DECENTRALIZED MACHINE UNLEARNING UNDER CLIENT DISCONNECTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized machine unlearning seeks to remove the influence of a client without centralized coordination. A distinct challenge arises when a *retained client* becomes unavailable during unlearning, as its learned contribution must remain part of the desired model. We propose a collaborative decentralized unlearning framework that addresses this setting by reconstructing the counterfactual learning dynamics. For active retained clients, we adapt trajectory-based Taylor–L-BFGS reconstruction to reduce direct gradient evaluations. For a disconnected retained client, its former neighbors maintain distributed *ghost replicas* that reconstruct both its missing local updates and its participation in decentralized mixing using finite-step consensus over the active network. We analyze the resulting algorithm from an optimization-dynamics perspective, showing that its deviation from counterfactual retraining decomposes into gradient-reconstruction and ghost-reconstruction errors, with the latter decreasing with the gossip budget and network mixing quality. Experiments show that the proposed method removes the contribution targeted for unlearning while preserving that of an unavailable client, closely approaching oracle retraining.

1 INTRODUCTION

Machine unlearning seeks to remove the influence of selected training data from a learned model without incurring the cost of repeating the entire training procedure Cao & Yang (2015); Bourtoule et al. (2021). A natural reference for successful unlearning is therefore the model that would have been obtained by training from scratch without the data to be forgotten. This task becomes particularly challenging in distributed learning, where training data are held by multiple clients and unlearning may concern the complete contribution of one participating client Romandini et al. (2024). In such settings, reproducing the train-from-scratch model can require the retained clients to repeat the original collaborative training process Liu et al. (2021); Halimi et al. (2022); Cao et al. (2023).

Most studies of distributed unlearning have focused on federated architectures, in which a central server orchestrates both learning and unlearning Liu et al. (2024b). Beyond aggregating updates during training, the server coordinates the unlearning procedure and, in many approaches, retains information used to reconstruct how the model would have evolved without the client to be forgotten. Fully decentralized learning removes this coordinating entity: clients instead maintain local models and learn through repeated exchanges with their neighbors Lamri & Maniatakos (2026), and the influence of each client is propagated throughout the network. Recent works have begun to address machine unlearning in such decentralized systems using gossip-based Bayesian inference, removable representations, and dedicated decentralized unlearning mechanisms Gong et al. (2021); Ye et al. (2024); Qiao et al. (2025). These works primarily consider how to remove the contribution of the client targeted for unlearning.

Distributed operation gives rise to an additional challenge when client availability changes. Consider a client whose contribution must be forgotten, while a different client that participated in the original training becomes unavailable during unlearning. These two events have different implications: the former client's influence must be removed, whereas the latter client's contribution should remain represented in the unlearned models despite it no longer being accessible. Simply training anew using the clients that remain connected does not meet this objective, since it would erase the contributions of both clients. Thus, client disconnection makes retraining from scratch not merely expensive, but unavailable as a fallback. This raises the central question studied here: can the clients collaboratively unlearn one participant while preserving the contribution of another participant that is no longer available?

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

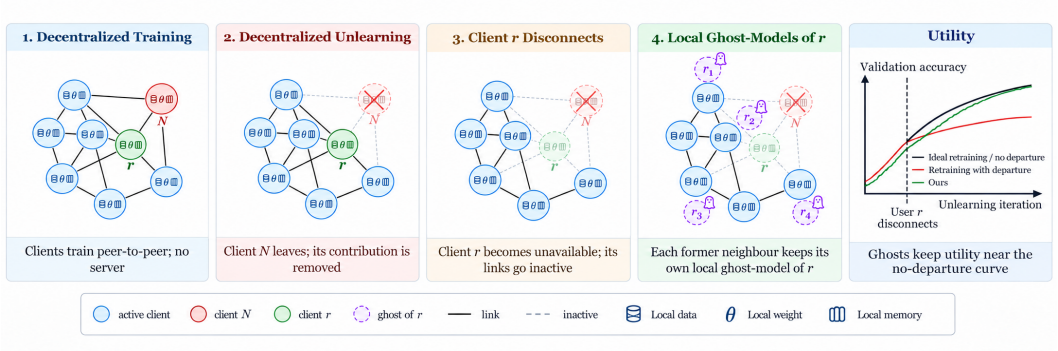

Figure 1: Decentralized unlearning under retained-client disconnection framework.

To address this challenge, we develop a decentralized unlearning framework that reconstructs the counterfactual learning dynamics using only information available to the active clients. The key difficulty is that retained-client disconnection changes the role of retraining: while retraining is ordinarily an expensive but valid unlearning reference, after client r disconnects the active clients can no longer reproduce the desired counterfactual by retraining, since excluding r would incorrectly erase the contribution of both the client to be forgotten and the disconnected client. Our method thus jointly addresses unlearning and disconnection. For available retained clients, we adapt the Taylor–L-BFGS reconstruction principle of Cao et al. (2023) to approximate counterfactual gradients while reducing direct gradient evaluations. For a disconnected retained client, its former neighbors maintain distributed *ghost replicas* that reconstruct both its missing local evolution and its otherwise absent participation through consensus over the active network, as illustrated in Fig. 1. We analyze the resulting procedure as an inexact reconstruction of counterfactual decentralized training and characterize the effects of gradient and ghost reconstruction errors. Our main contributions are thus:

- **Unlearning under client disconnections.** We formulate decentralized client-level unlearning when, in addition to the client whose contribution must be removed, a different retained client becomes unavailable. We show that retraining over the clients that remain active is not a valid fallback since it would also erase the contribution of the disconnected retained client.
- **Collaborative reconstruction with distributed ghost replicas.** We develop a decentralized unlearning algorithm that combines history-based gradient reconstruction with distributed ghost replicas. The active clients jointly reconstruct both the missing optimization updates and evolution of a disconnected client, allowing its contribution to remain represented.
- **Theoretical and empirical characterization.** We analyze the deviation of the reconstructed unlearning dynamics from counterfactual train-from-scratch learning and characterize the roles of optimization-reconstruction and decentralized-consensus errors. Experiments show that the proposed approach removes the contribution targeted for unlearning while preserving that of disconnected retained clients, closely approaching the oracle retraining reference.

2 Related Work

Federated Machine Unlearning. Federated unlearning extends machine unlearning to collaborative training in which client data remain local while a central server maintains the global model Liu et al. (2024b); Romandini et al. (2024). A common approach approximates retraining using information retained from the original optimization: FedEraser Liu et al. (2021) recalibrates historically stored client updates through additional collaboration with the retained clients; FedRecovery Zhang et al. (2023) uses historical gradient residuals together with calibrated perturbations to provide guarantees based on differential privacy; FedRecover Cao et al. (2023) reconstructs missing updates using first-order Taylor approximations and L-BFGS estimates, originally for recovering models from poisoning attacks. Subsequent approaches have targeted different aspects of the unlearning-efficiency tradeoff, including sequential client unlearning through checkpoint selection Fraboni et al. (2024), training-free reconstruction Huynh et al. (2024), and communication or memory efficiency Tao et al. (2024); Lang et al. (2026). Alternative methods avoid trajectory reconstruction via parameter negation Khalil et al.

(2025) and generative surrogates combined with transport-constrained optimization Qi et al. (2026). These diverse approaches all operate within the federated architecture with centralized orchestration.

Decentralized Machine Unlearning. Machine unlearning without a central coordinator has received substantially less attention. Early works considered gossip-based Bayesian unlearning Gong et al. (2021) and decentralized unlearning via removable distilled seed models Ye et al. (2024). More recent approaches provide guarantees under decentralized optimization: Qiao et al. (2025) use cached information and calibrated perturbations; Lamri & Maniatakos (2026) formulate certified unlearning from client-local network views; Wu et al. (2026a) use Newton-style corrective updates; and Wu et al. (2026b) derive trajectory-dependent sensitivity bounds for certified unlearning under sequential requests. A complementary line considers on-demand class unlearning in decentralized continual learning Zhang et al. (2026). These works show that client contributions can be removed without centralized orchestration, but do not address the case where a *different, retained client* becomes unavailable. Moreover, existing theoretical studies adopt certified-unlearning guarantees based on statistical indistinguishability and calibrated perturbations Qiao et al. (2025); Lamri & Maniatakos (2026); Wu et al. (2026a;b); we focus on the learning dynamics, analyzing how closely unlearning tracks the retraining process.

Unlearning under Dynamic Participation. A smaller body of work considers unlearning when the participants or information available to the unlearning procedure vary over time. In federated settings, Li et al. (2024) study unlearning in vehicular networks where clients may dynamically join or leave, while Lin et al. (2024) consider varying participation of retained clients and design client selection for unlearning. Similarly, Liu et al. (2024a) develop privacy-preserving federated unlearning under dynamic participation. Dynamic operation has also been considered without centralized orchestration: PDUDT Qiao et al. (2025) and TRACE-DU Wu et al. (2026b) accommodate time-varying decentralized topologies, while Zhang et al. (2026) consider on-demand class unlearning during decentralized continual learning when data of previous tasks become inaccessible. These settings do not address a retained client whose future computations become unavailable while its learned contribution must remain part of the desired model. This distinction is central, as client disconnection is not an unlearning request, and excluding the unavailable client from subsequent retraining would erase knowledge that should be retained.

3 Decentralized Unlearning Setup

3.1 Decentralized Training

We consider a decentralized learning system comprising N clients indexed by $\mathcal{V} = \{1, \dots, N\}$. The clients communicate over a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $(i, j) \in \mathcal{E}$ indicates that clients i and j can exchange messages, and $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ denotes the neighborhood of client i. Each client i holds a private local dataset $\mathcal{D}_i$ associated with a local objective $f_i : \mathbb{R}^d \to \mathbb{R}$. The clients collaboratively seek to minimize an aggregate objective over the model parameters $\mathbf{x} \in \mathbb{R}^d$, given by $F_N(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x})$. In the absence of a central server, each client maintains a local model $\mathbf{x}_{i,t}$ at time t. At each round, client i evaluates its local gradient and performs a local gradient step

$$\mathbf{z}_{i,t} = \mathbf{x}_{i,t} - \eta \nabla f_i(\mathbf{x}_{i,t}), \tag{1}$$

with step size $\eta > 0$. For clarity of the theoretical development, we formulate the decentralized optimization dynamics using full local gradients, although the proposed reconstruction mechanism is also applicable with stochastic mini-batch gradients, as evaluated experimentally.

