DELETION MUST REACH EVERY COPY: FEDERATED EXECUTION OF UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated unlearning (FU) removes the influence of withdrawn data from a model trained by clients that never centralize it. FU faces a distinctive obstacle: federated training leaves persistent copies on devices and at the edge, and each copy an edit misses returns the withdrawn content through later aggregation. Current methods edit the server's global model, and they certify that model rather than the copies a request governs. To address these, we first establish unlearning as an operation on the federated state: the server transcript does not determine which parties a request affects, and holder-local readings do. We then design **LOCUS**, a Locality-Ordered Correction of Unlearned State framework that executes each step where its evidence lives. Within this framework, Influence Localization has each holder report the retention cost of a candidate correction and what its copy still carries. Correction Propagation then delivers one minimum-norm correction to every holder in that closure and gates a copy out of aggregation until it acknowledges. In addition, Divided Verification assigns removal, retention and application checks to the parties that observe them and binds them into one certificate. Across two benchmarks, three deployments, five unlearning objectives and seven published FU protocols, **LOCUS** corrects 92–99% of the governed copies, while server-side execution corrects none and the protocols at most 60%. For example, on TOFU with Llama-3.2-1B, **LOCUS** with NPO corrects 95% of the copies and preserves the erasure throughout a 300-round continuation, whereas server-side NPO loses it after 17 rounds and Oblivionis, the longest-surviving published protocol, after 73 rounds.

1 INTRODUCTION

A person who withdraws consent asks that a system stop carrying their data, and the obligation runs over every copy the system made (Ginart et al., 2019). For a language model the operational answer is machine unlearning: a procedure edits the trained weights so the designated content is no longer produced, without retraining from scratch (Cao & Yang, 2015; Bourtoule et al., 2021). Federated deployments give that obligation an unusual shape, because training distributes the model by construction. Every round broadcasts weights to the participants, personalized adapters stay resident on the devices that fitted them, and released checkpoints answer user requests at the edge (McMahan et al., 2017; Zhang et al., 2024a). The request arrives at the server; the copies are everywhere else. We review related work in Appendix A.

As illustrated in Figure 1, federated unlearning answers the request on the server. Published procedures roll back the stored contribution of a departing client, ascend the forget loss on the global model, distil into a fresh model, or retrain a suffix of rounds (Liu et al., 2021; Zhang et al., 2026; Zhu et al., 2023; Liu et al., 2022), and the guarantee each publishes is a statement about the model the server holds when the edit finishes (Che et al., 2023; Tao et al., 2024). The request addresses the state the training loop maintains, which holds the replicas on client devices, the local datasets that decide what has to be retained, and the aggregation rule that runs again next round; and for each of those parts the evidence needed to act on it sits on the device that holds it. This raises the first question: ***I) Which parts of the federated state does a deletion request govern, which of them can a server determine from its own transcript, and which can only the client devices that hold them measure locally?***

The copies the edit misses do not sit still. An uncorrected replica is the starting point of its holder's next local step, so its upload can return the withdrawn contribution to the global model through

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.

aggregation. The effect is documented from the adversarial side, where a client controlling its uploads steers post-deletion rounds to restore erased knowledge (Wang et al., 2025), and asynchronous pipelines address stale contributions at the server (Su & Li, 2023). A deletion guarantee also needs the ordinary-training rate: how long an erasure survives when an outstanding copy creates positive net reflow, and what it costs to block that copy's contribution. This raises the second question: ***II) How does a correction reach every copy in the state, including copies whose holders are offline when it is issued, and how long does an erasure survive aggregation when a copy is left behind?***

A statement about the state has to be checkable, and no single party observes the state. The party that withdrew the data holds the content whose removal is at issue, and verification protocols have it probe the released model for that content (Gao et al., 2024). The loss that says whether a remaining client kept what it was entitled to keep is defined on that client's data, which no other party reads, so the server that cannot measure retention cannot attest to it; recent systems ask remaining clients to confirm their own contributions survived (Liu et al., 2025). Whether a device applied the correction is observable at that device, and a device that is offline during the edit produces no transcript, so the server holds the same view whether or not the copy was corrected. This raises the third question: ***III) Which party's observations ground which part of the claim, and how do these separate checks compose into one certificate?***

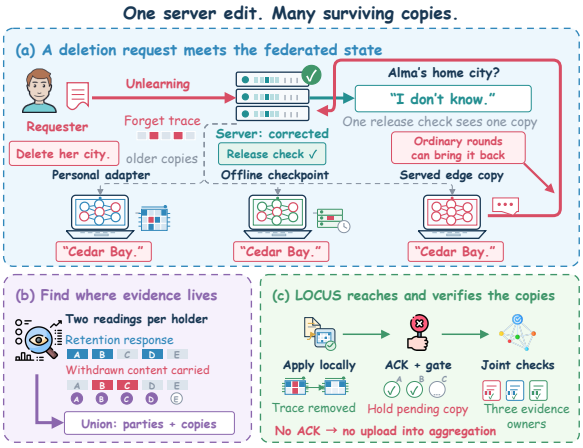


Figure 1: **Problem Illustration.** One server-side edit passes the release check while adapters, offline checkpoints and edge copies keep answering with the withdrawn content. **I)** Undetermined Localization: the server's transcript does not fix the parties and copies a request governs, and two device-side readings do. **II)** Aggregation Reflow: a copy the correction missed returns that content through ordinary rounds. **III)** Divided Evidence: removal, retention and acknowledgment sit with three different parties.

We address these challenges with **LOCUS**, a **L**ocality-**O**rdered **C**orrection of **U**nlearned **S**tate. Its three stages—**Influence Localization (IL)**, **Correction Propagation (CP)** and **Divided Verification (DV)**—place each operation with the party holding its evidence. A request therefore names the copies to correct, the retention budgets to respect and the acknowledgments needed before those copies rejoin aggregation. Our main contributions are:

❶ ***State-Level Separation.*** We identify and formalize a state-level separation for federated unlearning: a corrected server checkpoint can coexist with persistent copies that retain the withdrawn contribution, while the server's transcript leaves their carrying status and collateral retention cost undetermined. Paired constructions establish these information boundaries, and two holder-local readouts supply the measurements needed to form a correction closure. The separation extends to continued training: uniformly positive net aggregation reflow gives a threshold-crossing horizon, whereas an acknowledgment gate blocks outstanding copies and supports persistence under non-restoring admitted updates with controlled curvature expenditure. These results characterize where a deletion must act and what evidence supports completion.

❷ ***Local Correction Protocol.*** We introduce **LOCUS**, realizing state-level deletion through three coordinated mechanisms: IL measures each holder's retention sensitivity and residual carrying margin, forms the affected and carrying sets, and refines their closure through a complete nonlinear audit; CP sends the accepted correction to every governed copy, records versioned acknowledgments and gates the uploads of outstanding holders, keeping communication costs tied to the copies that must change; and DV assigns removal to the requester, retention to the affected clients and propagation to authenticated holder reports, binding these readings to one versioned request and a committed release. We back the protocol with localization results, a reflow horizon and a composite verification rule under an explicit model of protocol-following parties.

❸ ***Empirical Verification.*** Experiments on TOFU with Llama-3.2-1B and Qwen2.5-7B and on MUSE-News with Llama-2-7B, with 20 clients holding persistent adapters, show that **LOCUS** corrects

92–99% of the governed copies under five unlearning objectives, while server-side execution corrects none and seven published federated unlearning protocols at most 60%, with coverage, copy margin, survival over a 300-round continuation, retention excess and utility characterizing the corrected state. Stage ablations show that only the full configuration keeps the erasure through the continuation on all three deployments, and independent audits recheck every retained client and persistent copy after correction, keeping failed corrections and censored survival times on record.

2 PRELIMINARIES

Federated learning. We consider a federation of one server and a set $\mathcal{C}$ of clients, where client c holds a private dataset $\mathcal{D}_c$ of n_c samples that never leave it, and n denotes the total sample count across the federation. Training proceeds in rounds. In round t the server broadcasts the global model $w[t]$ to the participating clients, client c computes an update $\Delta_c[t]$ on its received copy from $\mathcal{D}_c$, and the server aggregates the uploads,

$$w[t+1] \;=\; w[t] + a[t] \;=\; w[t] \;-\; \alpha \sum_{c\in\mathcal{C}} \frac{n_c}{n} \sum_{k=0}^{E-1} \nabla \mathcal{L}_c\big(w_c^k[t]\big), \tag{1}$$

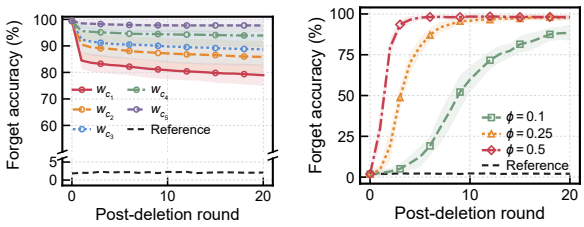
where $\mathcal{L}_c$ is the loss client c evaluates on $\mathcal{D}_c$, α the local learning rate, E the number of local steps, $w_c^k[t]$ the k-th local iterate starting from the broadcast $w[t]$, $a[t]$ the aggregate step of the round, and d the dimension of the trainable parameters, so the update $\Delta_c[t]$ is the sum of client c's local gradient steps. The server records what client c sent in a history H_c. Two facts follow. Every quantity the server observes is an uploaded update or a statistic of one, so the data, the losses it induces, and the evidence of what a model retains on it stay on the clients. And every broadcast leaves a copy behind.

Persistent copies. We study deployments in which a broadcast copy outlives the round that produced it, which is where a deletion request has copies to reach. Three arrangements produce such copies. In personalized federated fine-tuning, and in the federated low-rank adaptation (Hu et al., 2021) of language models we take as the working instance, client c keeps a local adapter that it fits on $\mathcal{D}_c$ and that is never replaced by a broadcast (Guo et al., 2025; Yang et al., 2024). In partial-participation and asynchronous training, a client that sits out rounds returns holding the model it last received and computes its next update on it (Nguyen et al., 2022). A released checkpoint handed to a deployment answers user requests for as long as it is served. We write w_c for party c's copy: a model the federation produced, stored outside the server, and where that party's next contribution starts.

Federated state. The object the training loop maintains is therefore larger than the global model. Σ denotes the tuple $\big(w,\ \{w_c\}_{c\in\mathcal{C}},\ \{\mathcal{D}_c\},\ \{H_c\},\ \mathrm{Agg}\big)$: the current global model, the copies, the local datasets, the update histories, and the aggregation rule of Eq. (1). The server holds the global model, the histories and the rule and reads all three in full; the holders keep the copies and the server sees only what they report of them; the datasets stay with the clients and the server never sees them. Training writes to all of Σ, since each round moves the global model, refreshes the copies of the clients it reaches, and appends to every history.

Unlearning problem. A deletion request arrives when a client q withdraws consent for a designated part of its data, from single records to its whole dataset, and the federation must produce a state in which the influence of the withdrawn forget set $\mathcal{D}_f$ is removed while performance on the remaining data is preserved. Write $\mathcal{L}_f$ for the loss on the forget set. For each copy class c, let $\widetilde{w}_c$ be its matched reference produced by the same federation without $\mathcal{D}_f$: the served copy uses the retrained global reference, and a persistent personalized copy uses its matched retrained personalized reference. The deployer fixes $\gamma_c = \mathcal{L}_f(\widetilde{w}_c)$ and uses $\mathcal{L}_c(\widetilde{w}_c)$ as the retention baseline. A copy answers as forgotten when its forget loss stands above its own γ_c; its collateral retention cost is the excess over the matched reference, which separates collateral damage from the intended removal of withdrawn influence. An unindexed γ below denotes the served-model threshold. A single global threshold or a pre-request retention baseline is invalid when matched copy classes have different post-deletion loss distributions. Current practice executes the request on one component of Σ: the server edits its copy of the global model, by rolling back stored updates, ascending $\mathcal{L}_f$, distilling into a fresh model, or partially retraining (Liu et al., 2021; Zhang et al., 2026; Zhu et al., 2023; Liu et al., 2022), and the edited model is the object the accompanying guarantee speaks about. We call this execution mode server-side editing. The request, however, is addressed to Σ. The evidence that separates what must

be removed from what must be retained sits in the local datasets, the withdrawn influence survives in every copy that predates the edit, and Eq. (1) runs again on updates computed from those copies.

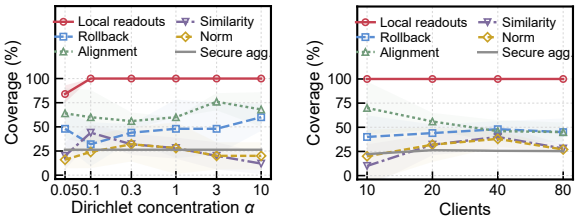


(a) Copy retention. (b) Aggregation reflow.

Figure 2: Forget accuracy of uncorrected copies and of the global model across post-deletion rounds.

Party model. All parties follow the protocol, and each observes only its own transcript. The server reads the uploads and the histories it stores; client c reads the broadcasts it receives, its copy w_c, and its own data. No party reads another's data or device, and a party that is offline emits nothing, so the server's transcript is the same whether an absent client applied a correction or did not. Each party can therefore ground only the claims its own observations support: the requester q holds $\mathcal{D}_f$ and can query a model with the withdrawn content itself, client c is the only party that evaluates $\mathcal{L}_c$, and the holder of w_c is the only party that reads the state of that copy. What follows the deletion request is ordinary continued training, whose rounds are the mechanism by which an uncorrected copy re-enters the global model; a client that chooses its local data steers those rounds against the erasure, which makes the benign rate a floor on what an adversarial one achieves.

Desiderata. We require of an unlearning execution the properties the training loop already grants to learning. (i) Localization is joint: the parties whose retention the correction moves, and the copies that still answer with the withdrawn content, are identified with the participation of the parties that hold them. (ii) Updates are complementary: the removal computed against $\mathcal{D}_f$ arrives paired with retention constraints that the remaining clients compute from their local data, each carrying a stated bound. (iii) Correction reaches every copy: the edit propagates to each copy the request governs, and a copy that has not applied it does not feed later aggregation. (iv) Verification follows the information: removal, retention and propagation are each checked by the party whose observations can support the check, and the three readings compose into one statement. An execution that satisfies none of the four is a write to w alone, which is the guarantee server-side editing offers, and each further property extends the statement to one more component of Σ.



