}.$$

This is the normalized historical expression in Section 4.1 of Wu et al. (2022) before its trajectory-skew term. Their implemented method instead uses unnormalized subtraction followed by knowledge distillation on external unlabeled data. zkFU develops the relation for weighted client sets and linear server rules, binds it to committed updates, analyzes trajectory mismatch, and introduces retained-aware sparse recovery. The Direct Accumulation baseline isolates subtraction without recovery; it is not a reproduction of that complete distillation method.

Table 25 compares zkFU with the verification-oriented systems of Section 2. The verified object separates them most sharply. Behavioral tests can be passed by an adaptive server without deleting anything (Zhang et al., 2024), aggregation proofs say nothing about a later deletion, and centralized and pruning proofs certify one trainer's computation, whereas zkFU certifies a trace-defined deletion relation whose proof target is $O(d)$ after public aggregation.

Table 25: Verification-oriented systems for unlearning and federated learning, compared by verified object, threat model, proof cost, and setting. N is the number of clients, T the number of rounds, and d the model dimension.

System	Verified object	Threat model	Proof cost	Setting
VeriFi / uS2U (Gao et al., 2024)	Behavioral evidence of forgetting	Server assumed to run the test honestly	none (no proof)	federated
zkFL (Wang et al., 2024)	Per-round aggregation correctness	Malicious aggregator during training	proof of every round's aggregation, grows with N, T, and d	federated
Verifiable unlearning (Eisenhofer et al., 2025)	Training/unlearning state and data non-membership	Untrusted centralized trainer	retraining, amnesiac, or optimization-based update proofs	centralized
ZK-APEX (Maheri et al., 2025)	Curvature-aware pruning for personalized edge models	Untrusted edge server	proof of the pruning procedure per model	personalized edge
zkFU (this paper)	Trace-defined deletion seed, exact subtraction of committed contribution	Untrusted reported endpoints/seeds; valid training trajectory for the quality bound	$O(d)$ after block-wise public aggregation	federated

J Discussion

Storage cost. TCGS requires openings for committed updates to be retained until deletion is no longer required, at the per-round or running-aggregate cost given in Appendix B. This is practical for cross-silo FL with a small number of high-value clients. Cross-device FL may require compressed sketches, periodic checkpoints, or selective trace retention, which trade storage for weaker or coarser certificates.

Beyond FedAvg. TCGS relies on a server rule that is linear in the submitted updates with public coefficients, which covers FedAvg, FedProx, FedAvgM, and FedNova (Appendix H.1), while non-linear robust aggregators need a different relation. Extending zkFU to SCAFFOLD (Karimireddy et al., 2020) or adaptive aggregation requires deriving the corresponding trace relation under the modified update rule. The proof architecture remains the same if the resulting deletion relation is affine or efficiently arithmetizable. Historical-update methods such as FedEraser and FedRecovery and shard-retraining architectures such as SISA (Bourtoule et al., 2021) would need method-specific relations for their reconstruction, recalibration, or retraining steps.

Proof backend. Homomorphic aggregation defines the certification relation independently of its proof backend. In the exact affine setting where the public seed determines $W_{\mathcal{F}}$, each block can publicly form

$$Y_b = c^{\mathsf{agg}}_{\mathcal{F},b} - \sum_{j \in b} W_{\mathcal{F},j} \mathsf{G}_j$$

and use a Schnorr proof of knowledge of r_b such that $Y_b = r_b \mathsf{H}$ (Hao, 2017); the seed commitment admits the analogous check. The challenge must bind the complete request, trace, encoding, and block context. This alternative retains public vector-processing costs. The reported timings characterize the Halo2 prototype and the implemented replay baseline; they do not establish superiority over this specialized opening proof. Comparing backends requires the same encoded relation and public-input checks.

Certification and recovery. zkFU concentrates cryptographic verification on the affine TCGS relation, giving an $O(d)$ proof target after public aggregation. TraceTrust supplies retained-aware recovery, whose forgetting–utility behavior is evaluated in Section 8. Extending certification to recovery would require constraints for retained-statistic aggregation, mask selection, calibration, and repair updates, together with an output commitment. Binding online predictions to the resulting model additionally calls for verifiable inference or runtime attestation.