To support subsequent unlearning, we assume that the message exchanged by each client with its neighbors contains both its current local model $\mathbf{x}_{i,t}$ and its gradient $\nabla f_i(\mathbf{x}_{i,t})$. Hence, every neighbor can also form the locally updated state $\mathbf{z}_{i,t}$ in (1). The decentralized learning dynamics themselves follow the standard mixing operation Yuan et al. (2016); Larsson & Michelusi (2025)

$$\mathbf{x}_{i,t+1} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{z}_{j,t}, \tag{2}$$

where $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{N \times N}$ is a mixing matrix on $\mathcal{G}$. By defining $\mathbf{X}_t = [\mathbf{x}_{1,t}, \dots, \mathbf{x}_{N,t}]^\top$, and $\mathbf{G}_t = \left[\nabla f_1(\mathbf{x}_{1,t}), \dots, \nabla f_N(\mathbf{x}_{N,t})\right]^\top$, the decentralized training dynamics can be written as $\mathbf{X}_{t+1} = \mathbf{W}\left(\mathbf{X}_t - \eta \mathbf{G}_t\right)$. The original training procedure runs for T rounds, producing $\{\mathbf{X}_t\}_{t=0}^{T}$.

The unlearning procedure can reuse information generated along this trajectory. Accordingly, during the original training process, each client retains its locally generated model-gradient pairs and those made available by its neighbors. In particular, client i maintains the history

$$\mathcal{H}_i = \{(\mathbf{x}_{j,t}, \nabla f_j(\mathbf{x}_{j,t})) : j \in \mathcal{N}_i \cup \{i\},\ t = 0, \dots, T-1\}\,. \tag{3}$$

This history contains optimization information only: the private datasets $\{\mathcal{D}_i\}$ are never exchanged. As detailed in Section 4, the stored $\mathcal{H}_i$ enables the clients to approximate otherwise costly gradient evaluations during unlearning and to reconstruct the evolution of a disconnected client.

3.2 Decentralized Unlearning

We next describe the unlearning setup. For clarity of exposition, we formulate a setting with a single client requesting unlearning and a single retained client that disconnects. Yet, the framework naturally extends to multiple retained-client disconnections, as demonstrated experimentally in the appendix.

Unlearning Objective. After T steps of the original training procedure, one client (assumed to be client N without loss of generality) requests that its contribution be removed from the learned models. We consider *unassisted unlearning*, in which client N becomes unavailable and can neither provide access to its data nor compute or communicate additional updates Romandini et al. (2024). The set of retained clients is $\mathcal{R} = \mathcal{V} \setminus \{N\} = \{1, \dots, N-1\}$, and the corresponding learning objective is

$$F_{N-1}(\mathbf{x}) = \frac{1}{N-1} \sum_{i \in \mathcal{R}} f_i(\mathbf{x}). \tag{4}$$

Removing client N also modifies the communication network. We denote by $\bar{\mathcal{G}} = (\mathcal{R}, \bar{\mathcal{E}})$ the subgraph induced by the retained clients and by $\bar{\mathcal{N}}_i$ the corresponding neighborhood of client $i \in \mathcal{R}$. We assume that $\bar{\mathcal{G}}$ remains connected. The mixing weights are recomputed over $\bar{\mathcal{G}}$ as $\bar{\mathbf{W}} = [\bar{w}_{ij}] \in \mathbb{R}^{(N-1)\times(N-1)}$.

The natural reference for unlearning is the counterfactual model that would have been obtained had client N never participated in training. We thus consider decentralized retraining from scratch over the complete retained-client set $\mathcal{R}$, using the same optimization protocol as in Section 3.1. Let $\widetilde{\mathbf{X}}_t = [\widetilde{\mathbf{x}}_{1,t}, \dots, \widetilde{\mathbf{x}}_{N-1,t}]^\top$ denote its state at round t, initialized as $[\mathbf{x}_{1,0}, \dots, \mathbf{x}_{N-1,0}]^\top$. Its evolution follows

$$\widetilde{\mathbf{X}}_{t+1} = \bar{\mathbf{W}} \left(\widetilde{\mathbf{X}}_t - \eta \widetilde{\mathbf{G}}_t \right), \qquad \widetilde{\mathbf{G}}_t = [\nabla f_1(\widetilde{\mathbf{x}}_{1,t}), \dots, \nabla f_{N-1}(\widetilde{\mathbf{x}}_{N-1,t})]^\top. \tag{5}$$

We refer to $\{\widetilde{\mathbf{X}}_t\}_{t=0}^T$ as the *counterfactual retraining trajectory*.

The objective of decentralized unlearning is to reproduce, or closely approximate, this counterfactual trajectory without repeating the complete training procedure. In particular, the retained clients should remove the influence of client N while approaching the models produced by retraining over $\mathcal{R}$.

Unlearning under Client Disconnection. The formulation above implicitly assumes that every client whose contribution is to be retained remains available during unlearning. We consider a more challenging setting in which a retained client becomes unavailable. Let $r \in \mathcal{R}$ denote such a client, and let τ_r be its final active round. Up to round τ_r, client r participates in unlearning; after its departure, its dataset and computations are no longer accessible. We assume that the graph induced by the remaining available clients, $\bar{\mathcal{G}}[\mathcal{R} \setminus \{r\}]$, remains connected. We denote the set of active clients at round t by $\mathcal{A}_t$.

The contribution of client r is still required to be represented in the desired model, and the target objective remains F_{N-1} in (4). Accordingly, the reference trajectory remains $\{\widetilde{\mathbf{X}}_t\}$ over the complete retained-client set $\mathcal{R}$. This distinguishes the roles of clients N and r: the influence of client N must be erased, while that of client r must be preserved even after it no longer participates. This distinction prevents retraining with the available clients from serving as a valid fallback. Following the disconnection, such retraining would instead optimize $F_{\mathcal{A}}(\mathbf{x}) = \frac{1}{N-2} \sum_{i \in \mathcal{R}\setminus\{r\}} f_i(\mathbf{x})$, whose outcome corresponds to forgetting both clients N and r. The resulting unlearning problem is thus fundamentally different from ordinary decentralized retraining. After τ_r, the active clients must continue approximating a trajectory whose evolution depends on a local objective and communication updates that are no longer directly available. The goal is to do so using only information available to the active clients, while simultaneously tracking the counterfactual retraining dynamics and maintaining agreement across their local models.

4 DECENTRALIZED UNLEARNING METHOD

Our approach treats decentralized unlearning as the reconstruction of the counterfactual retraining dynamics (5). This involves two distinct challenges: (i) even when all retained clients remain available, reproducing the counterfactual trajectory through exact retraining would require repeatedly evaluating their local gradients; and (ii) when a retained client disconnects, gradient reconstruction alone is insufficient. We address (i) by adapting the Taylor–L-BFGS mechanism of FedRecover Cao et al. (2023) to decentralized optimization. Rather than repeatedly evaluating the gradient at the current unlearning iterate, each retained client estimates its change relative to a stored training iterate using L-BFGS. Specifically, L-BFGS retains only a small number of recent pairs of model displacement and the corresponding gradient difference, and uses them to construct an implicit approximation of the local Hessian action.

The second challenge requires an additional collaborative mechanism. Once client r disconnects, its former neighbors maintain *ghost replicas* representing estimates of its state had it remained active. Reconstructing the subsequent mixing step requires combining this estimated local state with the evolving states of client r's former neighbors, which are distributed across the active network. We next detail these mechanisms, and summarize the unlearning algorithm.

4.1 TRAJECTORY RECONSTRUCTION FOR RETAINED CLIENTS

We initialize the unlearning procedure from the retained-client portion of the original initialization, $\bar{\mathbf{X}}_0 = \mathbf{X}_{0,\mathcal{R}}$, and denote the model of retained client i at unlearning round t by $\bar{\mathbf{x}}_{i,t}$. Reproducing the counterfactual trajectory (5) would require evaluating $\nabla f_i(\bar{\mathbf{x}}_{i,t})$ at every round. Instead, we exploit model-gradient information generated during the original training process to approximate this gradient.

To motivate the reconstruction, for a twice-differentiable objective define the mean Hessian between the stored training iterate $\mathbf{x}_{i,t}$ and the current unlearning iterate $\bar{\mathbf{x}}_{i,t}$ as

$$\overline{\mathbf{H}}_{i,t} = \int_0^1 \nabla^2 f_i(\mathbf{x}_{i,t} + \alpha(\bar{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}))\, d\alpha. \tag{6}$$

The gradient difference then satisfies $\nabla f_i(\bar{\mathbf{x}}_{i,t}) = \nabla f_i(\mathbf{x}_{i,t}) + \overline{\mathbf{H}}_{i,t}(\bar{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t})$ Larsson & Michelusi (2025). We therefore approximate the unavailable second term as

$$\mathbf{g}_{i,t}^{\mathrm{app}} = \nabla f_i(\mathbf{x}_{i,t}) + \widehat{\mathbf{h}}_{i,t}, \qquad \widehat{\mathbf{h}}_{i,t} \approx \overline{\mathbf{H}}_{i,t}(\bar{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}). \tag{7}$$

The Hessian is never explicitly formed. Whenever an exact gradient is evaluated at some round ℓ, client i obtains the secant pair $\big(\bar{\mathbf{x}}_{i,\ell} - \mathbf{x}_{i,\ell}, \nabla f_i(\bar{\mathbf{x}}_{i,\ell}) - \nabla f_i(\mathbf{x}_{i,\ell})\big)$. L-BFGS uses a buffer of the recent such pairs to construct an implicit quasi-Newton approximation of the Hessian Nocedal & Wright (2006), which is applied to $\bar{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}$ to obtain $\widehat{\mathbf{h}}_{i,t}$. The resulting computation is detailed in Appendix A.