(a) Heterogeneity sweep. (b) Federation size.

Figure 3: Affected-set coverage of server-side attribution and of local readouts across two federation axes.

Observation 1

A copy that outlives the round which produced it keeps answering with the withdrawn content after the server's edit, and ordinary aggregation carries that content back into the global model.

We read the two panels of Figure 2 on a Dirichlet-partitioned (Hsu et al., 2019) federation of CIFAR-10 image-classification clients (Krizhevsky, 2009) in which a share of the participants keep a persistent local copy across rounds. The server first replaces w with the global model retrained without the requester's data, and training then continues under Eq. (1) with no correction delivered to the copies. Panel a tracks the share of $\mathcal{D}_f$ that each retained copy w_c still answers with the withdrawn content, which measures what a copy still carries; panel b tracks the same share on the global model over the same rounds, one curve per fraction ϕ of copy-keeping clients, which measures what aggregation returns. The level of the retrained reference is drawn on both panels, so a curve above it marks a model that answers content the federation has undertaken to stop carrying.

Observation 2

Federations whose uploads, histories and aggregates agree every round can differ in which parties a request affects, so the server's transcript does not determine the set to correct.

Figure 3 reads the same federation against a request whose affected set is fixed by construction, so that coverage is defined without estimating it. Each panel plots the share of that set a rule recovers,

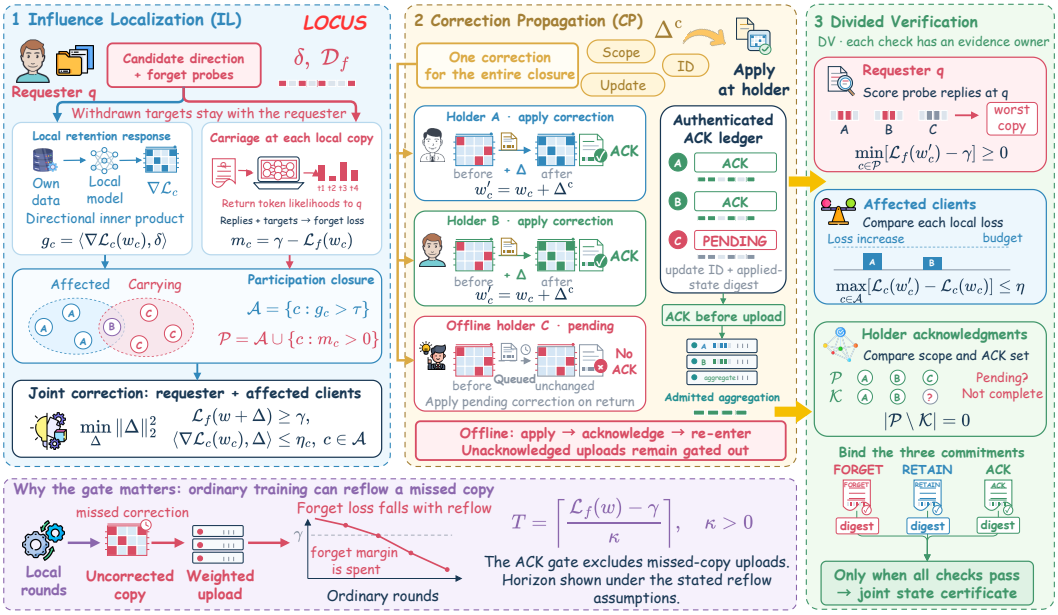


Figure 4: **Overview of LOCUS.** IL has each holder return a directional retention response and a carriage reading on its copy, whose union fixes the participation closure a request governs. CP solves one minimum-norm correction for that closure, applies it at every holder, and gates a copy out of aggregation until its acknowledgment arrives. DV assigns the forget, retain and acknowledgment checks to the parties observing them, binding them into one state certificate.

for four attribution rules a server can run on the uploads and histories it stores, for the chance level a server is left with under secure aggregation, and for the retention response each party measures on its own device. Panel a sweeps the Dirichlet concentration that governs how unevenly the data is split while holding the client count and the per-client sample count fixed, and panel b sweeps the client count at a fixed concentration, so that heterogeneity and federation size are read apart.

3 Methodology

3.1 Framework Overview

LOCUS executes deletion where its evidence lives (Figure 4). **I)** IL collects holder-local retention and carriage readings to determine the participation closure. **II)** CP applies the correction across that closure and gates outstanding uploads. **III)** DV binds removal, retention and application evidence into one state certificate. Algorithm 1 gives the protocol and Appendix C the derivations.

3.2 Influence Localization (IL)

Motivation. A correction concerns both clients whose retention it changes and copies that still carry the withdrawn content. IL obtains the two readings from the holders that can evaluate them.

The readouts. The requester q forms a candidate correction direction δ from $\mathcal{D}_f$ and releases it with a probe set drawn from the same data. Party c returns two scalars, the first computed from its own data and the second from its own copy:

$$g_c = \langle \nabla \mathcal{L}_c(w_c),\, \delta \rangle, \qquad m_c = \gamma_c - \mathcal{L}_f(w_c), \tag{2}$$

where δ is the candidate correction direction and γ_c is the forget loss of the matched leave-requester-out retraining reference for copy class c (the retrained global reference for the served model and any requester copy with no retained-client counterpart). The directional response g_c measures the local loss change along δ; positive m_c indicates a forget-loss shortfall. Holder c returns probe likelihoods, while the requester retains the withdrawn targets and assembles $\mathcal{L}_f(w_c)$.

The closure. The initial affected set contains responses above τ, and the carrying set contains positive carriage readings; their union defines the initial closure. Appendix C.4 constructs identical server transcripts with different directional affected sets, so the closure is read from the holders. That reading is exact under bounded reporting error: if each reported response lies within ε_g of g_c and each carriage reading within ε_m of m_c, thresholding returns both sets whenever $|g_c - \tau| > \varepsilon_g$ and $|m_c| > \varepsilon_m$ for every holder (Proposition 10). Each affected client supplies a retention allowance η_c. The shared correction targets every governed copy subject to the matched retention constraints,

$$\Delta^{\mathrm{c}} \;=\; \underset{\substack{\Delta\in\mathbb{R}^d:\ \mathcal{L}_f(w_h+\Delta)\ge\gamma_h,\ h\in\mathcal{P}\cup\{\mathrm{server}\}\\ \mathcal{L}_c(w_c+\Delta)-\mathcal{L}_c(\widetilde{w}_c)\le\eta_c,\ c\in\mathcal{A}}}{\arg\min} \big\|\Delta\big\|_2^2, \tag{3}$$

where Δ^{c} is the shared correction jointly determined from holder-local measurements. The retention term measures damage beyond the outcome that exact leave-requester-out retraining itself requires; anchoring it at w_c would simultaneously demand matched-reference removal and preservation of influence that the same reference removes. During the solve, removal is measured on every persistent copy and the served model, and retention is measured for every non-requesting client; after convergence, $\mathcal{A}$ is the set whose measured nonlinear loss increase over its matched baseline under the accepted correction exceeds the declared response tolerance, while $\mathcal{P}$ remains its union with the carrying set. Measuring all clients is necessary because the accepted correction need not be collinear with the candidate direction δ: the signs of the initial g_c readings alone do not determine the final affected set. A local sequential solver starts from δ, refreshes losses and gradients at trial displacements, and refines a feasible proposal toward the local minimum-norm boundary. Release requires an accepted correction that passes every audited constraint, and the audit record keeps the trials behind it. Holders participate in this solve before the scheduled disconnection event, since an offline holder's measurement is one only that holder can supply.

3.3 Correction Propagation (CP)

Motivation. An uncorrected copy can reintroduce withdrawn content through its next ordinary training upload. CP therefore binds delivery to the copy and admission to its current acknowledgment.

The survival horizon. The margin a round spends and the number of rounds the erasure survives are

$$\kappa \;=\; \sum_{c\in\mathcal{C}} \frac{n_c}{n}\left\langle -\nabla\mathcal{L}_f(w),\ \Delta_c\right\rangle \;-\; \frac{M}{2}\,\|a\|_2^2, \qquad T \;=\; \left\lfloor \frac{\mathcal{L}_f(w)-\gamma}{\kappa}\right\rfloor + 1, \tag{4}$$

where $\kappa > 0$ is a lower bound on the net descent rate throughout the audited rounds and M bounds the smoothness of $\mathcal{L}_f$. The full sum includes both outstanding holders $\mathcal{U}$ and already corrected clients: a non-restoring corrected update can offset reflow and must remain in the rate. If that rate stays positive, the served-model loss falls strictly below γ by round T. The floor-plus-one convention resolves equality at the threshold. An outstanding copy can create such a rate through its aggregation weight, but a nonempty outstanding set alone does not imply positive net descent. Appendix C.2 gives the bound and an affine instance attaining it.

Acknowledgment and re-entry. The correction is delivered to every member of $\mathcal{P}$, each holder applies it to its own copy and returns an authenticated acknowledgment of the resulting state, and the server admits an upload into Eq. (1) only from holders whose acknowledgment for the current correction has arrived,

$$w[t+1] = w[t] + \sum_{c\in\mathcal{E}[t]} \frac{n_c}{n}\Delta_c[t], \qquad \mathcal{E}[t] = \mathcal{C}\setminus\big(\mathcal{P}\setminus\mathcal{K}[t]\big), \tag{5}$$

where $\mathcal{K}[t]$ contains acknowledgments for the current correction and $\mathcal{E}[t]$ is the eligible client set. A governed holder applies Δ^{c} once to its source copy before acknowledgment; later local steps start from its current corrected state. Holders outside the closure remain eligible. A returning holder processes pending corrections before its first admitted upload. The gate removes outstanding copies from the aggregate, while their ledger entries remain pending until they acknowledge. Under the additional non-restoring condition on admitted corrected updates, the margin lost over k rounds is at most $(M/2)\sum_{j<k}\|a[t+j]\|^2$. The erasure persists for every prefix whose expenditure stays within the release margin, which is the condition the admitted rounds are held to. The audit logs the forget-loss margin and the held aggregation mass at every round (Appendix D.4).

Cost. Propagation costs one broadcast of the correction and one acknowledgment per member of the closure, which is one round of communication in the units the training loop already pays, and the correction itself is a low-rank object in the adapter setting we instantiate. The gate holds out the contributions of members that have not yet acknowledged, so the utility it defers is bounded by the aggregation mass of that set over the rounds it takes them to return,

$$\|a[t] - a^{\mathrm{g}}[t]\|_2 = \left\| \sum_{c \in \mathcal{U}[t]} \frac{n_c}{n} \Delta_c[t] \right\|_2 \le B\mu[t], \qquad \sum_{j<t} \|a[j] - a^{\mathrm{g}}[j]\|_2 \le B \sum_{j<t} \mu[j], \tag{6}$$

where a^{g} filters the same proposed uploads as a, B bounds each upload norm, and $\mu[t]$ is the held aggregation mass. This is the immediate update discrepancy injected by the gate; independently evolving training trajectories require a separate stability bound. Future injected discrepancy vanishes after the last acknowledgment. Correction trials add the exchanges counted in Appendix B.5.

3.4 Divided Verification (DV)

Motivation. Removal, retention and application are observed by different parties. DV assigns each predicate to its evidence source and binds the readings to the same corrected state.

The certificate. Write $\mathcal{K}$ for the set of holders whose acknowledgment has been received. The certificate asserts three statements, one per party class:

$$\min_{c \in \mathcal{P} \cup \{\mathrm{server}\}} \left[\mathcal{L}_f(w_c') - \gamma_c \right] \ge 0, \qquad \max_{c \in \mathcal{A}} \left[\mathcal{L}_c(w_c') - \mathcal{L}_c(\widetilde{w}_c) \right] \le \eta, \qquad \left| \mathcal{P} \setminus \mathcal{K} \right| = 0, \tag{7}$$

where w_c' is the corrected copy of Eq. (5) and η the retention budget the deployer declares, so the first statement reads the forget loss at every corrected copy against the threshold, the second reads each affected client's excess loss over its matched leave-requester-out reference, and the third reads the acknowledgment ledger. The requester scores removal from probe responses, clients report retention, and the server reads the acknowledgment ledger. Version-bound commitments tie these readings to the same request and corrected copies: each acknowledgment carries the request version, the source-copy and post-correction digests and the holder identity, so it closes only the request version it names; retained clients declare their budgets before a correction is chosen, the requester commits to its forget and audit splits, and the coordinator to the candidate correction and solver configuration.

Necessity of the acknowledgment. An offline holder can apply a correction without changing the server transcript. Appendix C.5 pairs that run with an unapplied copy: a server-only rule cannot decide application correctly on both. Holder-dependent evidence is necessary; an authenticated acknowledgment supplies it under honest reporting, while a forged state assertion requires a separate trusted copy-state check.

Degeneracy. If no holder copy persists and the accepted correction's nonlinear retention audit leaves no affected client, only the served-model removal condition remains. Otherwise, the closure identifies where additional evidence and propagation are needed. The protocol adds localization, correction-trial exchanges, delivery and acknowledgments; Appendix C.6 separates this accounting from the information needed to resolve an unobserved holder state.

4 Experiments

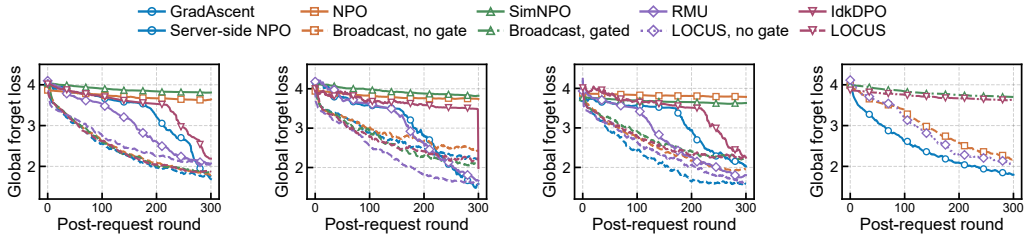
4.1 Experimental Setup

Deployments. We compare TOFU on Llama-3.2-1B-Instruct and Qwen2.5-7B-Instruct, and MUSE-News on Llama-2-7B (Maini et al., 2024; Shi et al., 2025; Grattafiori et al., 2024; Yang et al., 2025; Touvron et al., 2023). Each deployment has 20 clients with persistent rank-16 adapters. TOFU assigns ten fictitious authors to each client; MUSE partitions articles by topic. The default request removes the benchmark forget split; four other holders receive related paraphrases, and an independent all-client audit fixes the affected and carrying sets. After the correction is formed, 20% of holders remain offline for ten rounds; continuation lasts 300 rounds. The 1B setting uses three training seeds and each 7B setting one run. Appendix B.2 gives partitions and matched references.

Comparisons. GradAscent, NPO, SimNPO, RMU and IdkDPO each supply a correction under server-only and LOCUS execution (Jang et al., 2023; Zhang et al., 2024b; Fan et al., 2025; Li et al.,

Table 1: Objectives and execution controls, three deployments. Cov: corrected closure share, which server-side execution holds at zero by construction; M_c: minimum copy margin; T: survival rounds; RL: largest retention excess; U: benchmark utility. Parentheses give SD over three 1B runs; 7B cells use one. ≥ 300 is censored. Rose/blue/amber mark best/second/worst; retrain rows excluded.

	TOFU / Llama-1B					TOFU / Qwen-7B					MUSE / Llama-7B				
Objective / execution	**Cov↑**	M_c ↑	**T↑**	**RL↓**	**U↑**	**Cov↑**	M_c ↑	**T↑**	**RL↓**	**U↑**	**Cov↑**	M_c ↑	**T↑**	**RL↓**	**U↑**
(a) Server-side execution															
GradAscent (Maini et al., 2024)	0.00	-0.74(±0.01)	14(±2)	0.021(±0.002)	0.37(±0.003)	0.00	-0.67	16	0.039	0.47	0.00	-0.59	12	0.052	0.35
NPO (Zhang et al., 2024b)	0.00	-0.65(±0.01)	17(±1)	0.011(±0.0004)	0.40(±0.003)	0.00	-0.64	13	**0.014**	0.63	0.00	-0.59	18	0.020	0.49
SimNPO (Fan et al., 2025)	0.00	-0.62(±0.02)	19(±3)	**0.009**(±0.001)	0.44(±0.002)	0.00	-0.63	21	0.016	0.50	0.00	-0.58	15	0.020	0.47
RMU (Li et al., 2024)	0.00	-0.69(±0.01)	7(±2)	**0.006**(±0.001)	**0.63**(±0.01)	0.00	-0.71	9	**0.012**	0.62	0.00	-0.76	4	**0.013**	0.53
IdkDPO (Maini et al., 2024)	0.00	-0.63(±0.05)	16(±1)	0.013(±0.001)	0.35(±0.001)	0.00	-0.76	14	0.019	0.62	0.00	-0.63	14	0.031	0.46
(b) LOCUS execution															
LOCUS / GradAscent	0.92(±0.01)	0.09(±0.01)	210(±45)	0.020(±0.002)	0.51(±0.04)	0.93	0.10	156	0.040	0.48	**0.99**	0.05	172	0.045	0.35
LOCUS / NPO	0.95(±0.01)	**0.20**(±0.01)	$\geq$ 300	0.010(±0.001)	**0.54**(±0.01)	0.98	**0.24**	$\geq$ 300	0.015	0.67	0.94	**0.21**	$\geq$ 300	**0.015**	0.51
LOCUS / SimNPO	**0.99**(±0.003)	**0.19**(±0.01)	$\geq$ 300	**0.009**(±0.002)	0.53(±0.01)	0.97	0.21	$\geq$ 300	**0.014**	**0.70**	0.98	0.14	$\geq$ 300	0.018	**0.60**
LOCUS / RMU	0.94(±0.02)	0.05(±0.002)	118(±32)	0.023(±0.001)	0.51(±0.03)	0.97	0.08	144	0.016	**0.76**	0.96	0.06	100	**0.015**	**0.54**
LOCUS / IdkDPO	0.96(±0.01)	0.13(±0.01)	**244**(±31)	0.012(±0.001)	0.50(±0.02)	**0.99**	0.15	**299**	0.023	0.60	0.98	0.12	**219**	0.027	0.46
(c) Broadcast controls															
Broadcast-all / NPO, gated re-entry	**1.00**	**0.19**(±0.01)	$\geq$ 300	0.025(±0.001)	0.52(±0.01)	**1.00**	**0.22**	$\geq$ 300	0.038	0.62	**1.00**	**0.18**	$\geq$ 300	0.051	0.45
Broadcast-all / NPO, no gate	**1.00**	-0.56(±0.03)	86(±13)	0.029(±0.003)	0.47(±0.04)	**1.00**	-0.58	134	0.041	0.63	**1.00**	-0.66	76	0.057	0.48
(d) Federated protocols															
FedEraser [IWQoS'21] (Liu et al., 2021)	0.29(±0.03)	-0.46(±0.02)	32(±2)	0.037(±0.003)	0.48(±0.03)	0.33	-0.44	35	0.064	0.54	0.21	-0.42	26	0.069	0.40
FedRecover [S&P'23] (Cao et al., 2023)	0.33(±0.02)	-0.43(±0.01)	40(±7)	0.027(±0.002)	0.48(±0.01)	0.39	-0.51	31	0.047	0.57	0.31	-0.44	34	0.056	0.47
Exact-Fun [ICDM'23] (Xiong et al., 2023)	0.27(±0.01)	-0.47(±0.03)	33(±3)	0.018(±0.001)	0.46(±0.02)	0.25	-0.49	28	0.033	0.57	0.24	-0.49	19	0.051	0.42
FedOSD [AAAI'25] (Pan et al., 2025)	0.33(±0.04)	-0.41(±0.04)	51(±10)	0.015(±0.002)	0.49(±0.01)	0.41	-0.43	53	0.028	0.66	0.42	-0.44	31	0.032	0.50
VeriFi [TDSC'24] (Gao et al., 2024)	0.41(±0.01)	-0.34(±0.02)	46(±5)	0.016(±0.001)	0.51(±0.03)	0.40	-0.34	65	0.035	0.52	0.41	-0.38	47	0.042	0.40
BlockFUL [TIFS'25] (Liu et al., 2025)	0.37(±0.02)	-0.27(±0.02)	48(±3)	0.017(±0.002)	0.53(±0.03)	0.42	-0.31	37	0.033	0.57	0.41	-0.36	39	0.035	0.45
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.57(±0.02)	-0.22(±0.01)	73(±10)	0.014(±0.001)	0.52(±0.01)	0.56	-0.20	63	0.019	0.59	0.60	-0.23	63	0.029	0.43
Retrain w, old copies	0.00	-0.68(±0.06)	15(±2)	0.000	0.55(±0.03)	0.00	-0.71	14	0.000	0.63	0.00	-0.69	10	0.000	0.54
Retrain + all copies	1.00	0.37(±0.02)	$\geq$ 300	0.000	0.56(±0.02)	1.00	0.37	$\geq$ 300	0.000	0.74	1.00	0.28	$\geq$ 300	0.000	0.43



(a) TOFU / Llama-1B. (b) TOFU / Qwen-7B. (c) MUSE / Llama-7B. (d) NPO executions.

Figure 5: Continued training after deletion. (a–c) Each objective is paired under LOCUS (solid) and server-only execution (dashed). (d) NPO separates broadcast, localization and gated re-entry. Lines average available runs; survival entries summarize their first crossings.

2024; Rafailov et al., 2023). Broadcast controls deliver NPO to every holder with or without an acknowledgment gate, seven federated protocols supply further execution comparisons, and retraining either replaces only the global model or refits every governed copy. Pairing the objective, request and participation schedule isolates the effect of where the correction is applied.

Metrics. Table 1 pairs closure coverage, the share of governed copies that applied the correction, with the minimum copy margin, survival, retention excess and utility; affected-set and carrying-set F1 are separate localization metrics. Survival ends at the first global threshold crossing, and noncrossing trajectories are censored at 300 rounds. Appendix B.4 defines the aggregation units and resource accounting.

4.2 Correction and Continued Training

Figure 5 separates an objective's release behavior from the effect of correcting persistent copies, and the NPO controls separate delivery from admission: a delivery log records a message, whereas the gate decides whether an outstanding copy contributes its next update. Table 1 reads that persistence next to retention and utility, so a long survival time is judged with the quality of the corrected state. A server-side edit reaches the served checkpoint alone, so band (a) holds zero coverage.

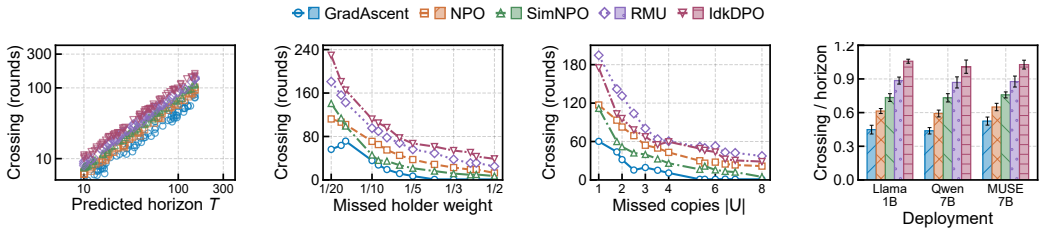
Table 2: Stage ablation under NPO. IL, CP and DV enable localization, propagation and verification. C: closure recovery; O: outstanding copies, which the all-disabled row leaves undefined because it addresses no copy. Deployment metrics and colors follow Table 1.

					TOFU / Llama-1B					TOFU / Qwen-7B					MUSE / Llama-7B				
IL	CP	DV	C↑	O↓	Cov↑	M_c ↑	T↑	RL↓	U↑	Cov↑	M_c ↑	T↑	RL↓	U↑	Cov↑	M_c ↑	T↑	RL↓	U↑
✓	✓	✓	**0.98**	**0**	**0.95**(±0.01)	**0.20**(±0.01)	≥ 300	**0.010**(±0.001)	0.54(±0.01)	**0.98**	**0.24**	≥ 300	**0.015**	**0.67**	**0.94**	**0.21**	≥ 300	**0.015**	**0.51**
✗	✓	✓	0.43	**0**	0.39	-0.30	63	0.022	0.53	0.41	-0.32	67	0.019	0.63	0.43	-0.35	48	0.022	0.47
✓	✗	✓	**0.98**	**4**	0.91	-0.10	32	0.017	0.57	0.93	-0.16	34	0.018	**0.66**	**0.94**	-0.13	34	**0.017**	0.49
✓	✓	✗	**0.99**	**0**	**0.97**	**0.16**	**220**	0.042	**0.58**	**0.99**	**0.17**	**235**	0.038	0.54	**0.98**	**0.14**	**212**	0.055	0.47
✓	✗	✗	**0.98**	7	**0.97**	-0.26	37	0.051	0.52	**0.98**	-0.29	33	0.043	0.52	0.92	-0.31	20	0.053	0.50
✗	✓	✗	0.43	**0**	0.40	-0.36	43	0.040	**0.59**	0.39	-0.37	49	0.043	0.65	0.43	-0.33	34	0.051	0.45
✗	✗	✓	0.43	5	0.31	-0.42	22	0.019	0.51	0.33	-0.42	26	0.016	0.54	0.40	-0.40	16	0.024	**0.54**
✗	✗	✗	0.01	–	0.00	-0.65(±0.01)	17(±1)	**0.011**(±0.0004)	0.40(±0.003)	0.00	-0.64	13	**0.014**	0.63	0.00	-0.59	18	0.020	0.49

4.3 The Three Execution Stages

Table 2 crosses the three stages under the same NPO correction: IL changes which holders enter the closure, CP which copies receive and acknowledge the update, and DV whether the resulting state meets all three predicates. Closure recovery and outstanding copies separate these failure locations, and the full and all-disabled configurations reuse their main-comparison records.

Horizon diagnostics. In Figure 6, varying the mass and the number of outstanding holders separates one heavy contributor from several light ones, which carry different net restoring rates, and the ratio panel reads that separation across deployments. A positive local rate motivates a forecast, while the theorem requires its lower bound to hold across the full prefix.



(a) Horizon calibration. (b) Missed holder mass. (c) Missed copies. (d) Crossing / horizon.

Figure 6: Horizon diagnostics: early-round horizon estimates against subsequent threshold crossings, missing-holder aggregation mass, missing copies, and the mean and SD of per-case ratios over 24 matched cases per objective and deployment.

Further comparisons. Appendix E pairs localization, gate cost and sequential-request plots with their certificate, overhead, granularity, information-release and scaling tables. Each varies one execution condition while preserving the declared request and matched references.

5 Conclusion

We formulate federated unlearning over the persistent copies governed by a deletion request. **LOCUS** combines holder-local measurements, correction over their closure and verification by the parties holding the evidence. The acknowledgment gate blocks outstanding uploads, uniformly positive net reflow fixes a threshold-crossing horizon, and persistence requires non-restoring admitted updates within the release margin. The composite certificate names the corrected copies, their matched removal and retention checks, and authenticated evidence of application. Deletion completion is thus a property of the distributed execution, checked by the parties that hold each piece of its evidence.

AI USE STATEMENT

In this work, generative AI tools assisted us in proposing and refining hypotheses, writing out proofs, designing experiments and commenting on the research methodology, cleaning and reformatting data, and interpreting results. We have not used them to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims or provide critical ingredients for their proofs, to implement methods, or to assist with translation; supporting qualitative and thematic data analysis is not applicable to this work. In addition, generative AI tools helped us discover research topics, plot the figures of our experimental results, search for and summarize prior work, draft the appendix notation table, polish the text for readability, and format the references. We have reviewed all AI-assisted work: the authors checked every AI-assisted proof step against the stated assumptions and every AI-plotted figure against the experiment logs, checked the research topic for overlap with existing work through a literature survey, and verified every reference against its publisher page, CrossRef or DBLP. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017. doi: 10.1145/3133956.3133982.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, 2021. doi: 10.1109/sp40001.2021.00019.

Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. FedRecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1366–1383, 2023. doi: 10.1109/sp46215.2023.10179336.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015. doi: 10.1109/sp.2015.35.