To control the accumulation of reconstruction errors, exact gradients are evaluated at a prescribed subset of rounds $\mathcal{T}_{\mathrm{exact}}$, including an initial stage that fills the L-BFGS memory and periodic correction rounds. At the remaining rounds, we use (7). As an additional safeguard, an approximate gradient whose magnitude exceeds a client-specific threshold Γ_i, determined from the scale of the historical gradients, is replaced by an exact evaluation. Thus, for an active client,

$$\widehat{\mathbf{g}}_{i,t} = \begin{cases} \nabla f_i(\bar{\mathbf{x}}_{i,t}), & t \in \mathcal{T}_{\mathrm{exact}} \text{ or } \|\mathbf{g}_{i,t}^{\mathrm{app}}\|_\infty > \Gamma_i, \\ \mathbf{g}_{i,t}^{\mathrm{app}}, & \text{otherwise.} \end{cases} \tag{8}$$

Whenever the first case requires an exact evaluation, the resulting gradient is also used to refresh the L-BFGS memory. Client i then performs the local update

$$\mathbf{z}_{i,t} = \bar{\mathbf{x}}_{i,t} - \eta \widehat{\mathbf{g}}_{i,t}. \tag{9}$$

As long as all retained clients remain active, these locally reconstructed updates are mixed according to the retained-client matrix $\bar{\mathbf{W}}$, exactly as in counterfactual decentralized retraining. The difference is that most gradient evaluations are replaced by the trajectory-based approximations above.

4.2 RECONSTRUCTING A DISCONNECTED CLIENT WITH DISTRIBUTED GHOST REPLICAS

We now consider the departure of client r after round τ_r. The challenge is twofold: (i) client r can no longer evaluate the local gradient required to continue its counterfactual optimization trajectory; and (ii) even if this gradient could be estimated, the subsequent mixing operation that determines its next model is also missing. We reconstruct these two operations distributively.

Distributed Ghost Replicas. At disconnection, each former neighbor $i \in \bar{\mathcal{N}}_r$ initializes a local *ghost replica* of client r from its final available post-mixing state as $\bar{\mathbf{x}}_{r,\tau_r+1}^{(i)} = \bar{\mathbf{x}}_{r,\tau_r+1}$. The L-BFGS information (whose memory is assumed to be initialized by round τ_r) available for client r is likewise fixed and used thereafter by its ghost owners. At round $t > \tau_r$, ghost replica owner i applies the same trajectory reconstruction principle as an active client, but around its own current replica:

$$\widehat{\mathbf{g}}_{r,t}^{(i)} = \nabla f_r(\mathbf{x}_{r,t}) + \widehat{\mathbf{h}}_{r,t}^{(i)}, \qquad \widehat{\mathbf{h}}_{r,t}^{(i)} \approx \nabla^2 f_r(\mathbf{x}_{r,t})\big(\bar{\mathbf{x}}_{r,t}^{(i)} - \mathbf{x}_{r,t}\big), \tag{10}$$

where $\widehat{\mathbf{h}}_{r,t}^{(i)}$ is obtained from the frozen L-BFGS information. Unlike an active client, however, a ghost owner cannot fall back to an exact evaluation because $\mathcal{D}_r$ is no longer accessible. We therefore use the same historical-scale test as a validity check and define the locally updated ghost state as

$$\mathbf{z}_{r,t}^{(i)} = \begin{cases} \bar{\mathbf{x}}_{r,t}^{(i)} - \eta \widehat{\mathbf{g}}_{r,t}^{(i)}, & \|\widehat{\mathbf{g}}_{r,t}^{(i)}\|_\infty \le \Gamma_r, \\ \bar{\mathbf{x}}_{r,t}^{(i)}, & \|\widehat{\mathbf{g}}_{r,t}^{(i)}\|_\infty > \Gamma_r. \end{cases} \tag{11}$$

Thus, an unreliable ghost gradient freezes the local optimization step rather than removing it from the decentralized dynamics. This preserves the mixing coefficients of the original retained-client system.

Reconstructing Mixing Step. If client r had remained available, its next state would be obtained by mixing its local update with those of its neighbors according to the rth row of $\bar{\mathbf{W}}$. As its neighbors are active, their contributions $\{\bar{w}_{rj}\mathbf{z}_{j,t}\}_{j\in\bar{\mathcal{N}}_r}$ are still available. The missing component is the self-update of client r, for which we have local ghost estimates. Letting $M_r = |\bar{\mathcal{N}}_r|$, the mixed state is reconstructed as

$$\mathbf{m}_{r,t} = \frac{\bar{w}_{rr}}{M_r} \sum_{i\in\bar{\mathcal{N}}_r} \mathbf{z}_{r,t}^{(i)} + \sum_{j\in\bar{\mathcal{N}}_r} \bar{w}_{rj}\mathbf{z}_{j,t}. \tag{12}$$

The first term aggregates the distributed estimates of the missing self-update, while the second consists of states that remain available. No active client possesses all terms in (12). We thus recover $\mathbf{m}_{r,t}$ through K ratio-consensus iterations over the post-disconnection graph $\bar{\mathcal{G}}[\mathcal{R}\setminus\{r\}]$. Let $\mathbf{P}$ be the mixing matrix for this graph. Each former neighbor initializes a numerator with its available contribution $\bar{w}_{ri}\mathbf{z}_{i,t}$, while each ghost owner inserts an equal share of the estimated self-contribution. To formulate this, we write the matrix $\mathbf{A}^{(k)}$ and vector $\mathbf{b}^{(k)}$ whose entries are initialized to $\mathbf{a}_i^{(0)} = \big(\bar{w}_{ri}\mathbf{z}_{i,t} + \frac{\bar{w}_{rr}}{M_r}\mathbf{z}_{r,t}^{(i)}\big)\mathbb{I}_{\{i\in\bar{\mathcal{N}}_r\}}$ and $b_i^{(0)} = \frac{1}{M_r}\mathbb{I}_{\{i\in\bar{\mathcal{N}}_r\}}$, with $\mathbb{I}_{\{\cdot\}}$ being the indicator operator. The network then iterates

$$\mathbf{A}^{(k+1)} = \mathbf{P}\mathbf{A}^{(k)}, \qquad \mathbf{b}^{(k+1)} = \mathbf{P}\mathbf{b}^{(k)}, \qquad k = 0, \dots, K-1. \tag{13}$$

Each ghost owner i then sets $\bar{\mathbf{x}}_{r,t+1}^{(i)} = \mathbf{a}_i^{(K)}/b_i^{(K)}$. As K increases, these local ratios approach the desired mixed state $\mathbf{m}_{r,t}$; for finite K, they form local approximations used as the next ghost replicas.

The ghost replicas also allow the physically active neighbors of client r to preserve its contribution in their own mixing operations. Specifically, for each active client $j \in \mathcal{A}_t$,

$$\bar{\mathbf{x}}_{j,t+1} = \sum_{k\in(\bar{\mathcal{N}}_j\setminus\{r\})\cup\{j\}} \bar{w}_{jk}\mathbf{z}_{k,t} + \mathbb{I}_{\{j\in\bar{\mathcal{N}}_r\}}\bar{w}_{jr}\mathbf{z}_{r,t}^{(j)}. \tag{14}$$

Hence, clients that were not connected to r continue their ordinary decentralized updates, whereas each former neighbor replaces only the unavailable contribution of r by its local ghost estimate.

4.3 Complete Algorithm

The complete procedure alternates between local trajectory reconstruction and decentralized mixing. Before the retained-client disconnection, all clients in $\mathcal{R}$ follow the reconstructed gradient dynamics of Section 4.1. After client r disconnects, the active clients continue the same procedure, while its former neighbors additionally evolve local ghost replicas according to Section 4.2. Algorithm 1 summarizes the resulting method; the compact L-BFGS computation and the implementation details of the ratio-consensus procedure are deferred to Appendices A and B, respectively.

Algorithm 1 Collaborative Decentralized Unlearning with Ghost Replicas

Require: Retained graph $\bar{\mathcal{G}}$ and mixing matrix $\bar{\mathbf{W}}$; available training history $\{\mathcal{H}_i\}$; step size η;
Initialize $\bar{\mathbf{X}}_0 = \mathbf{X}_{0,\mathcal{R}}$ and the L-BFGS memories
for $t = 0, \dots, T-1$ **do**
  for all $i \in \mathcal{A}_t$ **do**
    Reconstruct $\widehat{\mathbf{g}}_{i,t}$ using (7)-(8)
    Set $\mathbf{z}_{i,t} \leftarrow \bar{\mathbf{x}}_{i,t} - \eta \widehat{\mathbf{g}}_{i,t}$
  if $t \le \tau_r$ **then**
    Mix the retained-client updates using $\bar{\mathbf{W}}$
    if $t = \tau_r$ **then**
      Initialize a ghost replica of client r at every $i \in \bar{\mathcal{N}}_r$
  else
    for all $i \in \bar{\mathcal{N}}_r$ **do**
      Reconstruct the ghost gradient $\widehat{\mathbf{g}}_{r,t}^{(i)}$ using (10)
      Form $\mathbf{z}_{r,t}^{(i)}$ using (11)
    Update all active clients using (14)
    Perform K ratio-consensus iterations using (13)
    Update the ghost replicas $\bar{\mathbf{x}}_{r,t+1}^{(i)} = \mathbf{a}_i^{(K)} / b_i^{(K)}$

5 Theoretical Analysis

We analyze the proposed unlearning procedure from an optimization-dynamics perspective. Our objective is to characterize how closely the reconstructed decentralized trajectory follows the counterfactual retraining trajectory in (5). We show that the proposed dynamics can be viewed as an inexact version of decentralized retraining, with perturbations arising from gradient reconstruction and from reconstructing the evolution of the disconnected retained client. Proofs are delegated to Appendix C.

Perturbed Retraining Dynamics. For the analysis, let $\bar{\mathbf{X}}_t \in \mathbb{R}^{(N-1)\times d}$ represent the reconstructed retained-client state. Before disconnection, its rth row is the physical model of client r; afterwards, we identify this row with the average ghost state $\bar{\mathbf{x}}_{r,t}^{\mathrm{gh}} = \frac{1}{|\bar{\mathcal{N}}_r|} \sum_{i \in \bar{\mathcal{N}}_r} \bar{\mathbf{x}}_{r,t}^{(i)}$. Let $\widehat{\mathbf{G}}_t$ collect the gradients used by the reconstructed dynamics, with its rth row after disconnection being the corresponding average ghost-gradient estimate. Define the gradient-reconstruction error as $\mathbf{B}_t = \widehat{\mathbf{G}}_t - \nabla \mathbf{f}_{\mathcal{R}}(\bar{\mathbf{X}}_t)$, where $\nabla \mathbf{f}_{\mathcal{R}}(\mathbf{X})$ stacks $\{\nabla f_i(\mathbf{x}_i)\}_{i \in \mathcal{R}}$. The implemented dynamics are

$$\bar{\mathbf{X}}_{t+1} = \bar{\mathbf{W}} \left(\bar{\mathbf{X}}_t - \eta \widehat{\mathbf{G}}_t \right) + \mathbf{C}_t = \bar{\mathbf{W}} \left(\bar{\mathbf{X}}_t - \eta \nabla \mathbf{f}_{\mathcal{R}}(\bar{\mathbf{X}}_t) \right) - \eta \bar{\mathbf{W}} \mathbf{B}_t + \mathbf{C}_t, \tag{15}$$

where $\mathbf{C}_t = \mathbf{0}$ while client r remains active, and after its disconnection $\mathbf{C}_t$ collects the discrepancy induced by the distributed reconstruction of its local and mixing updates. In particular, $\mathbf{C}_t$ captures both consensus error and disagreement among the ghost replicas used by the neighbors of client r.

Assumptions. We assume that $\bar{\mathbf{W}}$ is symmetric and doubly stochastic over a connected graph, and that every retained objective f_i is μ-strongly convex and L-smooth, with $0 < \mu \le L$. We further assume that the gradient-reconstruction errors satisfy $\|\mathbf{B}_t\|_F \le \beta_t$, and write $\|\mathbf{C}_t\|_F \le \gamma_t$, where $\gamma_t = 0$ for $t \le \tau_r$. While β_t abstracts the realized Taylor–L-BFGS approximation quality, Appendix C.2 derives an explicit bound on γ_t in terms of the gossip budget K and the spectral contraction of $\bar{\mathcal{G}}[\mathcal{R} \setminus \{r\}]$.

Counterfactual Tracking. Our main result bounds the deviation from the trajectory $\{\widetilde{\mathbf{X}}_t\}$ in (5).

Theorem 1. *Suppose the above assumptions hold and* $0 < \eta \le 2/(L+\mu)$. *Let* $c = 1 - \eta\mu \in [0, 1)$. *Since* $\bar{\mathbf{X}}_0 = \widetilde{\mathbf{X}}_0$, *then, for every* $t \in \{1, \dots, T\}$, *it holds that*

$$\left\| \bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t \right\|_F \le \eta \sum_{k=0}^{t-1} c^{\,t-1-k} \beta_k + \sum_{k=\tau_r+1}^{t-1} c^{\,t-1-k} \gamma_k. \tag{16}$$

Theorem 1 separates the two sources of deviation from counterfactual retraining. Gradient-reconstruction errors affect the trajectory throughout unlearning, whereas ghost-reconstruction errors

appear only after the retained client disconnects; in both cases, errors introduced at earlier rounds are geometrically discounted. In particular, if $\beta_t \le \beta$ and $\gamma_t \le \gamma$, then

$$\left\|\bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t\right\|_F \le \frac{\beta}{\mu}(1-c^t) + \frac{\gamma}{\eta\mu}\left(1 - c^{\max\{(t-\tau_r-1),0\}}\right). \tag{17}$$

Thus, exact reconstruction recovers the counterfactual dynamics, while persistent bounded reconstruction errors confine the unlearning trajectory to a controlled neighborhood thereof.

Relation to Retained-Client Optimum. Let $\mathbf{x}^\star = \arg\min_{\mathbf{x}} F_{N-1}(\mathbf{x})$. Tracking counterfactual retraining does not necessarily imply convergence to a confined distance from $\mathbf{X}^\star = \mathbf{1}(\mathbf{x}^\star)^\top$, as constant-step decentralized gradient descent can exhibit a topology- and heterogeneity-dependent bias. The two effects are separated by

$$\left\|\bar{\mathbf{X}}_t - \mathbf{X}^\star\right\|_F \le \underbrace{\left\|\bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t\right\|_F}_{\text{unlearning reconstruction}} + \underbrace{\left\|\widetilde{\mathbf{X}}_t - \mathbf{X}^\star\right\|_F}_{\text{decentralized retraining}}. \tag{18}$$

The first term is controlled by Theorem 1; the second consists of a geometrically decaying transient and the standard persistent decentralized optimization bias, which depends on the network topology and objective heterogeneity. An explicit bound, together with the computational, communication, and storage complexity analyses, is provided in Appendix D. We note that client disconnection does not alter the contraction factor c of the retraining dynamics; rather, it enters through the ghost-reconstruction error γ_t. The two error sources can thus be controlled through distinct mechanisms: exact-gradient refreshes reduce β_t, whereas increasing the gossip budget reduces the consensus component of γ_t geometrically.

6 Experimental Study

Setup and Evaluation. We evaluate our method on MNIST and CIFAR-10 using convolutional networks trained with mini-batch Stochastic Gradient Descent (SGD). We use Erdős–Rényi graphs with edge probability $p = 0.5$ and Metropolis mixing, with $N = 20$ clients for MNIST and $N = 10$ for CIFAR-10. To distinguish information that should be *forgotten* from information that should be *preserved*, we control the data heterogeneity of the two relevant clients. A prescribed fraction of the disconnected retained client's data may belong to a client-specific label absent from the other clients, making its contribution difficult to recover once it disconnects. The same construction can be applied to the client designated for unlearning, in which case successful forgetting necessarily entails some loss of predictive information. We additionally backdoor 50% of the forgotten client's data and report attack success rate (ASR) as a measure of residual influence. Performance is evaluated using test accuracy, ASR, and relative consensus error. Complete experimental settings, hyperparameters, and additional results with additional datasets and models are provided in Appendix E. The implementation is available at `https://github.com/D-UNLEARNING/Client-Disconnections.git`.

We compare against the proposed method without ghost reconstruction (*No Ghost*), TRACE-DU Wu et al. (2026b), and Certified-DFU Wu et al. (2026a). We additionally report two retraining references: *Retraining with Departure*, which continues only with the physically available clients after r disconnects, and *Retraining without Departure*, which removes only the forgotten client and serves as the desired oracle counterfactual. For TRACE-DU, the retained client is disconnected at the same round as in our method and training continues over the remaining graph; details of all baseline adaptations are given in Appendix E.1.4. To quantify the computational benefit of trajectory reconstruction, we also report the *direct-gradient evaluation rate*, defined as the fraction of active-client rounds in which a gradient is directly evaluated rather than reconstructed; retraining requires direct evaluation in every such round.

Preserving an Unavailable Client. Table 1 shows that the proposed method consistently remains closest to the oracle retraining reference. The benefit of ghost reconstruction is particularly clear when the disconnected client carries information unavailable elsewhere: on MNIST with an IID forgotten client, our method attains 97.52% accuracy, compared with 89.49% without ghost reconstruction and 89.08% when simply retraining after departure, versus 97.76% for the oracle. The same behavior is observed on CIFAR-10, where our method reaches 82.18% compared with 78.56% and 78.75%, respectively, while the oracle achieves 82.74%. This confirms the central distinction motivating our formulation: excluding an unavailable retained client solves the wrong counterfactual, whereas the ghost mechanism preserves most of its contribution.

Table 1: Final test accuracy (%) under retained-client disconnection. For each dataset, we report two levels of client-specific information at the disconnected retained client and consider either an IID forgotten client or one with a 60% unique-label component. The final row is the desired oracle counterfactual, obtained by retraining without the retained-client departure. Our method directly evaluates gradients in **10.9%** of active-client rounds on MNIST and **18.9%** on CIFAR-10 on average (retraining: 100%).

	MNIST				CIFAR-10			
	Departing: 100% Unique		Departing: 60% Unique		Departing: 80% Unique		Departing: 60% Unique	
Method	**IID**	**60%**	**IID**	**60%**	**IID**	**60%**	**IID**	**60%**
Ours	**97.52**	**86.26**	**97.80**	**87.48**	**82.18**	**73.34**	**84.70**	**76.13**
No Ghost	89.49	79.56	89.62	79.50	78.56	70.49	79.05	70.83
TRACE-DU	89.18	79.61	89.57	79.65	78.77	70.71	78.65	70.66
Certified-DFU	96.94	85.22	97.75	86.67	79.87	63.74	81.40	64.40
Retraining w. Departure	89.08	80.30	90.19	80.49	78.75	70.83	79.45	70.90
Oracle: Retraining w/o Departure	*97.76*	*87.73*	*98.41*	*88.00*	*82.74*	*75.45*	*84.95*	*76.47*

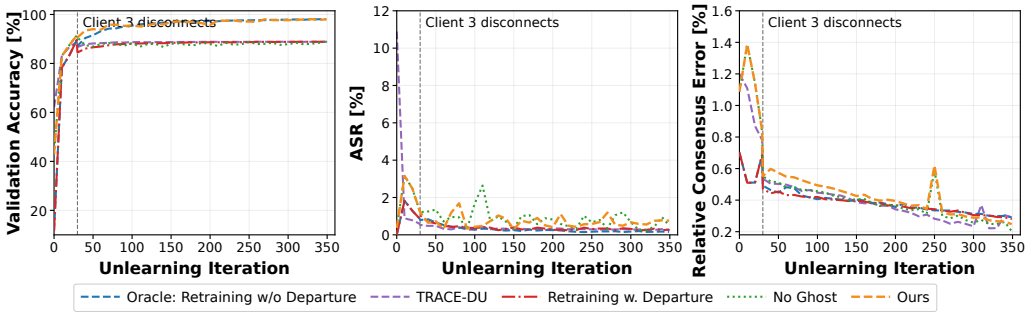

Figure 2: Unlearning dynamics in accuracy (left), ASR (center), and consensus (right) on MNIST with $N = 10$. Retained client has 60% client-specific data and disconnects at $t = 30$ (dashed vertical line).

When the forgotten client itself contains unique information, some utility loss is unavoidable because that information should indeed be removed. Nevertheless, our method remains close to oracle retraining, indicating that most of the remaining degradation is attributable to the requested deletion rather than to retained-client disconnection. Across all main configurations, the pre-unlearning models exhibit a mean ASR of at least 96.83%, while our method reduces the final mean ASR below 0.98% and maintains relative consensus error below 0.29%. Certified-DFU achieves competitive accuracy in some MNIST settings but retains substantially higher ASR in several cases (up to approximately 9.9%); complete accuracy, ASR, and consensus results are provided in the appendix.