Zheng Chai, Yujing Chen, Ali Anwar, Liang Zhao, Yue Cheng, and Huzefa Rangwala. FedAT: A high-performance and communication-efficient federated learning system with asynchronous tiers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2021. doi: 10.1145/3458817.3476211.

Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. Fast federated machine unlearning with nonlinear functional theory. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4241–4268. PMLR, 2023. URL https://proceedings.mlr.press/v202/che23b.html.

Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2022.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 2021.

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, 2025. doi: 10.1109/satml64287.2025.00033.

Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In Hugo

Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/24389bfe4fe2eba8bf9aa9203a44cdad-Abstract.html.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems 38*, pp. 1776–1803, 2025. doi: 10.52202/085713-0058.

Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. VeriFi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5720–5736, 2024. doi: 10.1109/tdsc.2024.3382321.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew

Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi (Jack) Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu (Sid) Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations*, pp. 99003–99027, 2025.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, 2023. doi: 10.18653/v1/2023.acl-long.805.

Alex Krizhevsky. Learning multiple layers of features from tiny images. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2009.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Ian Steneker, David Campbell, Brad Jokubaitis, Steven Basart, Stephen Fitz, Ponnurangam Kumaraguru, Kallol Krishna Karmakar, Uday Kiran Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28525–28550. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/li24bc.html.

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, 2021. doi: 10.1109/iwqos52092.2021.9521274.

Xiao Liu, Mingyuan Li, Guangsheng Yu, Xu Wang, Wei Ni, Lixiang Li, Haipeng Peng, and Ren Ping Liu. BlockFUL: Enabling unlearning in blockchained federated learning. *IEEE Transactions on Information Forensics and Security*, 20:6635–6650, 2025. doi: 10.1109/tifs.2025.3583109.

Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pp. 1749–1758, 2022. doi: 10.1109/infocom48880.2022.9796721.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. Tofu: A task of fictitious unlearning for llms, 2024. URL https://arxiv.org/abs/2401.06121.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 3581–3607. PMLR, 2022.

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. doi: 10.1609/aaai.v39i19.34181.

Krishna Pillutla, Kshitiz Malik, Abdelrahman Mohamed, Michael G. Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17716–17758. PMLR, 2022. URL https://proceedings.mlr.press/v162/pillutla22a.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*, pp. 53728–53741, 2023. doi: 10.52202/075280-2338.

Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. MUSE: machine unlearning six-way evaluation for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=TArmA033BU.

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, 2017. doi: 10.1109/sp.2017.41.

David M. Sommer, Liwei Song, Sameer Wagh, and Prateek Mittal. Athena: Probabilistic verification of machine unlearning. *Proceedings on Privacy Enhancing Technologies*, 2022(3):268–290, 2022. doi: 10.56553/popets-2022-0072.

Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pp. 1–10, 2023. doi: 10.1109/infocom53939.2023.10229075.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *International Conference on Learning Representations*, pp. 17978–17994, 2024.

Youming Tao, Cheng-Long Wang, Miao Pan, Dongxiao Yu, Xiuzhen Cheng, and Di Wang. Communication efficient and provable federated unlearning. *Proceedings of the VLDB Endowment*, 17(5):1119–1131, 2024. doi: 10.14778/3641204.3641220.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*, pp. 622–632, 2022. doi: 10.1145/3485447.3512222.

Wenbin Wang, Qiwen Ma, Zifan Zhang, Yuchen Liu, Zhuqing Liu, and Minghong Fang. Poisoning attacks and defenses to federated unlearning. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1365–1369, 2025. doi: 10.1145/3701716.3715494.

Jiasi Weng, Shenglong Yao, Yuefeng Du, Junjie Huang, Jian Weng, and Cong Wang. Proof of unlearning: Definitions and instantiation. *IEEE Transactions on Information Forensics and Security*, 19:3309–3323, 2024. doi: 10.1109/tifs.2024.3358993.

Zuobin Xiong, Wei Li, Yingshu Li, and Zhipeng Cai. Exact-Fun: An exact and efficient federated unlearning approach. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1439–1444, 2023. doi: 10.1109/icdm58522.2023.00188.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report, 2025. URL https://arxiv.org/abs/2501.15383.

Yiyuan Yang, Guodong Long, Tao Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. In *Advances in Neural Information Processing Systems 37*, pp. 39409–39433, 2024. doi: 10.52202/079017-1245.

Fuyao Zhang, Xinyu Yan, Tiantong Wu, Wenjie Li, Tianxiang Chen, Yang Cao, Ran Yan, Longtao Huang, Wei Yang Bryan Lim, and Qiang Yang. Oblivionis: A lightweight learning and unlearning framework for federated large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28185–28193, 2026. doi: 10.1609/aaai.v40i33.40045.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the FederatedGPT: Federated instruction tuning. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6915–6919, 2024a. doi: 10.1109/icassp48485.2024.10447454.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *CoRR*, abs/2404.05868, 2024b. doi: 10.48550/ARXIV.2404.05868. URL https://doi.org/10.48550/arXiv.2404.05868.

Xiangrong Zhu, Guangyao Li, and Wei Hu. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM Web Conference 2023*, pp. 2444–2454, 2023. doi: 10.1145/3543507.3583305.

APPENDIX CONTENTS

A Related Work **17**

B Technical Appendix **17**

B.1 Notation Table 17

B.2 Dataset Details 17

B.3 Baseline Descriptions 18

B.4 Evaluation Metrics 18

B.5 Implementation Details 19

B.6 Alternative Designs 19

B.7 Algorithm 20

B.8 Broader Impacts 20

C Proofs of Theoretical Results **20**

C.1 Assumptions and the One-Round Identity 20

C.2 The Survival Horizon 22

C.3 The Propagation Gate 23

C.4 What the Server Cannot Localize 24

C.5 Acknowledgments and State Verification 24

C.6 Degeneracy and Communication 25

D Audit Protocol **25**

D.1 Request and State Commitment 25

D.2 Independent Removal and Retention Checks 25

D.3 Acknowledgment and Re-entry 26

D.4 Audit of Continued Training 26

E Supplementary Experiments **26**

E.1 Localization and Certificate Evidence 26

E.2 Gate Cost and Communication 27

E.3 Request Granularity and Execution Sensitivity 28

E.4 Sequential Requests and Adversarial Reflow 29

E.5 Copy Classes and Information Release 29

E.6 Robustness, Scale and Protocol Coverage 30

A Related Work

Federated unlearning. Federated approaches include update calibration (Liu et al., 2021), history-based recovery (Cao et al., 2023), gradient-based removal (Pan et al., 2025; Zhang et al., 2026) and retained-knowledge distillation (Zhu et al., 2023). Other designs use checkpoint replay (Liu et al., 2022; Tao et al., 2024), selective retraining (Xiong et al., 2023) or parameter isolation (Wang et al., 2022; Su & Li, 2023) to reduce request-time work. These procedures supply corrections to a trained model. Our comparison concerns the distributed state over which a correction is executed: retained-client data determine its collateral cost, persistent copies determine its propagation scope, and later aggregation determines whether an outstanding copy can restore the withdrawn contribution. **LOCUS** coordinates these three parts through holder-local measurements and an acknowledgment gate.

Persistent state in federated learning. Personalized learning separates shared and client-specific parameters (Collins et al., 2021; Pillutla et al., 2022), including private heads (Chen & Chao, 2022), local fine-tuning (Fallah et al., 2020) and compact adapters (Sun et al., 2024; Zhang et al., 2024a). Asynchronous training maintains model versions across different participation times (Nguyen et al., 2022; Chai et al., 2021), while secure aggregation restricts access to individual uploads (Bonawitz et al., 2017). Together, these mechanisms distribute both the state and the evidence a deletion needs. Our indistinguishability constructions identify what the server transcript leaves undetermined; local readouts supply the retention and carrying measurements, and versioned acknowledgments govern each corrected copy's return to aggregation.

Certified deletion and verification. Certified removal compares an updated model with a retraining reference under stated assumptions (Guo et al., 2020; Ginart et al., 2019; Che et al., 2023). Verification also uses membership probes (Shokri et al., 2017), planted markers (Sommer et al., 2022) and authenticated execution transcripts (Eisenhofer et al., 2025; Weng et al., 2024). **LOCUS** supplies a complementary state-level declaration: governed copies, matched removal references, retention budgets and holder acknowledgments. The requester, retained clients and copy holders contribute different observations to one composite check. The reflow analysis then connects this committed state to its survival under subsequent aggregation.

B Technical Appendix

B.1 Notation Table

We summarize the commonly used notation throughout this paper in Table 3.

B.2 Dataset Details

TOFU. The two deployments use Llama-3.2-1B-Instruct and Qwen2.5-7B-Instruct on the fictitious-author benchmark (Maini et al., 2024). The 200 authors are partitioned into 20 clients, ten authors per client. The default request withdraws the ten authors of one requester, corresponding to forget05. Four designated holders receive paraphrases associated with the request; the audited affected and carrying sets are computed from their local responses, not assigned from these four identifiers.

MUSE-News. The Llama-2-7B deployment uses BBC articles partitioned among 20 clients by topic (Shi et al., 2025). The benchmark forget split defines the default request and the remaining articles define retention. Record-level and whole-client requests reuse the same topic partition. The matched reference removes exactly the requested record identifiers.

Copies and repetitions. Every client maintains a persistent rank-16 adapter. Served, private, asynchronous and synchronized copies are tracked separately. The 1B deployment uses three independent training seeds; the 7B deployments use one run each. Correction formation precedes a ten-round disconnection affecting 20% of holders. The audit then follows 300 continuation rounds. Additional model-family runs use Llama-3.2-3B, Llama-3.1-8B (Grattafiori et al., 2024) and Zephyr-7B with 20, 50 or 100 clients. The image-classification observations in Section 2 are separate controlled diagnostics, not deployments in the main comparison.

Table 3: Notation and Definitions.

Notation	Definition
$\mathcal{C}$, q	The client set and requesting client.
$\mathcal{D}_c$, $\mathcal{D}_f$	The local dataset and withdrawn records.
n_c, n	The local and total sample counts.
w, w_c	The served global model and holder-local copy.
H_c	The history of client c's uploads.
Agg	The aggregation rule of Eq. (1).
$\widetilde{w}_c$	The matched retraining reference for copy class c.
Σ	The joint state of models, data, histories and aggregation.
$\mathcal{L}_f$, $\mathcal{L}_c$	The forget loss and client retention loss.
γ_c, γ	The copy-specific and served-model removal thresholds.
δ, Δ^{c}	The candidate direction and accepted correction.
g_c, m_c	The directional retention response and carriage shortfall.
τ, η_c, η	The response tolerance and individual or common retain budget.
$\mathcal{A}$	The clients affected by the accepted correction.
$\mathcal{P}$	The correction closure of affected clients and carriers.
$\mathcal{K}[t]$	The holders acknowledging the current correction by round t.
$\mathcal{U}[t]$	The closure members still awaiting acknowledgment.
$\mathcal{E}[t]$	The clients eligible to upload in round t.
$\Delta_c[t]$, $a[t]$	The uploaded client update and aggregate round update.
$a^{\mathrm{g}}[t]$	The gated aggregate of the same proposed uploads.
$r_c[t]$	The restoring component of a client update.
M, κ	The smoothness bound and uniform net-descent lower bound.
T	The crossing horizon under uniformly positive net descent.
B, $\mu[t]$	The client-update norm bound and held aggregation mass.
w_c'	The holder-local copy after the accepted correction.
$d_f(c)$, $d_r(c)$	The removal margin and retention excess of the corrected copy.
$\mathcal{A}_{\mathrm{audit}}$, $\widehat{\mathcal{A}}$	The independently audited and reported affected sets.

B.3 Baseline Descriptions

Paired objectives. GradAscent, NPO, SimNPO, RMU and IdkDPO form candidate corrections under the same request (Jang et al., 2023; Zhang et al., 2024b; Fan et al., 2025; Li et al., 2024; Rafailov et al., 2023). Their server-side versions edit only the global checkpoint. Their LOCUS versions use the correction direction within localization, constrained correction and copy verification. This pairing separates the deletion objective from execution over persistent state.

Federated protocols. FedEraser and FedRecover use training histories and reconstruction; Exact-Fun provides an exact-unlearning execution; FedOSD supplies a retention-aware direction (Liu et al., 2021; Cao et al., 2023; Xiong et al., 2023; Pan et al., 2025). VeriFi, BlockFUL and Oblivionis add verification or distributed deletion mechanisms (Gao et al., 2024; Liu et al., 2025; Zhang et al., 2026). Each row keeps its native information interface. The overhead table records whether the forget records leave the requester, the number of exchanges and the complete elapsed request cost.

Controls. Broadcast-all sends NPO to every holder without localization or retention constraints. Gated and ungated variants share the same correction. Retrain-w replaces the served model while old copies remain; retrain-all also refits each governed copy. Both use the matched leave-requester-out training procedure and remain outside rankings.

B.4 Evaluation Metrics

Coverage and localization. Let $\mathcal{P}^*$ be the independently audited holder closure and $\mathcal{K}_{\mathrm{applied}}$ the copies that applied the current correction. Closure coverage and affected-set precision and recall are

$$\mathrm{Cov} = \frac{|\mathcal{K}_{\mathrm{applied}} \cap \mathcal{P}^*|}{|\mathcal{P}^*|}, \qquad \mathrm{Prec}_A = \frac{|\widehat{\mathcal{A}} \cap \mathcal{A}_{\mathrm{audit}}|}{|\widehat{\mathcal{A}}|}, \qquad \mathrm{Rec}_A = \frac{|\widehat{\mathcal{A}} \cap \mathcal{A}_{\mathrm{audit}}|}{|\mathcal{A}_{\mathrm{audit}}|}. \tag{8}$$

F1 is their harmonic mean; carrying-set F1 is computed analogously. Empty-set cases are tracked separately. Physical application and receipt of its acknowledgment are distinct, so a lost acknowledgment can leave the protocol pending even after application.

Removal and retention. The minimum copy margin is $M_c = \min_c[\mathcal{L}_f(w'_c) - \gamma_c]$. The maximum retention excess is $\mathrm{RL} = \max_c[\mathcal{L}_c(w'_c) - \mathcal{L}_c(\widetilde{w}_c)]$. Benchmark utility uses TOFU model utility or MUSE retain memorization. These metrics accompany each other: forget loss alone can increase when a model is damaged.