Unlearning Dynamics. Figure 2 illustrates a representative MNIST run in which the retained client disconnects at iteration 30. Before disconnection, our method and its no-ghost ablation follow the same trajectory; afterwards, their behavior separates sharply. Ghost reconstruction drives the active clients toward the oracle trajectory, whereas both No Ghost and TRACE-DU approach the retraining-with-departure behavior, reflecting the loss of the unavailable retained client's contribution. Meanwhile, ASR remains low and the clients maintain near-consensus, showing that the utility improvement does not come at the expense of forgetting or decentralized agreement. Additional experiments with different graph connectivities and disconnection times are reported in Appendix E.

7 Conclusions

We studied decentralized machine unlearning when a retained client becomes unavailable during unlearning. We proposed a collaborative reconstruction framework that combines trajectory-based gradient approximation with distributed ghost replicas, allowing the active network to preserve the unavailable client's contribution while tracking the desired counterfactual retraining dynamics. Our theoretical analysis and experiments show that the resulting procedure closely follows the oracle retraining reference while maintaining effective forgetting and substantially reducing gradient evaluations.

REPRODUCIBILITY STATEMENT

The proposed decentralized unlearning framework and ghost-reconstruction mechanism are described in the main paper, with additional algorithmic details provided in Appendices A–B. Theoretical derivations, assumptions, and proofs are provided in Appendices C–D. Detailed experimental protocols, model architectures, training configurations, baseline implementations, hyperparameter settings, and additional robustness experiments are provided in Appendix E. The accompanying code repository contains the implementation and experimental configurations used to produce the reported results.

GENERATIVE AI USE STATEMENT

Generative AI tools, including ChatGPT and Claude, were used for language editing and coding assistance. In particular, these tools assisted with code refinement and documentation, as well as with the implementation of benchmark methods for which no official implementation was publicly available.

All AI-assisted content and code were reviewed, verified, and, where applicable, tested by the authors. The authors take full responsibility for the final content of the paper, the implementation, and the reported results.

REFERENCES

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

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

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

Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. doi: 10.1137/060676386.

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

Jinu Gong, Osvaldo Simeone, and Joonhyuk Kang. Bayesian variational federated learning and unlearning in decentralized networks. In *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 216–220, 2021.

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

Thanh Trung Huynh, Trong Bang Nguyen, Phi Le Nguyen, Thanh Tam Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. Fast-FedUL: A training-free federated unlearning with provable skew resilience. In *Machine Learning and Knowledge Discovery in Databases*, pp. 55–72, 2024.

Yasser H Khalil, Leo Brunswic, Soufiane Lamghari, Xu Li, Mahdi Beitollahi, and Xi Chen. NoT: Federated unlearning via weight negation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25759–25769, 2025.

Hithem Lamri and Michail Maniatakos. Fully decentralized certified unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24577–24586, 2026.

Natalie Lang, Alon Helvits, and Nir Shlezinger. Memory-efficient distributed unlearning. *IEEE Access*, 14:24361–24378, 2026.

Erik G. Larsson and Nicolo Michelusi. Unified analysis of decentralized gradient descent: A contraction mapping framework. *IEEE Open Journal of Signal Processing*, 6:507–529, 2025.

Guofeng Li, Xia Feng, Liangmin Wang, Haiqin Wu, and Boris Düdder. Federated unlearning in the internet of vehicles. In *IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pp. 96–103, 2024.

Yijing Lin, Zhipeng Gao, Hongyang Du, Dusit Niyato, Jiawen Kang, and Xiaoyuan Liu. Incentive and dynamic client selection for federated unlearning. In *Proceedings of the ACM web conference 2024*, pp. 2936–2944, 2024.

Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. FedEraser: Enabling efficient client-level data removal from federated learning models. In *IEEE/ACM International Symposium on Quality of Service (IWQOS)*, 2021.

Ziyao Liu, Yu Jiang, Weifeng Jiang, Jiale Guo, Jun Zhao, and Kwok-Yan Lam. Guaranteeing data privacy in federated unlearning with dynamic user participation. *IEEE Transactions on Dependable and Secure Computing*, 22(3):2072–2085, 2024a.

Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024b.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5.

Pian Qi, Daniela Annunziata, Chiara Jappelli, Fabio Giampaolo, and Francesco Piccialli. FuGuard: Client-level federated unlearning via generative surrogates and optimal transport. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Jing Qiao, Yu Liu, Zengzhe Chen, Mingyi Li, Yuan Yuan, Xiao Zhang, and Dongxiao Yu. PDUDT: Provable decentralized unlearning under dynamic topologies. In *International Conference on Machine Learning*, 2025.

Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11697–11717, 2024.

Trond Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983. doi: 10.1137/0720042.

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.

Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu. Stochastic quasi-newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017. doi: 10.1137/15M1053141.

Hengliang Wu, Youming Tao, Anhao Zhou, Shuzhen Chen, Falko Dressler, and Dongxiao Yu. Certified unlearning in decentralized federated learning. In *IEEE Conference on Computer Communications*, 2026a.

Hengliang Wu, Jiale Yang, Shuzhen Chen, Di Wang, Dongxiao Yu, and Youming Tao. Trajectory-aware certified decentralized unlearning via SGD stability. In *International Conference on Machine Learning*, 2026b.

Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. Heterogeneous decentralised machine unlearning with seed model distillation. *CAAI Transactions on Intelligence Technology*, 9(3):608–619, 2024.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. FedRecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18:4732–4746, 2023. doi: 10.1109/TIFS.2023.3297905.

Xiao Zhang, Zengzhe Chen, Mingyi Li, Jing Qiao, Fuzhen Zhuang, Yuan Yuan, and Dongxiao Yu. Forgetting whenever you want: A decentralized continual learning framework with on-demand unlearning. In *International Conference on Machine Learning*, 2026.

APPENDIX

APPENDIX CONTENTS

A Compact L-BFGS Computation **14**

B Ratio-Consensus Reconstruction of the Missing Mixing Step **14**

C Supporting Theory for Counterfactual Tracking **15**

C.1 Effective Dynamics and Technical Assumptions 15

C.2 Finite-Gossip Reconstruction Error 16

C.3 Contraction of the Retraining Operator 18

C.4 Proof of the Counterfactual Tracking Bound 18

D Additional Analysis **18**

D.1 Distance to the Retained-Client Optimum 18

D.2 Complexity Analysis 19

E Additional Experimental Results **19**

E.1 Experimental Details 19

E.1.1 Datasets, Models, and Training 19

E.1.2 Data Partitioning and Client Roles 20

E.1.3 Backdoor Attacks and Evaluation Metrics 20

E.1.4 Baseline Implementations and Adaptations 20

E.1.5 Hyperparameter Selection and Baseline Tuning 22

E.1.6 Numerical Stability Safeguards 23

E.2 Additional Decentralized Experiments 24

E.2.1 Robustness Across Network and Data Configurations 24

E.2.2 Robustness to Disconnection Dynamics 26

E.2.3 Sensitivity to Method-Inherent Parameters 28

E.2.4 Generalization Across Model Architectures 29

E.3 Centralized Experiments 29

F Limitations and Scope **30**

A Compact L-BFGS Computation

L-BFGS Curvature Buffers. Let $\mathbf{S}_{i,t}$ and $\mathbf{Y}_{i,t}$ denote the L-BFGS buffers available to client i at the beginning of round t. Whenever an exact gradient is evaluated at round ℓ, client i forms the curvature pair

$$\mathbf{s}_{i,\ell} = \bar{\mathbf{x}}_{i,\ell} - \mathbf{x}_{i,\ell}, \qquad \mathbf{y}_{i,\ell} = \nabla f_i(\bar{\mathbf{x}}_{i,\ell}) - \nabla f_i(\mathbf{x}_{i,\ell}). \tag{19}$$

The pair records the displacement from the stored training iterate and the corresponding change in the local gradient. Client i retains the B most recent curvature pairs in $\mathbf{S}_{i,t}$ and $\mathbf{Y}_{i,t}$. These buffers are refreshed following exact-gradient evaluations and remain unchanged during approximation rounds. The initial exact-update stage fills the buffers before the first approximation round. When client r disconnects, its final buffers are made available to its ghost owners and remain frozen thereafter.

Given curvature buffers $\mathbf{S}$ and $\mathbf{Y}$ and a displacement vector $\mathbf{v}$, the compact L-BFGS computation used to approximate the corresponding Hessian–vector product is summarized below.

Algorithm 2 Compact L-BFGS Hessian–Vector Product

Require: $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_B]$, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_B]$, and $\mathbf{v} \in \mathbb{R}^d$
Ensure: Approximated Hessian–vector product $\widehat{\mathbf{h}}$

$\mathbf{A} \leftarrow \mathbf{S}^\top \mathbf{Y}$
$\mathbf{D} \leftarrow \mathrm{diag}(A_{11}, \ldots, A_{BB})$
$\mathbf{L} \leftarrow \mathrm{tril}(\mathbf{A}, -1)$
$\sigma \leftarrow \dfrac{\mathbf{y}_B^\top \mathbf{s}_B}{\mathbf{s}_B^\top \mathbf{s}_B}$
Solve $\begin{bmatrix} -\mathbf{D} & \mathbf{L}^\top \\ \mathbf{L} & \sigma \mathbf{S}^\top \mathbf{S} \end{bmatrix} \mathbf{p} = \begin{bmatrix} \mathbf{Y}^\top \mathbf{v} \\ \sigma \mathbf{S}^\top \mathbf{v} \end{bmatrix}$
$\widehat{\mathbf{h}} \leftarrow \sigma \mathbf{v} - \begin{bmatrix} \mathbf{Y} & \sigma \mathbf{S} \end{bmatrix} \mathbf{p}$
return $\widehat{\mathbf{h}}$

For an active client, Algorithm 2 is applied with $\mathbf{v} = \bar{\mathbf{x}}_{i,t} - \mathbf{x}_{i,t}$. For a ghost replica maintained by client i, it is applied using the frozen buffers of client r and $\mathbf{v} = \bar{\mathbf{x}}_{r,t}^{(i)} - \mathbf{x}_{r,t}$.

B Ratio-Consensus Reconstruction of the Missing Mixing Step

The ratio-consensus procedure in Algorithm 3 reconstructs the missing mixing step that client r would have performed had it remained active, thereby producing the next state of its ghost replicas. The reconstruction is carried out collaboratively over the entire active post-disconnection graph. At initialization, only the former neighbors of r contribute nonzero states, since they hold the terms that originally participated in client r's mixing operation. Specifically, each former neighbor initializes its numerator with its own weighted contribution together with an equal share of the estimated ghost self-contribution, while its denominator is initialized to $1/M_r$. All other active clients initialize both states to zero.