Survival. The crossing time is the first continued-training round with $\mathcal{L}_f(w[t]) < \gamma$. Noncrossing records are right-censored at 300 rounds. The displayed capped mean and the trajectory summaries share their crossing records. An early-round estimate of κ supplies the horizon diagnostic; the theorem instead requires a lower bound valid throughout the relevant prefix.

Evidence audit. Each trial contains an applied state and corresponding offline, lost-acknowledgment and stale-copy states. Sound pairs accept the first and reject all three incomplete states. False acceptance averages the three incomplete-state acceptance rates; false rejection is the applied-state rejection rate. Forged acknowledgments are scored separately. The audit uses 200 paired requests per replicate.

Resources and uncertainty. Count localization, nonlinear correction trials, propagation and acknowledgment messages. Time includes the complete request workflow. Parenthesized SDs summarize independent 1B training runs; single-run deployments have no cross-run SD. Repeated probes of one copy do not create additional independent training runs.

B.5 Implementation Details

Matched references. Fix the model architecture, initialization rule, local optimizer, participation trace and request identifiers before comparing corrections. The served model is compared with the global leave-request-out reference. Personalized copies use the corresponding personalized references. A client-specific budget uses the same loss normalization as its reference, and the server contributes an explicit removal row to the constraint system.

Nonlinear correction. At trial j, each governed holder evaluates the removal slack and gradient at $w_c + \Delta_j$; each retained client evaluates its excess-loss slack and gradient at that point. The coordinator solves the minimum-norm linearized subproblem in a trust region, then validates the proposal with the original nonlinear losses. A failed proposal reduces the trust region or triggers a new trial. Termination requires all audited constraints to pass within declared numerical tolerances; reaching a step limit is a failed correction. The final affected set is recomputed from these accepted nonlinear responses.

Versioned propagation. Bind a correction to a request identifier, the source-copy digest and the post-correction digest. A holder applies that correction once and returns an authenticated acknowledgment for that version. Replayed acknowledgments cannot clear a newer request. Holders outside the closure remain eligible; a holder inside it is eligible only after acknowledgment. Preserve the original aggregation weights when excluding an upload, as required by the held-mass calculation.

Continued training. Clone the same corrected state to compare gated and ungated execution under one participation schedule. An offline holder's return is scheduled independently of the removal audit. Record the round's forget-loss gradient, admitted updates, restoring components, total step norm and threshold margin. This distinguishes reflow from outstanding copies from movement induced by already corrected clients. An acknowledgment confirms propagation under the party model; later persistence is checked through these trajectory quantities.

B.6 Alternative Designs

This section records why the three stages of Section 3 take the form they do, by ruling out three designs that are simpler to implement. **(i) Localization from the server transcript.** The server already stores every upload, so reading the affected and carrying sets off the histories $\{H_c\}$ needs no holder participation. Theorem 8 is the obstruction: two federations that differ in which holder carries the withdrawn content can produce the same transcript, so a decision function of the transcript alone cannot separate them, and secure aggregation only widens the gap. The readouts of Section 3.2

are therefore collected from the parties whose observations support them. **(ii) A retention penalty instead of per-client constraints.** Adding a weighted retention term to the removal objective gives one scalar problem and avoids a constraint system (Maini et al., 2024; Zhang et al., 2024b). A scalarized penalty spends retention wherever it is cheapest, so no individual client is promised anything: a client whose budget η_c is small is traded away when another client's loss moves more per unit of step. The minimum-norm formulation keeps one bound per client, which is what the retention line of the certificate reads. **(iii) Delivery logs instead of an acknowledgment gate.** Recording that a correction was sent is cheaper than holding uploads until the ledger closes. A delivery log states what the server did, not what the copy did, and the gate exists because an unacknowledged copy is exactly the one whose next update reflows the withdrawn content through Eq. (1). The broadcast controls of Section 4 measure the difference this makes.

B.7 Algorithm

The three stages of Section 3 run as the single pipeline of Algorithm 1, once per deletion request. A deployment changes only how the probe set is drawn from $\mathcal{D}_f$ and which parties the participation trace makes reachable in a given round; the localization readouts, the constraint system and the three certificate lines are the same in every deployment compared in Section 4.

Each phase of Algorithm 1 produces records that the experiments report. Phase I returns the affected set and the carrying set, and Figure 7 scores both sets under reporting noise, limited probes and secure aggregation. Phase II delivers the correction to the closure and holds back uploads from members that have not acknowledged it. In Table 2, the full configuration leaves 0 outstanding copies, and disabling CP alone leaves 4. Phase III binds the removal, retention and propagation readings into one certificate, which Table 4 tests against offline, lost-acknowledgment, stale and forged states. In Table 5, **LOCUS** / NPO uses 1 round and 41 messages per request in all three deployments.

The protocol runs once per deletion request and costs one localization exchange, one broadcast with one acknowledgment per member of the closure, three commitments, and one measurement exchange per accepted or rejected nonlinear trial. If the solver takes J trials, the correction stage therefore adds J probe-and-gradient exchanges with the governed holders; reporting only the propagation broadcast would understate the revised method's cost. Phase II spans rounds only for members that were unreachable when the correction was issued, since a member present at issue time acknowledges before the next aggregation. The ledger $\mathcal{K}$ is the object the third line of Eq. (7) reads, and the gate that consults it sits at the same point in the loop as the aggregation of Eq. (1).

B.8 Broader Impacts

LOCUS lowers the cost of honoring deletion requests in federated deployments, where the obligation runs over copies no single party holds, and supports compliance with regulations such as GDPR. Two indirect concerns remain: an unlearning execution can be misused to remove evidence of unwanted training data, which deployments should mitigate by logging each request and the correction issued for it; and the acknowledgment ledger records which party holds which copy, so a deployment should keep it under the access controls that already govern the training transcript.

C Proofs of Theoretical Results

This appendix expands the round identities, the two persistence statements and the information requirements of LOCUS. All losses and copy thresholds are those of Section 3. A round index always refers to a complete aggregation step.

C.1 Assumptions and the One-Round Identity

Write $p_c = n_c/n$, so $p_c \geq 0$ and $\sum_c p_c = 1$. For proposed uploads $\Delta_c[t]$, define

$$a[t] = \sum_c p_c \Delta_c[t], \qquad r_c[t] = -\langle \nabla \mathcal{L}_f(w[t]), \Delta_c[t] \rangle, \qquad m[t] = \mathcal{L}_f(w[t]) - \gamma. \tag{9}$$

A positive $r_c[t]$ restores the withdrawn content to first order. The weights are preserved when the gate excludes a holder; they are not renormalized.

Algorithm 1 LOCUS: **L**ocality-**O**rdered **C**orrection of **U**nlearned **S**tate

Input: deletion request from client q; forget set $\mathcal{D}_f$; probe set drawn from $\mathcal{D}_f$; candidate correction δ; matched reference losses $\{\gamma_c = \mathcal{L}_f(\widetilde{w}_c), \mathcal{L}_c(\widetilde{w}_c)\}$; response tolerance τ; retention budget η; smoothness bound M; round index t.

Requester q forms δ and the probe set from $\mathcal{D}_f$ and releases both to the federation
▷ *Phase I: Influence Localization (IL)*
for each party c **in parallel do**
  Return the retention response g_c measured on $\mathcal{D}_c$ ▷ Eq. (2)
  Run the probe set through w_c and return per-token likelihoods; q assembles $\mathcal{L}_f(w_c)$ and the carriage reading m_c
end for
Affected set $\mathcal{A}$ from the responses above τ; carrying set from the positive readings; closure $\mathcal{P}$ as their union
Each c in $\mathcal{A}$ submits its retention bound η_c
repeat
  Every $h \in \mathcal{P} \cup \{\text{server}\}$ reports its measured removal margin and gradient at the trial correction
  Every non-requesting client reports its nonlinear excess-loss slack relative to $\mathcal{L}_c(\widetilde{w}_c)$ and its gradient
  Solve the local minimum-norm linearized subproblem and accept only a step that reduces measured infeasibility
until all measured constraints pass or the solver reports failure
Recompute $\mathcal{A}$ from the accepted correction's measured excess over the matched retention baselines and set $\mathcal{P}$ to its union with the carrying set
if the solve or online release audit fails **then**
  Preserve the audit record and abort the release
end if
▷ *Phase II: Correction Propagation (CP)*
Broadcast the correction to every member of $\mathcal{P}$ and open the acknowledgment ledger $\mathcal{K}$
repeat
  Each reachable member applies the correction to its own copy and returns an authenticated acknowledgment into $\mathcal{K}$
  if an upload arrives from $c \in \mathcal{P} \setminus \mathcal{K}$ **then**
    Hold the upload out of aggregation and reissue the pending correction to that member ▷ Eq. (1)
  end if
  Aggregate the admitted uploads and advance the round
until $\mathcal{P}$ is contained in $\mathcal{K}$
▷ *Phase III: Divided Verification (DV)*
Requester q scores the removal line from the probe responses of every corrected copy ▷ Eq. (7)
Each c in $\mathcal{A}$ reports the retention line from $\mathcal{L}_c(w'_c) - \mathcal{L}_c(\widetilde{w}_c)$ on $\mathcal{D}_c$
Read the propagation line off $\mathcal{P} \setminus \mathcal{K}$ and bind the three readings under the commitments issued with them
Report the survival horizon T at the reflow rate the admitted rounds realize ▷ Eq. (4)
return certificate $\big(\mathcal{A},\ \mathcal{P},\ \{g_c, m_c, \eta_c\},\ \mathcal{K},\ T\big)$; corrected state Σ'

Assumption 1 (Smoothness)**.** *The gradient of $\mathcal{L}_f$ is M-Lipschitz on a convex region containing every segment $[w[t], w[t] + a[t]]$ under consideration.*

Assumption 2 (Non-restoring admission)**.** *For the gated persistence result, every admitted upload satisfies $r_c[t] \leq 0$ at the current global state.*

The second assumption concerns the current upload, not just the copy at release. Applying a correction and acknowledging it establish the propagation predicate; they do not determine the sign of a later training update.

Let $w = w[t]$ and $a = a[t]$. The fundamental theorem of calculus gives the following exact remainder and its bound:

$$\begin{aligned} e_t &= \mathcal{L}_f(w+a) - \mathcal{L}_f(w) - \langle \nabla \mathcal{L}_f(w), a \rangle \\ &= \int_0^1 \langle \nabla \mathcal{L}_f(w+ua) - \nabla \mathcal{L}_f(w), a \rangle \, du, \\ |e_t| &\le \int_0^1 \|\nabla \mathcal{L}_f(w+ua) - \nabla \mathcal{L}_f(w)\| \, \|a\| \, du \\ &\le \int_0^1 Mu\|a\|^2 \, du = \frac{M}{2}\|a\|^2. \end{aligned} \tag{10}$$

Expanding the aggregate, rather than a single holder's contribution, yields

$$\begin{aligned} m[t+1] - m[t] &= \langle \nabla \mathcal{L}_f(w[t]), \sum_c p_c \Delta_c[t] \rangle + e_t \\ &= \sum_c p_c \langle \nabla \mathcal{L}_f(w[t]), \Delta_c[t] \rangle + e_t \\ &= -\sum_c p_c r_c[t] + e_t. \end{aligned} \tag{11}$$

Both the horizon and the gate result follow from this identity, using opposite sides of the remainder bound.

C.2 The Survival Horizon

Lemma 3 (Round descent)**.** *Under Assumption 1, let* $\kappa_t = \sum_c p_c r_c[t] - (M/2)\|a[t]\|^2$. *Then* $m[t+1] \le m[t] - \kappa_t$.

Proof. Substituting the upper remainder bound into Eq. (11),

$$m[t+1] = m[t] - \sum_c p_c r_c[t] + e_t \le m[t] - \sum_c p_c r_c[t] + \frac{M}{2}\|a[t]\|^2 = m[t] - \kappa_t. \tag{12}$$

For $\mathcal{U}[t] = \mathcal{P} \setminus \mathcal{K}[t]$, the full sum is $\sum_{c \in \mathcal{U}[t]} p_c r_c[t] + \sum_{c \notin \mathcal{U}[t]} p_c r_c[t]$. Non-restoring corrected uploads can offset the first sum; omitting them would give an unjustified descent rate. □

Theorem 4 (Survival horizon)**.** *Suppose* $m[t] \ge 0$ *and* $\kappa_{t+j} \ge \kappa > 0$ *at each round before the first strict crossing. Then the first crossing occurs no later than* $T = \lfloor m[t]/\kappa \rfloor + 1$.

Proof. For a prefix of length k on which the rate condition holds, expand every increment:

$$\begin{aligned} m[t+k] &= m[t] + \sum_{j=0}^{k-1} \big(m[t+j+1] - m[t+j] \big) \\ &= m[t] - \sum_{j=0}^{k-1} \sum_c p_c r_c[t+j] + \sum_{j=0}^{k-1} e_{t+j} \\ &\le m[t] - \sum_{j=0}^{k-1} \kappa_{t+j} \le m[t] - k\kappa. \end{aligned} \tag{13}$$

If no crossing occurred before T, all T increments would satisfy the premise. But

$$T\kappa = \left(\left\lfloor \frac{m[t]}{\kappa} \right\rfloor + 1 \right) \kappa > m[t], \qquad m[t+T] \le m[t] - T\kappa < 0, \tag{14}$$

which forces a crossing by that round. Equality at $m = 0$ still meets the removal condition, explaining the floor-plus-one convention. □

Proposition 5 (Attainment)**.** *The horizon can be attained with equality.*

Proof. Choose a unit vector u, initial margin $m_0 \geq 0$ and $\mathcal{L}_f(w) = \gamma + m_0 + \langle u, w - w[0] \rangle$ on the audited region. One outstanding client has weight $\mu > 0$ and sends $-\rho u$, with $\rho > 0$; all other uploads are zero. Here $M = 0$ and

$$\begin{aligned} a[t] &= -\mu\rho u, \qquad r_c[t] = \langle -u, -\rho u \rangle = \rho, \qquad \kappa_t = \mu\rho, \\ m[k] &= m_0 + \left\langle u, \sum_{j=0}^{k-1} a[j] \right\rangle \\ &= m_0 + \langle u, -k\mu\rho u \rangle = m_0 - k\mu\rho. \end{aligned} \tag{15}$$

Thus the first strict crossing is exactly $\lfloor m_0/(\mu\rho) \rfloor + 1$. A finite audited interval suffices; the affine loss can be shifted to be nonnegative on that interval. □

C.3 The Propagation Gate

Let $\mathcal{E}[t] = \mathcal{C} \setminus \mathcal{U}[t]$ and $a^{\mathrm{g}}[t] = \sum_{c \in \mathcal{E}[t]} p_c \Delta_c[t]$.

Theorem 6 (Gated rounds)**.** *Under Assumptions 1 and 2, a gated trajectory starting at margin $m[t] \geq 0$ satisfies*

$$m[t+k] \geq m[t] - \frac{M}{2} \sum_{j=0}^{k-1} \|a^{\mathrm{g}}[t+j]\|^2.$$

Every prefix whose right-hand side is nonnegative preserves removal.

Proof. For the gated increment, the lower remainder bound gives

$$\begin{aligned} m[t+1] &= m[t] - \sum_{c \in \mathcal{E}[t]} p_c r_c[t] + e_t^{\mathrm{g}} \\ &\geq m[t] - \sum_{c \in \mathcal{E}[t]} p_c r_c[t] - \frac{M}{2} \|a^{\mathrm{g}}[t]\|^2 \\ &\geq m[t] - \frac{M}{2} \|a^{\mathrm{g}}[t]\|^2. \end{aligned} \tag{16}$$

The last step uses $p_c \geq 0$ and $r_c[t] \leq 0$ for admitted clients. Summing the exact increments before applying the bound,

$$\begin{aligned} m[t+k] - m[t] &= -\sum_{j=0}^{k-1} \sum_{c \in \mathcal{E}[t+j]} p_c r_c[t+j] + \sum_{j=0}^{k-1} e_{t+j}^{\mathrm{g}} \\ &\geq -\frac{M}{2} \sum_{j=0}^{k-1} \|a^{\mathrm{g}}[t+j]\|^2. \end{aligned} \tag{17}$$

The squared-step sum is nondecreasing in the prefix length. Bounding it at k therefore bounds every earlier prefix as well. □

Proposition 7 (Cost of the gate)**.** *For a common set of proposed uploads with $\|\Delta_c[t]\| \leq B$, let $\mu[t] = \sum_{c \in \mathcal{U}[t]} p_c$. Then $\|a[t] - a^{\mathrm{g}}[t]\| \leq B\mu[t]$ and $\sum_{j<t} \|a[j] - a^{\mathrm{g}}[j]\| \leq B \sum_{j<t} \mu[j]$.*

Proof. The admitted terms cancel exactly:

$$\begin{aligned} a[t] - a^{\mathrm{g}}[t] &= \sum_{c \in \mathcal{C}} p_c \Delta_c[t] - \sum_{c \in \mathcal{E}[t]} p_c \Delta_c[t] = \sum_{c \in \mathcal{U}[t]} p_c \Delta_c[t], \\ \|a[t] - a^{\mathrm{g}}[t]\| &\leq \sum_{c \in \mathcal{U}[t]} p_c \|\Delta_c[t]\| \leq B \sum_{c \in \mathcal{U}[t]} p_c = B\mu[t]. \end{aligned} \tag{18}$$

Summing gives the claim and also bounds the norm of the cumulative injected discrepancy. After the last acknowledgment, $\mathcal{U}[t] = \varnothing$, hence $\mu[t] = 0$ and subsequent increments vanish. This comparison fixes the proposed uploads; independently evolving training trajectories need an additional stability argument. □

C.4 What the Server Cannot Localize

Theorem 8 (Transcript indistinguishability)**.** *There exist two federations with identical per-client uploads, histories and global iterates, but disjoint directional affected sets. A rule using only the server transcript cannot recover both sets.*

Proof. Let V be a proper subspace, let P_V be its orthogonal projector, and choose a unit $u \perp V$. Take $\theta, w[0] \in V$ and the base local objective $L_c(w) = \frac{1}{2}(w-\theta)^\top P_V(w-\theta)$. All server-started gradient steps and their aggregates stay in V by induction. A holder's persistent copy is $w_c = v_c + u$ with $v_c \in V$; it is not overwritten by those server-started steps.

Perturb one holder's curvature to $P_V + \beta uu^\top$, with $\beta > \tau \geq 0$, keeping θ fixed. At a visited point $w[t] \in V$,

$$\begin{aligned} \nabla L_c'(w[t]) - \nabla L_c(w[t]) &= \beta uu^\top (w[t]-\theta) \\ &= \beta u \langle u, w[t]-\theta\rangle = 0. \end{aligned} \tag{19}$$

The uploads coincide at the first round; equality of the aggregate preserves the next global point, so equality holds inductively for the entire transcript. At the persistent copy and candidate direction $\delta = u$,

$$\begin{aligned} g_c &= \langle P_V(v_c+u-\theta), u\rangle = 0, \\ g_c' &= \langle (P_V + \beta uu^\top)(v_c+u-\theta), u\rangle \\ &= g_c + \beta\langle u, v_c+u-\theta\rangle \|u\|^2 = \beta > \tau. \end{aligned} \tag{20}$$

Perturb holder 1 in the first federation and holder 2 in the second. Their directional affected sets are $\{1\}$ and $\{2\}$, although their server transcripts are identical. A transcript rule returns the same set on both and must fail on at least one. This establishes non-identifiability of the initial directional readout; the accepted nonlinear correction is evaluated afresh at all clients. □

Proposition 9 (Secure aggregation)**.** *Observing only aggregate uploads cannot identify the affected set in general.*

Proof. If $\mathsf{T}_1 = \mathsf{T}_2$ are the equal per-client transcripts above, deterministic aggregation A gives $\mathsf{A}(\mathsf{T}_1) = \mathsf{A}(\mathsf{T}_2)$. Any rule of this coarser observation still returns the same answer on the two distinct affected sets. The argument does not assume a uniform posterior over compatible histories. □

Proposition 10 (Readout separation)**.** *Suppose $|\widehat{g}_c - g_c| \leq \varepsilon_g$ and $|\widehat{m}_c - m_c| \leq \varepsilon_m$. Thresholding recovers the directional affected set and carrying set whenever $|g_c - \tau| > \varepsilon_g$ and $|m_c| > \varepsilon_m$ for every holder.*

Proof. For affected and unaffected holders, respectively,

$$\begin{aligned} g_c > \tau + \varepsilon_g &\implies \widehat{g}_c \geq g_c - \varepsilon_g > \tau, \\ g_c < \tau - \varepsilon_g &\implies \widehat{g}_c \leq g_c + \varepsilon_g < \tau. \end{aligned} \tag{21}$$

Replacing $(g_c, \tau, \varepsilon_g)$ by $(m_c, 0, \varepsilon_m)$ proves the carriage claim. If the gap closes, either threshold may change under an admissible reporting error. For a revised correction direction, the corresponding responses must be recomputed. □

C.5 Acknowledgments and State Verification

Theorem 11 (Holder-dependent evidence is necessary)**.** *Consider two runs identical except that one offline closure holder applies the correction in one run and not in the other. A predicate of the existing server transcript cannot be both sound and complete for physical application on these runs. Any additional evidence that decides application on every such pair must distinguish the holder's post-correction states.*

Proof. Let $b \in \{0,1\}$ indicate application and let T_b be the server transcript. The offline holder sends no upload, and its local write is absent from every server-visible channel. Thus

$$\mathsf{T}_0 = \mathsf{T}_1, \qquad f(\mathsf{T}_0) = f(\mathsf{T}_1), \qquad b_0 = 0 \neq 1 = b_1. \tag{22}$$

If f accepts both, the first run is a false acceptance; if it rejects both, the second is a false rejection. A conservative pending decision remains sound but does not decide physical completion.

Now add evidence Z_b. A correct decision $d(\mathsf{T}_b, Z_b) = b$ requires

$$d(\mathsf{T}_0, Z_0) = 0 \neq 1 = d(\mathsf{T}_1, Z_1) \quad \Longrightarrow \quad Z_0 \neq Z_1. \tag{23}$$

For every unresolved holder, the evidence must therefore depend on that holder's applied state. Under honest reporting, a version-bound authenticated acknowledgment provides this distinction. Authentication establishes the sender and version; a dishonest sender's assertion requires a separate trusted copy-state channel. Evidence can be aggregated into one message, so the necessity is informational rather than a lower bound of one network packet per holder. □

C.6 Degeneracy and Communication

Proposition 12 (Centralized reduction)**.** *Suppose only the served model survives a round, and the accepted correction's nonlinear retention audit leaves no affected client under the declared tolerance. Then the propagation and retention predicates are vacuous, and Eq.* (7) *reduces to* $\mathcal{L}_f(w') \geq \gamma$.

Proof. There are no persistent holder copies to acknowledge and $\mathcal{A} = \varnothing$. With the conventions $\max \varnothing = -\infty$ and an empty holder ledger,

$$\begin{aligned} &\min_{c \in \{\text{server}\}} \{\mathcal{L}_f(w'_c) - \gamma_c\} = \mathcal{L}_f(w') - \gamma, \\ &\max_{c \in \varnothing} \{\mathcal{L}_c(w'_c) - \mathcal{L}_c(\widetilde{w}_c)\} = -\infty \leq \eta, \qquad |\mathcal{P}_{\text{holders}} \setminus \mathcal{K}| = 0. \end{aligned} \tag{24}$$

Only the served-model removal inequality remains. A small initial directional response alone is insufficient for this reduction because the accepted correction can change direction.

For the distributed execution, count a localization exchange, J nonlinear measurement exchanges, propagation to the closure, its acknowledgments and the three evidence commitments:

$$C_{\text{protocol}} = C_{\text{IL}} + \sum_{j=1}^{J} C_{\text{trial},j} + C_{\text{broadcast}} + \sum_{c \in \mathcal{P}_{\text{holders}}} C_{\text{ack},c} + C_{\text{commitments}}. \tag{25}$$

An exact localization procedure needs additional holder-dependent information for each membership unresolved by its transcript, by Theorem 8. Batching changes packet counts but not this information requirement. □

D Audit Protocol

D.1 Request and State Commitment

An audit begins with a request identifier, the withdrawn-record identifiers, a copy inventory and the current aggregation rule. Each inventory entry specifies a holder, a copy class, a parameter digest and the reference used to set its forget threshold. Retained clients declare their loss normalization and budget before a correction is chosen. The requester commits to the forget and audit splits; the coordinator commits to the candidate correction and solver configuration. These records identify which state the certificate covers.

The audit distinguishes three times: the state at request receipt, the accepted correction before propagation, and each acknowledged post-correction state. A holder that changes its copy during the solve restarts its local validation against the new version. A reference trained without the withdrawn records is kept separate from the pre-request checkpoint, because the intended effect of removal changes both forget and retention losses.

D.2 Independent Removal and Retention Checks

After the solver accepts a correction, evaluate it again using the reserved audit records. The requester scores every governed holder and the served model against their matched γ_c. Each retained client

recomputes its excess loss over $\mathcal{L}_c(\widetilde{w}_c)$ and reports its own budget slack. The audit uses the complete holder set, not only the clients selected by the initial directional test. A newly affected client extends the closure and triggers another correction check before release.

For unequal client budgets the joint retention predicate is

$$\max_{c \in \mathcal{A}} \{\mathcal{L}_c(w'_c) - \mathcal{L}_c(\widetilde{w}_c) - \eta_c\} \leq 0. \tag{26}$$

For a common budget this is the retention line of Eq. (7). Record the minimum removal slack and the maximum budget-adjusted retention slack, together with all failing holders. A failed nonlinear solve, a missing audit response and an unsatisfied constraint are distinct outcomes; none is counted as a successful state correction.

D.3 Acknowledgment and Re-entry

The acknowledgment contains the request version, source digest, corrected digest and holder identity. The coordinator verifies authentication and checks that the acknowledgment refers to the correction currently pending for that holder. Outstanding members remain in $\mathcal{P} \setminus \mathcal{K}$ and their uploads are excluded. A returning holder applies pending corrections in version order and is re-audited before its first admitted upload. Receiving an acknowledgment for an earlier request does not close a later one.

Under the protocol-following party model, the authenticated report binds the holder's measurement to its state. A digest alone does not prove semantic removal. The requester and retained clients provide the loss evidence, while the ledger supplies propagation evidence. A composed certificate contains all three records and their common request identifier.

D.4 Audit of Continued Training

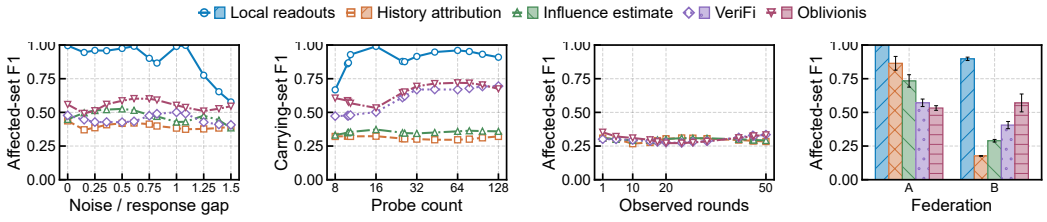
Use the same participation trace for gated and ungated continuations. At every round log the aggregate displacement, the forget-loss margin, each admitted restoring component and the held aggregation mass. The ungated horizon test checks a positive lower bound on the full net descent rate. The gated test checks the absence of outstanding contributions and the non-restoring condition on corrected updates. Accumulated curvature expenditure is compared with the release margin at every prefix of the trajectory.

Report the first failed removal round and the observation limit for censored trajectories. Break out server, active personalized, returning and deployed-copy results rather than averaging them into one loss. The communication ledger includes rejected solver trials and delayed acknowledgments; this aligns the reported cost with the complete execution whose success is being audited.

E Supplementary Experiments

E.1 Localization and Certificate Evidence

Figure 7 varies reporting noise, probe count and server visibility. The affected-set and carrying-set scores refer to different localization tasks; the paired-federation panel compares the holder evidence with the common server transcript. Bars report the mean and SD across three runs.



(a) Reporting noise. (b) Carriage probes. (c) Secure aggregation. (d) Indistinguishable pair.

Figure 7: Localization under reporting noise, limited probes and secure aggregation.