Following this initialization, all active clients participate identically in the K gossip iterations over the post-disconnection graph, repeatedly mixing both the numerator and denominator states with their active neighbors. This allows the initially localized contributions to diffuse throughout the network. Since no single active client has access to all terms of the missing mixing operation, the resulting ratio at each ghost owner combines these distributed contributions into a local reconstruction of the missing mixed state. If a ghost-gradient estimate is unreliable, its local update is frozen according to (11), while the resulting unchanged ghost state still participates in the reconstruction. For finite K, the reconstructed states may differ across ghost owners due to incomplete mixing over the active graph. As K increases, their local ratios approach the target mixed state $\mathbf{m}_{r,t}$ defined in (12).

Algorithm 3 Gossip Reconstruction of the Missing Mixing Step

Require: Round t; active post-disconnection graph $\bar{\mathcal{G}}[\mathcal{R} \setminus \{r\}]$; former-neighbor set $\bar{\mathcal{N}}_r$; original mixing weights $\{\bar{w}_{rj}\}_{j \in \bar{\mathcal{N}}_r \cup \{r\}}$; active-neighbor updates $\{\mathbf{z}_{j,t}\}_{j \in \bar{\mathcal{N}}_r}$; ghost updates $\{\mathbf{z}_{r,t}^{(i)}\}_{i \in \bar{\mathcal{N}}_r}$; gossip budget K

Ensure: Reconstructed ghost states $\{\bar{\mathbf{x}}_{r,t+1}^{(i)}\}_{i \in \bar{\mathcal{N}}_r}$

$M_r \leftarrow |\bar{\mathcal{N}}_r|$
Let $\mathbf{P} = [p_{ij}]$ be the Metropolis matrix associated with $\bar{\mathcal{G}}[\mathcal{R} \setminus \{r\}]$
for all $i \in \mathcal{R} \setminus \{r\}$ **do**
  $\mathbf{a}_i^{(0)} \leftarrow \mathbf{0}, b_i^{(0)} \leftarrow 0$
  if $i \in \bar{\mathcal{N}}_r$ **then**
    $\mathbf{a}_i^{(0)} \leftarrow \bar{w}_{ri}\mathbf{z}_{i,t} + \dfrac{\bar{w}_{rr}}{M_r}\mathbf{z}_{r,t}^{(i)}$
    $b_i^{(0)} \leftarrow \dfrac{1}{M_r}$
for $k = 0, \ldots, K-1$ **do**
  for all $i \in \mathcal{R} \setminus \{r\}$ **do**
    $\mathbf{a}_i^{(k+1)} \leftarrow \sum_{j \in \mathcal{R} \setminus \{r\}} p_{ij}\mathbf{a}_j^{(k)}$
    $b_i^{(k+1)} \leftarrow \sum_{j \in \mathcal{R} \setminus \{r\}} p_{ij} b_j^{(k)}$
for all $i \in \bar{\mathcal{N}}_r$ **do**
  $\bar{\mathbf{x}}_{r,t+1}^{(i)} \leftarrow \dfrac{\mathbf{a}_i^{(K)}}{b_i^{(K)}}$
return $\{\bar{\mathbf{x}}_{r,t+1}^{(i)}\}_{i \in \bar{\mathcal{N}}_r}$

C Supporting Theory for Counterfactual Tracking

C.1 Effective Dynamics and Technical Assumptions

We first make precise the perturbation terms used in Section 5. Define the exact retained-client diffusion operator

$$Q(\mathbf{X}) = \bar{\mathbf{W}}\left(\mathbf{X} - \eta \nabla \mathbf{f}_{\mathcal{R}}(\mathbf{X})\right), \tag{20}$$

such that the counterfactual retraining trajectory obeys

$$\widetilde{\mathbf{X}}_{t+1} = Q(\widetilde{\mathbf{X}}_t). \tag{21}$$

Recall that $\widehat{\mathbf{G}}_t$ denotes the gradients effectively used by the reconstructed dynamics. After client r disconnects, its rth row is the average gradient applied by its ghost replicas; when a ghost replica is frozen by (11), its effective applied gradient is taken to be zero. Accordingly,

$$\mathbf{B}_t = \widehat{\mathbf{G}}_t - \nabla \mathbf{f}_{\mathcal{R}}(\bar{\mathbf{X}}_t). \tag{22}$$

To explicitly characterize the second perturbation, for $t > \tau_r$ define the average locally updated ghost state

$$\bar{\mathbf{z}}_{r,t} = \frac{1}{M_r} \sum_{i \in \bar{\mathcal{N}}_r} \mathbf{z}_{r,t}^{(i)}, \qquad M_r = |\bar{\mathcal{N}}_r|. \tag{23}$$

The mixed state that client r would obtain from these reconstructed local updates is

$$\mathbf{m}_{r,t} = \bar{w}_{rr}\bar{\mathbf{z}}_{r,t} + \sum_{j \in \bar{\mathcal{N}}_r} \bar{w}_{rj}\mathbf{z}_{j,t}. \tag{24}$$

The implemented dynamics differ from $\bar{\mathbf{W}}(\bar{\mathbf{X}}_t - \eta\widehat{\mathbf{G}}_t)$ in two ways. First, the effective rth row is the average of the reconstructed ghost states rather than $\mathbf{m}_{r,t}$. Second, each former neighbor of r mixes using its own local ghost update rather than the average $\bar{\mathbf{z}}_{r,t}$. Consequently, for $t > \tau_r$ we define

$$[\mathbf{C}_t]_{i,:}^{\top} = \begin{cases} \bar{\mathbf{x}}_{r,t+1}^{\mathrm{gh}} - \mathbf{m}_{r,t}, & i = r, \\ \bar{w}_{ir}\big(\mathbf{z}_{r,t}^{(i)} - \bar{\mathbf{z}}_{r,t}\big), & i \in \bar{\mathcal{N}}_r, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{25}$$

and $\mathbf{C}_t = \mathbf{0}$ for $t \le \tau_r$. The reconstructed dynamics therefore satisfy

$$\bar{\mathbf{X}}_{t+1} = Q(\bar{\mathbf{X}}_t) - \eta \bar{\mathbf{W}} \mathbf{B}_t + \mathbf{C}_t. \tag{26}$$

Technical Assumptions. We use the following conditions throughout the analysis.

Assumption 1 (Mixing)**.** *The retained-client matrix $\bar{\mathbf{W}}$ is symmetric and doubly stochastic, has positive diagonal entries, and is supported on a connected undirected graph.*

Assumption 2 (Objective regularity)**.** *Every retained objective f_i is differentiable, μ-strongly convex, and L-smooth, with common constants $0 < \mu \le L$.*

Assumption 3 (Gradient reconstruction)**.** *For every $t = 0, \dots, T-1$, the realized gradient-reconstruction error satisfies*

$$\|\mathbf{B}_t\|_F \le \beta_t \le \beta < \infty. \tag{27}$$

The first two conditions are standard in convergence analyses of decentralized gradient methods Yuan et al. (2016). Assumption 3 is the usual bounded-error abstraction for inexact first-order methods d'Aspremont (2008); it does not require the Taylor–L-BFGS approximation to be uniformly accurate. An explicit bound on the second reconstruction error $\mathbf{C}_t$ is derived next.

C.2 Finite-Gossip Reconstruction Error

Let $n_{\mathrm{a}} = |\mathcal{R} \setminus \{r\}| = N - 2$ and $\mathbf{J} = \frac{1}{n_{\mathrm{a}}} \mathbf{1}\mathbf{1}^\top$, and recall that $\mathbf{P}$ is the Metropolis matrix used by the auxiliary post-disconnection gossip procedure. Define

$$\rho_{\mathbf{P}} = \|\mathbf{P} - \mathbf{J}\|_2 < 1. \tag{28}$$

At round $t > \tau_r$, collect the ratio-consensus initializations into $\mathbf{A}_t^{(0)} \in \mathbb{R}^{n_{\mathrm{a}} \times d}$ and $\mathbf{b}_t^{(0)} \in \mathbb{R}^{n_{\mathrm{a}}}$, whose ith entries are

$$\mathbf{a}_{i,t}^{(0)} = \mathbb{I}_{\{i \in \bar{\mathcal{N}}_r\}} \left(\bar{w}_{ri} \mathbf{z}_{i,t} + \frac{\bar{w}_{rr}}{M_r} \mathbf{z}_{r,t}^{(i)} \right), \qquad b_{i,t}^{(0)} = \frac{1}{M_r} \mathbb{I}_{\{i \in \bar{\mathcal{N}}_r\}}. \tag{29}$$

By construction,

$$\mathbf{1}^\top \mathbf{A}_t^{(0)} = \mathbf{m}_{r,t}^\top, \qquad \mathbf{1}^\top \mathbf{b}_t^{(0)} = 1. \tag{30}$$

After K gossip iterations, we have that $\mathbf{A}_t^{(K)} = \mathbf{P}^K \mathbf{A}_t^{(0)}$ and $\mathbf{b}_t^{(K)} = \mathbf{P}^K \mathbf{b}_t^{(0)}$. Define

$$\xi_t = \left\| \mathbf{A}_t^{(0)} - \mathbf{b}_t^{(0)} \mathbf{m}_{r,t}^\top \right\|_F, \qquad \delta_b = \left\| \mathbf{b}_t^{(0)} - \frac{1}{n_{\mathrm{a}}} \mathbf{1} \right\|_2 = \sqrt{\frac{1}{M_r} - \frac{1}{n_{\mathrm{a}}}}. \tag{31}$$

Lemma 1 (Finite-gossip ghost reconstruction)**.** *Suppose that*

$$\underline{b}_K \triangleq \frac{1}{n_{\mathrm{a}}} - \rho_{\mathbf{P}}^K \delta_b > 0. \tag{32}$$

Then every ghost owner $i \in \bar{\mathcal{N}}_r$ satisfies

$$\left\| \frac{\mathbf{a}_{i,t}^{(K)}}{b_{i,t}^{(K)}} - \mathbf{m}_{r,t} \right\|_2 \le \varepsilon_{K,t}, \qquad \varepsilon_{K,t} \triangleq \frac{\rho_{\mathbf{P}}^K \xi_t}{\underline{b}_K}. \tag{33}$$