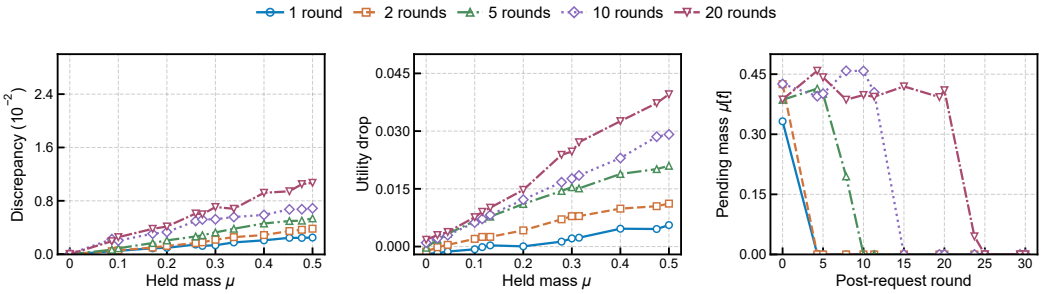
Table 4 pairs applied copies with offline, missing-acknowledgment and stale states. Sound pairs require accepting the applied member and rejecting the incomplete members of the same trial. Forged acknowledgments form a separate condition. The three server-grounded certificates accept between 93.8% and 100.0% of the offline states and hold 0.0% sound pairs. The divided certificate of **LOCUS** accepts 98.2% of the applied states, 2.7% of the offline states and none of the lost-acknowledgment states, reaching 93.8% sound pairs; adding a copy probe raises this to 95.5%.

Table 4: Certificate evidence on TOFU/Llama-1B, with 200 paired requests per state. A/O/L/S/F are pass rates for applied, offline, lost-ACK, stale and forged-ACK states. Sound, FA and FR denote sound pairs, false acceptance and false rejection (%). M and KB are messages and kilobytes per copy. Forged ACKs test the honest-holder boundary.

Evidence	A↑	O↓	L↓	S↓	F↓	Sound↑	FA↓	FR↓	M↓	KB↓
(a) Server-grounded certificates										
Global forget score FQ g	**100.0**(±0.0)	100.0(±0.0)	98.8(±1.2)	95.0(±0.5)	100.0(±0.0)	0.0(±0.0)	97.9(±0.5)	**0.0**(±0.0)	**0**	**0.0**
FQ g + history attribution	**100.0**(±0.0)	94.7(±1.2)	98.8(±0.8)	96.8(±0.8)	99.7(±0.6)	0.0(±0.0)	96.8(±0.4)	**0.0**(±0.0)	**0**	3.4(±0.3)
FQ g + influence estimate	99.2(±0.3)	93.8(±1.6)	97.5(±0.9)	91.7(±2.8)	96.0(±1.3)	0.0(±0.0)	94.3(±0.7)	0.8(±0.3)	**0**	1.2(±0.2)
(b) Single-party certificates										
Requester probe (VeriFi [TDSC'24] (Gao et al., 2024))	99.0(±0.5)	98.0(±1.0)	98.2(±0.6)	96.7(±1.9)	97.5(±0.5)	0.0(±0.0)	97.6(±0.6)	1.0(±0.5)	**0**	0.4(±0.03)
Affected retention self-report	96.5(±2.2)	62.3(±1.6)	94.7(±2.5)	54.2(±5.0)	98.8(±0.3)	0.8(±0.3)	70.4(±1.3)	3.5(±2.2)	**1**	0.9(±0.1)
Committee (BlockFUL [TIFS'25] (Liu et al., 2025))	98.8(±0.6)	95.3(±1.8)	97.3(±0.8)	93.7(±2.1)	99.5(±0.5)	0.0(±0.0)	95.4(±1.3)	1.2(±0.6)	3	6.5(±0.5)
Broadcast-all, delivery log only	**99.3**(±0.3)	98.2(±1.0)	96.2(±1.5)	95.5(±0.5)	98.2(±0.6)	0.0(±0.0)	96.6(±0.2)	**0.7**(±0.3)	**1**	**0.2**(±0.03)
(c) Divided certificate										
LOCUS, three lines bound	98.2(±0.3)	**2.7**(±1.5)	**0.0**(±0.0)	**1.8**(±0.3)	**95.2**(±0.8)	**93.8**(±1.4)	**1.5**(±0.6)	1.8(±0.3)	**1**	1.1(±0.1)
LOCUS + copy probe	97.5(±0.9)	**1.7**(±0.3)	**0.3**(±0.3)	**0.0**(±0.0)	**0.8**(±0.3)	**95.5**(±0.9)	**0.7**(±0.0)	2.5(±0.9)	2	2.0(±0.1)

E.2 Gate Cost and Communication

Figure 8 compares aggregates of the same proposed uploads under five return schedules. Pending mass identifies which uploads are still blocked, while retained-utility change records the cost of waiting for their correction. Table 5 places the required exchanges and runtime beside this comparison. Its Upload column concerns raw forget records, separately from correction messages and sketches.



(a) Common-upload cost. (b) Retained utility. (c) Pending mass.

Figure 8: Gate cost under five return schedules. The panels report the discrepancy between aggregates of the same proposed uploads, retained-utility change and pending aggregation mass.

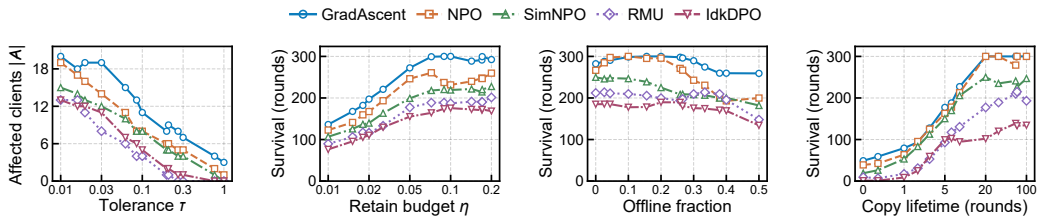
In Table 5, **LOCUS** / NPO completes a request in 1 round and 41 messages in every deployment, against 40 messages for broadcast-all, and transfers 0.8, 1.6 and 2.4 MB per client on TOFU / Llama-1B, TOFU / Qwen-7B and MUSE / Llama-7B. Its wall-clock cost is 420, 2790 and 757 seconds, against 221, 1224 and 525 seconds for server-side NPO, which edits the served model alone and requires the raw forget records to leave the requester. Broadcast-all / NPO, gated takes 319, 2114 and 906 seconds. Among the published protocols, FedOSD and VeriFi need 2 rounds, Oblivionis 3, BlockFUL 4, Exact-Fun 5, FedEraser 6 and FedRecover 8, and FedRecover is the most expensive at 160 messages and 2216, 15267 and 5621 seconds. Exact-Fun transfers 3.1, 6.8 and 10.0 MB per client. Oblivionis also requires the raw forget records to leave the requester, and BlockFUL leaves this unspecified. Retraining the served model and every copy takes 200 rounds and 4000 messages, and 61742, 454532 and 122657 seconds across the three deployments.

Table 5: Execution overhead per request. Upload: raw forget records leave the requester (✓: required; ✗: not required; –: unspecified). R: communication rounds; M: messages; MB: megabytes per client; S: wall-clock seconds, including solver trials.

		TOFU / Llama-1B				TOFU / Qwen-7B				MUSE / Llama-7B			
Execution	**Upload**	**R↓**	**M↓**	**MB↓**	**S↓**	**R↓**	**M↓**	**MB↓**	**S↓**	**R↓**	**M↓**	**MB↓**	**S↓**
Server-side NPO (Zhang et al., 2024b)	✓	**0**	**0**	**0.0**	**221**(±31)	**0**	**0**	**0.0**	**1224**	**0**	**0**	**0.0**	**525**
Broadcast-all / NPO, gated	✗	**1**	**40**	0.9(±0.1)	**319**(±40)	**1**	**40**	1.8	**2114**	**1**	**40**	2.7	906
FedEraser [IWQoS'21] (Liu et al., 2021)	✗	6	120	4.8(±0.7)	1799(±232)	6	120	7.8	10619	6	120	11.4	4952
FedRecover [S&P'23] (Cao et al., 2023)	✗	8	160	5.1(±0.1)	2216(±198)	8	160	10.8	15267	8	160	15.4	5621
Exact-Fun [ICDM'23] (Xiong et al., 2023)	✗	5	100	3.1(±0.4)	1314(±155)	5	100	6.8	7149	5	100	10.0	2810
FedOSD [AAAI'25] (Pan et al., 2025)	✗	2	**40**	1.5(±0.04)	672(±114)	2	**40**	1.9	4621	2	**40**	4.1	1804
VeriFi [TDSC'24] (Gao et al., 2024)	✗	2	42	1.4(±0.1)	624(±154)	2	42	2.9	3992	2	42	4.2	1754
BlockFUL [TIFS'25] (Liu et al., 2025)	–	4	96	3.1(±0.4)	984(±197)	4	96	5.4	7616	4	96	9.0	3771
Oblivionis [AAAI'26] (Zhang et al., 2026)	✓	3	60	2.5(±0.1)	1022(±181)	3	60	4.9	6193	3	60	5.9	2069
LOCUS / NPO	✗	**1**	41	**0.8**(±0.1)	420(±14)	**1**	41	**1.6**	2790	**1**	41	**2.4**	**757**
Retrain + all copies	✗	200	4000	106.1(±8.8)	61742(±4448)	200	4000	182.9	454532	200	4000	287.5	122657

E.3 Request Granularity and Execution Sensitivity

Figure 9 changes response tolerance, retention budget, offline fraction and copy lifetime one at a time. Table 6 then changes the requested records. Its subset band reuses the default request in Table 1; the other bands retain the same execution controls.



(a) Response tolerance. (b) Retention budget. (c) Offline holders. (d) Copy lifetime.

Figure 9: Execution sensitivity to response tolerance, retention budget, offline fraction and persistent-copy lifetime. Each curve uses one correction objective under LOCUS execution.

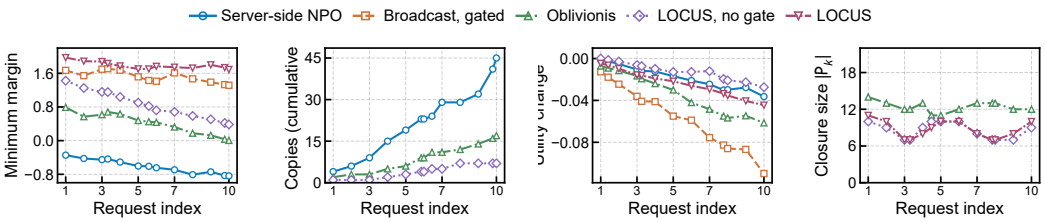
In Table 6, **LOCUS** / NPO keeps a positive minimum copy margin in every band and deployment, between 0.13 and 0.29. Every one of its survival entries is censored at 300 rounds. Server-side NPO on TOFU / Llama-1B moves from a margin of -0.29 and 33 rounds for single records to -0.84 and 10 rounds for a whole client. Its largest retention excess rises from 0.003 to 0.023. Broadcast-all / NPO, gated also reaches coverage 1.00 in every band, but on the whole-client request its retention excess reaches 0.048, 0.084 and 0.085 across the three deployments, against 0.023, 0.029 and 0.050 for **LOCUS** / NPO. FedOSD and Oblivionis stay between 0.33 and 0.71 coverage in every band.

Table 6: Request granularity across three deployments. TOFU uses four QA records, the forget split or the requesting client; MUSE uses article records, the forget split or a whole client. Metrics follow Table 1.

	TOFU / Llama-1B					TOFU / Qwen-7B					MUSE / Llama-7B				
Granularity / execution	**Cov↑**	**M_c↑**	**T↑**	**RL↓**	**U↑**	**Cov↑**	**M_c↑**	**T↑**	**RL↓**	**U↑**	**Cov↑**	**M_c↑**	**T↑**	**RL↓**	**U↑**
Single records (1% of the requester's data)															
Server-side NPO (Zhang et al., 2024b)	0.00	-0.29(±0.01)	33(±3)	**0.003**(±0.0001)	0.53(±0.005)	0.00	-0.27	33	**0.005**	**0.70**	0.00	-0.29	25	**0.005**	0.50
Broadcast-all / NPO, gated	**1.00**	**0.24**(±0.01)	≥ 300	0.007(±0.001)	**0.56**(±0.02)	**1.00**	0.23	≥ 300	0.012	0.64	**1.00**	**0.22**	≥ 300	0.013	**0.51**
FedOSD [AAAI'25] (Pan et al., 2025)	0.55(±0.02)	-0.13(±0.001)	72(±7)	**0.005**(±0.0003)	0.50(±0.03)	0.49	-0.17	70	0.008	0.62	0.51	-0.18	65	0.009	**0.51**
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.64(±0.03)	-0.10(±0.003)	**83**(±11)	**0.003**(±0.001)	0.54(±0.01)	0.71	-0.13	**93**	0.007	0.56	0.67	-0.11	**71**	**0.005**	**0.54**
LOCUS / NPO	**0.97**(±0.01)	**0.26**(±0.01)	≥ 300	**0.003**(±0.001)	0.50(±0.01)	**0.98**	**0.29**	≥ 300	**0.006**	0.57	**0.94**	**0.22**	≥ 300	**0.006**	0.44
Subset (the benchmark forget split)															
Server-side NPO (Zhang et al., 2024b)	0.00	-0.65(±0.01)	17(±1)	0.011(±0.0004)	0.40(±0.003)	0.00	-0.64	13	0.014	0.63	0.00	-0.59	18	0.020	0.49
Broadcast-all / NPO, gated	**1.00**	0.19(±0.01)	≥ 300	0.025(±0.001)	0.52(±0.01)	**1.00**	0.22	≥ 300	0.038	0.62	**1.00**	0.18	≥ 300	0.051	0.45
FedOSD [AAAI'25] (Pan et al., 2025)	0.33(±0.04)	-0.41(±0.04)	51(±10)	0.015(±0.002)	0.49(±0.01)	0.41	-0.43	53	0.028	0.66	0.42	-0.44	31	0.032	0.50
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.57(±0.02)	-0.22(±0.01)	73(±10)	0.014(±0.001)	0.52(±0.01)	0.56	-0.20	63	0.019	0.59	0.60	-0.23	63	0.029	0.43
LOCUS / NPO	0.95(±0.01)	0.20(±0.01)	≥ 300	0.010(±0.001)	0.54(±0.01)	**0.98**	**0.24**	≥ 300	0.015	**0.67**	**0.94**	**0.21**	≥ 300	0.015	**0.51**
Whole client (the requester leaves)															
Server-side NPO (Zhang et al., 2024b)	0.00	-0.84(±0.04)	10(±2)	0.023(±0.002)	0.46(±0.02)	0.00	-0.81	14	0.038	0.55	0.00	-0.94	11	0.054	0.47
Broadcast-all / NPO, gated	**1.00**	0.14(±0.01)	≥ 300	0.048(±0.006)	0.49(±0.02)	**1.00**	0.18	≥ 300	0.084	0.53	**1.00**	0.13	≥ 300	0.085	0.31
FedOSD [AAAI'25] (Pan et al., 2025)	0.42(±0.02)	-0.51(±0.08)	39(±1)	0.039(±0.003)	0.47(±0.01)	0.51	-0.56	28	0.062	0.57	0.37	-0.51	32	0.098	0.41
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.43(±0.03)	-0.51(±0.06)	48(±4)	0.028(±0.001)	**0.61**(±0.01)	0.50	-0.53	63	0.060	0.53	0.47	-0.54	38	0.077	0.46
LOCUS / NPO	0.94(±0.02)	0.19(±0.01)	≥ 300	0.023(±0.001)	0.51(±0.01)	0.92	0.17	≥ 300	0.029	0.65	0.92	0.13	≥ 300	0.050	**0.51**

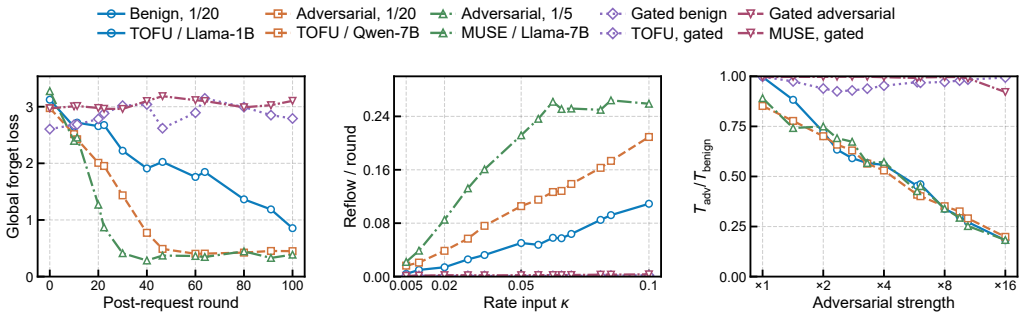
E.4 Sequential Requests and Adversarial Reflow

Figure 10 follows ten requests through copy margin, accumulated outstanding copies, retained utility and closure size. These quantities describe different costs of leaving persistent copies uncorrected. Figure 11 changes holder mass and adversarial strength, with gated continuations shown separately. A local rate forecast and a horizon justified over an entire prefix have different requirements.



(a) Removal margin. (b) Outstanding copies. (c) Retained utility. (d) Request closure.

Figure 10: Ten sequential requests: minimum copy margin, accumulated outstanding copies, retained utility and closure size. Legends identify executions across panels; invariant series are omitted.



(a) Recovery trajectories. (b) Net reflow rate. (c) Adversarial survival.

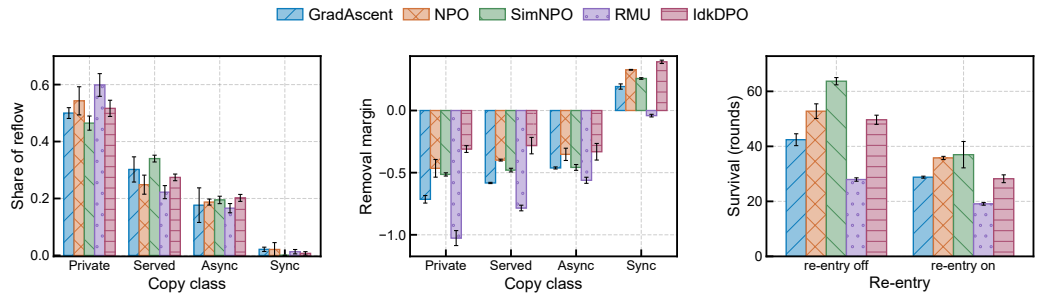
Figure 11: Benign and adversarial reflow. (a–b) Trajectories and net restoring rates under different holder masses and gates. (c) Survival relative to benign continuation as adversarial strength increases.

E.5 Copy Classes and Information Release

Figure 12 separates private, served, asynchronous and synchronous copies. Reflow shares sum to one within each run and objective; error bars are SD across three runs. Table 7 compares the information exposed by localization interfaces with the localization they retain. Extraction and membership probes assess the former; affected-set and carrying-set scores assess the latter. Releasing the full correction with the full probe set exposes an extraction recall of 0.30 and a TPR of 0.18, while the rank-8 sketch with paraphrased probes lowers them to 0.11 and 0.03 and keeps an affected-set F1 of 1.00.

Table 7: Information released by IL. Rec., R-L and EM measure extraction; AUC and TPR measure membership inference, with TPR at 1% FPR. Target records whether probes expose withdrawn targets, and reads – for the lower reference, which releases nothing and so probes nothing. A-F1, C-F1 and C measure affected-set, carrying-set and closure recovery.

IL release	Rec.↓	R-L↓	EM↓	AUC↓	TPR↓	Target	A-F1↑	C-F1↑	C↑
Forget set (upper reference)	1.00	1.00	1.00	1.00	1.00	yes	1.00	1.00	1.00
Full δ + full probe set	0.30(±0.02)	0.29(±0.02)	0.05(±0.01)	0.57(±0.04)	0.18(±0.02)	prompts	**1.00**	**1.00**	0.93(±0.07)
Rank-32 sketch of δ + full probe set	0.18(±0.03)	0.18(±0.01)	0.02(±0.002)	0.58(±0.05)	0.13(±0.01)	prompts	0.90(±0.06)	0.90(±0.09)	**0.97**(±0.06)
Rank-8 sketch of δ + 32 sampled probes	0.12(±0.01)	**0.10**(±0.01)	**0.01**(±0.01)	0.59(±0.02)	0.05(±0.004)	prompts	0.90(±0.07)	**0.92**(±0.03)	0.83(±0.03)
Rank-8 sketch + paraphrased probes	**0.11**(±0.01)	**0.08**(±0.01)	**0.01**(±0.003)	**0.54**(±0.04)	**0.03**(±0.004)	no	**1.00**(±0.01)	0.88(±0.02)	0.89(±0.05)
Rank-8 + paraphrase + likelihoods	**0.08**(±0.01)	**0.08**(±0.01)	**0.00**	**0.47**(±0.02)	**0.02**(±0.002)	no	**0.94**(±0.09)	0.82(±0.03)	**0.98**(±0.03)
Server NPO (Zhang et al., 2024b) (lower reference)	0.00	0.00	0.00	0.50	0.01	–	0.00	0.00	0.00

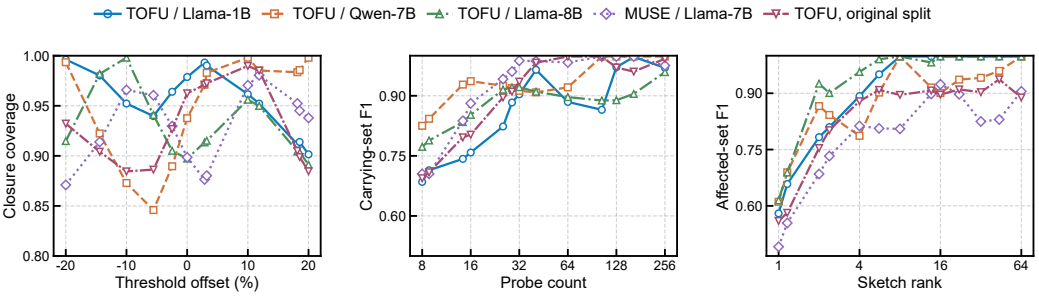


(a) Reflow contribution. (b) Copy removal margin. (c) Asynchronous return.

Figure 12: Copy-class audit. Reflow shares sum to one within each objective; removal margins compare private, served, asynchronous and synchronous copies. The last panel enables or disables asynchronous re-entry.

E.6 Robustness, Scale and Protocol Coverage

Figure 13 varies the removal threshold, probe count and correction-sketch rank. Table 8 changes model family and client count under paired requests and participation schedules. Table 9 codes 47 published protocols against four deletion requirements, each stated with the external source it derives from, and reports how often each requirement is met.



(a) Removal threshold. (b) Probe count. (c) Correction sketch.

Figure 13: Robustness to the removal threshold, number of carriage probes and rank of the correction sketch. Coverage and the two F1 scores evaluate different parts of the execution.

In Table 8, **LOCUS** / NPO sends 41, 77 and 131 messages at 20, 50 and 100 clients, and broadcast-all sends 40, 100 and 200. Every **LOCUS** survival entry is censored at 300 rounds, and its corrected closure share lies between 0.85 and 0.99 across the three model families and client counts. Server-side NPO corrects none of the copies and loses the erasure within 16 to 29 rounds. The three published protocols correct between 0.29 and 0.68 of the closure and survive between 37 and 89 rounds. At 100 clients, FedEraser sends 600 messages and takes 18429, 48256 and 57786 seconds on Llama-3B, Llama-8B and Zephyr-7B, while **LOCUS** / NPO takes 973, 2633 and 1933 seconds and broadcast-all takes 2080, 4362 and 4317 seconds. FedOSD sends 200 messages at 100 clients, as many as broadcast-all, and corrects between 0.45 and 0.52 of the closure.

In Table 9, requirement (iii), that a correction reaches every copy, has the lowest coverage: 1 of the 47 protocols satisfies it, 3 address it partially and 43 leave it unaddressed, so 9% cover it at least partially. Requirement (ii) has the highest coverage at 43%, with 6 protocols satisfying it and 14 addressing it partially. Requirement (i) is satisfied by 1 of 14 protocols from 2024 and by none of those from 2025 and 2026. Requirement (iv), that verification follows the information, is satisfied by 2 protocols and addressed partially by 8, for 21%.

Table 8: Model family and client count on TOFU. C: corrected closure share; T: survival rounds; M: messages; S: wall-clock seconds. The same request and participation schedule are paired across executions.

	Llama-3B				Llama-8B				Zephyr-7B			
Clients / execution	**C↑**	**T↑**	**M↓**	**S↓**	**C↑**	**T↑**	**M↓**	**S↓**	**C↑**	**T↑**	**M↓**	**S↓**
20 clients												
Server-side NPO (Zhang et al., 2024b)	0.00	25	**0**	509	0.00	22	**0**	**1395**	0.00	21	**0**	**1211**
Broadcast-all / NPO, gated	**1.00**	≥ 300	**40**	722	**1.00**	≥ 300	**40**	2291	**1.00**	≥ 300	**40**	2024
FedEraser [IWQoS'21] (Liu et al., 2021)	0.29	37	120	4692	0.35	47	120	10084	0.34	46	120	11946
FedOSD [AAAI'25] (Pan et al., 2025)	0.48	66	**40**	1548	0.48	55	**40**	3630	0.48	62	**40**	4304
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.58	**75**	60	2054	0.50	**86**	60	5835	0.68	**89**	60	5284
LOCUS / NPO	0.92	≥ 300	41	831	0.94	≥ 300	41	2274	**0.99**	≥ 300	41	2012
50 clients												
Server-side NPO (Zhang et al., 2024b)	0.00	29	**0**	**442**	0.00	22	**0**	**1446**	0.00	20	**0**	1374
Broadcast-all / NPO, gated	**1.00**	≥ 300	100	1350	**1.00**	≥ 300	100	3661	**1.00**	≥ 300	100	2106
FedEraser [IWQoS'21] (Liu et al., 2021)	0.35	44	300	12657	0.30	53	300	33204	0.32	49	300	25940
FedOSD [AAAI'25] (Pan et al., 2025)	0.50	61	100	3890	0.50	74	100	8515	0.50	74	100	9264
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.55	65	150	4576	0.56	75	150	16176	0.59	63	150	11489
LOCUS / NPO	**0.98**	≥ 300	77	887	**0.98**	≥ 300	77	2602	0.98	≥ 300	77	1545
100 clients												
Server-side NPO (Zhang et al., 2024b)	0.00	16	**0**	**478**	0.00	21	**0**	1500	0.00	22	**0**	**1114**
Broadcast-all / NPO, gated	**1.00**	≥ 300	200	2080	**1.00**	≥ 300	200	4362	**1.00**	≥ 300	200	4317
FedEraser [IWQoS'21] (Liu et al., 2021)	0.31	49	600	18429	0.34	40	600	48256	0.30	54	600	57786
FedOSD [AAAI'25] (Pan et al., 2025)	0.47	55	200	7708	0.45	78	200	25153	0.52	44	200	18619
Oblivionis [AAAI'26] (Zhang et al., 2026)	0.56	65	300	12690	0.56	62	300	35791	0.55	60	300	27822
LOCUS / NPO	**0.98**	≥ 300	131	973	0.85	≥ 300	131	2633	0.95	≥ 300	131	1933

Table 9: Coverage of four deletion requirements, drawn from McMahan et al. (2017), Bourtoule et al. (2021), the GDPR and Sommer et al. (2022), across 47 published protocols. S/P/N: satisfied, partial, unaddressed; years count satisfied cases; Any: satisfied or partial.

Requirement	**S**	**P**	**N**	**2024**	**2025**	**2026**	**Any (%)**
(i) Localization is joint	1	10	36	1/14 (7%)	0/18 (0%)	0/15 (0%)	23
(ii) Updates are complementary	6	14	27	3/14 (21%)	2/18 (11%)	1/15 (7%)	43
(iii) Correction reaches every copy	1	3	43	0/14 (0%)	1/18 (6%)	0/15 (0%)	9
(iv) Verification follows the information	2	8	37	0/14 (0%)	1/18 (6%)	1/15 (7%)	21