Moreover, letting

$$\Delta_{r,t} = \left(\sum_{i \in \bar{\mathcal{N}}_r} \|\mathbf{z}_{r,t}^{(i)} - \bar{\mathbf{z}}_{r,t}\|_2^2 \right)^{1/2}, \qquad \omega_r = \max_{i \in \bar{\mathcal{N}}_r} \bar{w}_{ir}, \tag{34}$$

the perturbation in (25) obeys

$$\|\mathbf{C}_t\|_F \le \gamma_t(K) \triangleq \sqrt{\varepsilon_{K,t}^2 + \omega_r^2 \Delta_{r,t}^2}. \tag{35}$$

Proof. Since (30) holds, $\mathbf{J}\left(\mathbf{A}_t^{(0)} - \mathbf{b}_t^{(0)}\mathbf{m}_{r,t}^\top\right) = \mathbf{0}$. Hence

$$\mathbf{A}_t^{(K)} - \mathbf{b}_t^{(K)}\mathbf{m}_{r,t}^\top = (\mathbf{P}^K - \mathbf{J})\left(\mathbf{A}_t^{(0)} - \mathbf{b}_t^{(0)}\mathbf{m}_{r,t}^\top\right). \tag{36}$$

For every row i,

$$\left\|\mathbf{a}_{i,t}^{(K)} - b_{i,t}^{(K)}\mathbf{m}_{r,t}\right\|_2 \le \rho_{\mathbf{P}}^K \xi_t. \tag{37}$$

Similarly, $\mathbf{b}_t^{(K)} - \frac{1}{n_{\mathrm{a}}}\mathbf{1} = (\mathbf{P}^K - \mathbf{J})\left(\mathbf{b}_t^{(0)} - \frac{1}{n_{\mathrm{a}}}\mathbf{1}\right)$, and therefore

$$b_{i,t}^{(K)} \ge \frac{1}{n_{\mathrm{a}}} - \rho_{\mathbf{P}}^K \delta_b = \underline{b}_K. \tag{38}$$

Combining (37) and (38) proves (33).

Since the effective ghost row is the average of the ghost-owner outputs, convexity of the norm gives

$$\|\bar{\mathbf{x}}_{r,t+1}^{\mathrm{gh}} - \mathbf{m}_{r,t}\|_2 \le \varepsilon_{K,t}. \tag{39}$$

From (25),

$$\|\mathbf{C}_t\|_F^2 = \|\bar{\mathbf{x}}_{r,t+1}^{\mathrm{gh}} - \mathbf{m}_{r,t}\|_2^2 + \sum_{i\in\mathcal{N}_r} \bar{w}_{ir}^2 \|\mathbf{z}_{r,t}^{(i)} - \bar{\mathbf{z}}_{r,t}\|_2^2 \le \varepsilon_{K,t}^2 + \omega_r^2 \Delta_{r,t}^2, \tag{40}$$

which proves (35). □

Lemma 1 holds without requiring the individual ghost replicas to make identical validity decisions. It shows explicitly that the finite-consensus component of γ_t decreases geometrically with the gossip budget K through $\rho_{\mathbf{P}}^K$. A stronger statement is available when the local ghost-update map is common across replica owners.

Corollary 2 (Geometric dependence on the gossip budget)**.** *Suppose that at round t the ghost local-update rule can be represented by a common map Φ_t satisfying $\|\Phi_t(\mathbf{x}) - \Phi_t(\mathbf{y})\|_2 \le q_t\|\mathbf{x}-\mathbf{y}\|_2$. Then, for $t > \tau_r + 1$,*

$$\Delta_{r,t} \le q_t \sqrt{M_r}\, \varepsilon_{K,t-1}. \tag{41}$$

Consequently,

$$\gamma_t(K) \le \sqrt{\varepsilon_{K,t}^2 + \omega_r^2 M_r q_t^2 \varepsilon_{K,t-1}^2}. \tag{42}$$

If $\xi_t \le \Xi$ and $q_t \le q$ uniformly over the analyzed horizon and independently of K, then, for all sufficiently large K satisfying (32), it holds that $\gamma_t(K) = O(\rho_{\mathbf{P}}^K)$.

Proof. Each ghost replica at round t is the output of the previous ratio-consensus procedure. By Lemma 1, $\|\bar{\mathbf{x}}_{r,t}^{(i)} - \mathbf{m}_{r,t-1}\|_2 \le \varepsilon_{K,t-1}$. Hence

$$\begin{aligned}\sum_{i\in\mathcal{N}_r} \|\bar{\mathbf{x}}_{r,t}^{(i)} - \bar{\mathbf{x}}_{r,t}^{\mathrm{gh}}\|_2^2 &\le \sum_{i\in\mathcal{N}_r} \|\bar{\mathbf{x}}_{r,t}^{(i)} - \mathbf{m}_{r,t-1}\|_2^2 \\ &\le M_r \varepsilon_{K,t-1}^2. \end{aligned} \tag{43}$$

Applying the q_t-Lipschitz property of Φ_t gives (41). Substitution into (35) proves (42). The final statement follows directly from (33). □

For the accepted Taylor–L-BFGS branch, the ghost update is affine and one may take $q_t = \left\|\mathbf{I} - \eta\widehat{\mathbf{H}}_{r,t}\right\|_2$, where $\widehat{\mathbf{H}}_{r,t}$ is the common L-BFGS Hessian approximation. A unanimously frozen round corresponds to $q_t = 1$. The more general bound in Lemma 1 remains applicable when different ghost owners make different validity decisions.

C.3 Contraction of the Retraining Operator

Lemma 2 (Contraction). *Under Assumptions 1–2, let $0 < \eta \le 2/(L+\mu)$ and $c = 1 - \eta\mu$. Then*

$$\|Q(\mathbf{X}) - Q(\mathbf{Y})\|_F \le c\|\mathbf{X} - \mathbf{Y}\|_F. \tag{44}$$

Proof. The gradient mapping $\mathbf{X} \mapsto \mathbf{X} - \eta\nabla\mathbf{f}_{\mathcal{R}}(\mathbf{X})$ is c-contractive under strong convexity and smoothness for $0 < \eta \le 2/(L+\mu)$. Specifically, by the Mean Hessian Theorem Larsson & Michelusi (2025), for each retained objective and any $\mathbf{x}, \mathbf{y}$, there exists $\mu\mathbf{I} \preceq \mathbf{A} \preceq L\mathbf{I}$ such that the corresponding gradient difference equals $\mathbf{A}(\mathbf{x} - \mathbf{y})$. Hence, for $0 < \eta \le 2/(L+\mu)$, the row-wise gradient map is $(1 - \eta\mu)$-contractive. Moreover, $\|\bar{\mathbf{W}}\|_2 = 1$ by Assumption 1. Therefore,

$$\begin{aligned}\|Q(\mathbf{X}) - Q(\mathbf{Y})\|_F &\le \|(\mathbf{X} - \eta\nabla\mathbf{f}_{\mathcal{R}}(\mathbf{X})) - (\mathbf{Y} - \eta\nabla\mathbf{f}_{\mathcal{R}}(\mathbf{Y}))\|_F \\ &\le c\|\mathbf{X} - \mathbf{Y}\|_F.\end{aligned} \tag{45}$$

□

Corollary 3. *The operator Q admits a unique fixed point $\widehat{\mathbf{X}} = Q(\widehat{\mathbf{X}})$.*

Proof. Lemma 2 and completeness of $\mathbb{R}^{(N-1)\times d}$ allow direct application of the Banach fixed-point theorem. □

C.4 Proof of the Counterfactual Tracking Bound

Proof of Theorem 1. Using (21) and (26),

$$\bar{\mathbf{X}}_{t+1} - \widetilde{\mathbf{X}}_{t+1} = Q(\bar{\mathbf{X}}_t) - Q(\widetilde{\mathbf{X}}_t) - \eta\bar{\mathbf{W}}\mathbf{B}_t + \mathbf{C}_t. \tag{46}$$

Therefore,

$$\|\bar{\mathbf{X}}_{t+1} - \widetilde{\mathbf{X}}_{t+1}\|_F \le c\|\bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t\|_F + \eta\beta_t + \gamma_t, \tag{47}$$

where we used Lemma 2, $\|\bar{\mathbf{W}}\|_2 = 1$, and the definitions of β_t and γ_t.

Iterating (47) gives

$$\|\bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t\|_F \le c^t\|\bar{\mathbf{X}}_0 - \widetilde{\mathbf{X}}_0\|_F + \eta\sum_{k=0}^{t-1} c^{t-1-k}\beta_k + \sum_{k=0}^{t-1} c^{t-1-k}\gamma_k. \tag{48}$$

Since $\bar{\mathbf{X}}_0 = \widetilde{\mathbf{X}}_0$ and $\gamma_k = 0$ for $k \le \tau_r$, this reduces to (16). □

If $\beta_t \le \beta$ and $\gamma_t \le \gamma$, geometric summation yields $\eta\sum_{k=0}^{t-1} c^{t-1-k}\beta_k \le \frac{\beta}{\mu}(1 - c^t)$ and $\sum_{k=\tau_r+1}^{t-1} c^{t-1-k}\gamma_k \le \frac{\gamma}{\eta\mu}\left(1 - c^{\max\{t-\tau_r-1,0\}}\right)$, which proves (17).

D Additional Analysis

D.1 Distance to the Retained-Client Optimum

Let $\mathbf{x}^\star = \arg\min_{\mathbf{x}} F_{N-1}(\mathbf{x})$, where $\mathbf{X}^\star = \mathbf{1}(\mathbf{x}^\star)^\top$, and let $\widehat{\mathbf{X}}$ denote the unique fixed point of Q established in Corollary 3. Define

$$\Lambda_{\bar{\mathbf{W}}} = 2\left\|(\mathbf{I} - \bar{\mathbf{W}})^\dagger\bar{\mathbf{W}}\right\|_2. \tag{49}$$

A standard constant-step decentralized-gradient result specialized to our matrix-valued dynamics gives

$$\|\widehat{\mathbf{X}} - \mathbf{X}^\star\|_F \le \eta\frac{L}{\mu}\Lambda_{\bar{\mathbf{W}}}\|\nabla\mathbf{f}_{\mathcal{R}}(\mathbf{X}^\star)\|_F, \tag{50}$$

provided, when $\Lambda_{\bar{\mathbf{W}}} > 0$,

$$\eta \le \frac{1}{L\Lambda_{\bar{\mathbf{W}}}}. \tag{51}$$

This follows directly from the vectorized diffusion analysis of Larsson & Michelusi (2025).

Theorem 4 (Distance to the retained-client optimum)**.** *Under the conditions of Theorem 1 and (51),*

$$\|\bar{\mathbf{X}}_t - \mathbf{X}^\star\|_F \le \eta \sum_{k=0}^{t-1} c^{t-1-k}\beta_k + \sum_{k=\tau_r+1}^{t-1} c^{t-1-k}\gamma_k + c^t\|\widetilde{\mathbf{X}}_0 - \widehat{\mathbf{X}}\|_F + \eta\frac{L}{\mu}\Lambda_{\bar{\mathbf{W}}}\|\nabla \mathbf{f}_{\mathcal{R}}(\mathbf{X}^\star)\|_F. \tag{52}$$

Proof. By the triangle inequality, $\|\bar{\mathbf{X}}_t - \mathbf{X}^\star\|_F \le \|\bar{\mathbf{X}}_t - \widetilde{\mathbf{X}}_t\|_F + \|\widetilde{\mathbf{X}}_t - \mathbf{X}^\star\|_F$. The first term follows from Theorem 1. For the second, Lemma 2 yields

$$\|\widetilde{\mathbf{X}}_t - \widehat{\mathbf{X}}\|_F \le c^t\|\widetilde{\mathbf{X}}_0 - \widehat{\mathbf{X}}\|_F,$$

which combined with (50) proves the claim. □

The first two terms in (52) are due to the proposed reconstruction mechanism, whereas the last two are intrinsic to constant-step decentralized retraining. In particular, the final term is a persistent topology- and heterogeneity-dependent diffusion bias; it vanishes when the retained local objectives share a common minimizer.

D.2 Complexity Analysis

Let d be the number of model parameters, $m = \max_{i\in\mathcal{R}} |\mathcal{D}_i|$, and B the L-BFGS memory size. A full local-gradient evaluation costs $O(md)$, whereas one compact L-BFGS reconstruction costs $O(B^2d + B^3)$. Full retraining therefore costs $O(T|\mathcal{R}|md)$.

Let M_{ex}, M_{ap}, and M_{fb} denote the total numbers of exact evaluations, accepted approximations, and fallback evaluations. The computational complexity of the proposed procedure is

$$O\left((M_{\mathrm{ex}} + M_{\mathrm{fb}})md + \left[M_{\mathrm{ap}} + M_{\mathrm{fb}} + (T - \tau_r - 1)|\bar{\mathcal{N}}_r|\right](B^2d + B^3)\right). \tag{53}$$

Hence, the method reduces computation whenever trajectory reconstruction is substantially cheaper than direct gradient evaluation and exact/fallback updates are sufficiently infrequent.

Communication and Storage. Relative to ordinary decentralized training, retaining the gradient together with each transmitted model incurs $O(T|\mathcal{E}|d)$ additional transmitted scalars without adding communication rounds. Transfer of the L-BFGS memory of client r at disconnection costs $O(|\bar{\mathcal{N}}_r|Bd)$. If $\mathcal{E}^{\mathrm{act}}$ denotes the post-disconnection edge set, the K auxiliary gossip iterations incur $O\left((T - \tau_r - 1)K|\mathcal{E}^{\mathrm{act}}|d\right)$ additional transmitted scalars. Finally, client i requires $O(Td(|\mathcal{N}_i| + 1) + Bd)$ storage for its trajectory history and L-BFGS buffers.

E Additional Experimental Results

This section complements the main experimental study with additional implementation details and evaluations across alternative network topologies, datasets, model architectures, and coordination settings. We first provide further details on the experimental setup, benchmark adaptations, and hyperparameter selection. We then report additional decentralized experiments varying the number of clients, communication-graph connectivity, retained-client heterogeneity, the number and timing of client disconnections, and model architecture. Finally, we consider the standard centralized federated learning setting, in which an orchestrating server coordinates the participating clients. Finally, we consider the standard centralized federated learning setting, in which an orchestrating server coordinates the participating clients.

E.1 Experimental Details

E.1.1 Datasets, Models, and Training

We evaluate the proposed method on MNIST, Fashion-MNIST, and CIFAR-10 using a CNN trained with SGD, unless stated otherwise. MNIST and CIFAR-10 are evaluated in the decentralized setting, where

clients communicate over Erdős–Rényi (ER) graphs with connection probability $p = 0.5$ in the main experiments. For MNIST, we consider $N \in 10, 20$ clients, while CIFAR-10 uses $N = 10$. Fashion-MNIST is evaluated in the centralized federated setting with $N = 10$ clients and an orchestrating server.

The main paper reports the MNIST results for $N = 20$ and the CIFAR-10 results for $N = 10$; here, we examine additional configurations and robustness studies described in the sequel.

E.1.2 Data Partitioning and Client Roles

In the standard experimental setting, we consider two distinguished clients: the client designated for unlearning and a different retained client that disconnects during unlearning. In the experiments, the retained client disconnects within the first 20% of the unlearning iterations. Appendix E.2.2 separately examines a broader range of disconnection times.

To isolate the respective contributions of the above two clients to learning and unlearning performance, we consider a challenging non-IID scenario. For either client, a prescribed fraction of its samples has a unique label unavailable to all other clients, while the remaining samples are drawn IID from the remaining labels.

The above non-IID construction separately controls the information to be removed and retained. A larger unique-label fraction for the retained client makes its missing contribution harder to reconstruct. For the forgotten client, the presence of a unique label means that successful unlearning is expected to remove the knowledge contributed for that label, leading to reduced accuracy on that label and, consequently, overall accuracy. This drop should therefore not be interpreted by itself as poorer unlearning performance. Accuracy is thus interpreted relative to the corresponding retraining reference and jointly with ASR, defined below. Appendix E.2.1 further increases retained-client heterogeneity by assigning two unique labels to the retained client.

E.1.3 Backdoor Attacks and Evaluation Metrics

The client designated for unlearning serves as the malicious client during the original training procedure. We select 50% of its local training samples uniformly at random, modify each selected sample by adding a fixed backdoor trigger, and relabel the resulting poisoned sample to a fixed target class. At evaluation, the same trigger is applied only to test samples whose ground-truth label differs from the attack target, and ASR is measured as the fraction of these triggered samples classified as the target class.

For each experimental configuration, we report results averaged over five independent runs. Final performance is evaluated over the clients that remain available at the end of unlearning, referred to as the active clients. Within each run, test accuracy is averaged over the active clients and then across the five independent runs. We also evaluate ASR over the active clients and, in decentralized settings, the relative consensus error.

Across all evaluated configurations, the pre-unlearning models exhibit a mean ASR of at least 96.83%, confirming that the backdoor is strongly embedded before unlearning. For the configurations summarized in the tables, the proposed method reduces the final mean ASR to at most 0.98% and, in the decentralized settings, maintains a final mean relative consensus error of at most 0.29%. Unless stated otherwise, the compared methods attain comparable final ASR and consensus-error values, with any deviations noted explicitly. The tables therefore report only final test accuracy.

E.1.4 Baseline Implementations and Adaptations

We compare the proposed method with an ablation, retraining references, and decentralized unlearning benchmarks. To provide a centralized reference under the simpler assumption of server-based coordination, we additionally compare against recent federated unlearning methods.

No Ghost. No Ghost follows the same trajectory-reconstruction procedure as Ours while the retained client is active, but performs no ghost reconstruction after client r disconnects, isolating the contribution of the proposed mechanism.

Retraining references. Retraining with Departure retrains under the same disconnection event; after client r leaves, training continues only with the available clients and therefore excludes the contributions of both the client designated for unlearning and the disconnected retained client.

Retraining without Departure instead retrains over the complete retained-client set, with client r remaining available throughout, and serves as the desired oracle counterfactual in which the requested contribution is removed while all retained-client contributions are preserved.

TRACE-DU. We evaluate TRACE-DU Wu et al. (2026b). Since no official implementation is publicly available, we implemented the method based on the algorithmic description and pseudocode provided in the original paper and adapted it to our experimental environment. To incorporate the retained-client departure into our setting, client r is removed at the prescribed round and the communication graph and mixing weights are updated accordingly. The original unlearning mechanism is otherwise preserved, and no additional mechanism is introduced to reconstruct the disconnected client's contribution. We use the same client partitions, model architecture, local training configuration, communication topology, and mixing rule as in the corresponding experiment. The unlearning point is matched to that used by our method, after the model has reached high validation accuracy while maintaining a high ASR. TRACE-DU is given the same number of post-unlearning training rounds as the number of unlearning iterations used by our method, providing a matched iterative computation budget after the unlearning request.

Certified-DFU. We evaluate Certified-DFU Wu et al. (2026a). Since no official implementation is publicly available, we implemented the method based on the algorithmic description and pseudocode provided in the original paper and adapted it to our data partition, model, and evaluation pipeline. Certified-DFU applies a Newton-style corrective update using information from the client designated for unlearning together with curvature information from the retained clients, followed by calibrated perturbation and decentralized propagation. This differs from our setting in its information-access assumptions: Certified-DFU relies on information provided by the client designated for unlearning, whereas our method performs unlearning without further access to that client. In addition, its main unlearning operation consists of a single corrective step, so we evaluate it without retained-client disconnection during unlearning. Following the regularized formulation in the original paper, we include an ℓ_2 regularization term in the training objective and use the proposed empirical Fisher approximation in place of the full Hessian for our CNN implementation.

FedRecover. In the centralized experiments, we evaluate FedRecover Cao et al. (2023). Since no official implementation used in our experiments is publicly available, we implemented the method based on the algorithmic description and pseudocode in the original paper and adapted it to our centralized experimental environment. FedRecover serves as a trajectory-reconstruction baseline without ghost reconstruction: after the retained client disconnects, its subsequent contribution is no longer represented. We provide FedRecover with the historical model and optimization information required by its trajectory-reconstruction procedure and use the same client partition, model architecture, local training configuration, and unlearning request as in the corresponding centralized experiment. When client r disconnects, it is removed from subsequent training and reconstruction steps, with no surrogate state introduced to approximate its future contribution.

Fast-FedUL. For Fast-FedUL Huynh et al. (2024), we use the authors' official implementation as the basis for our experiments and preserve the original unlearning algorithm. We adapt the implementation only as needed to integrate it into our experimental environment, including the common data partition, model interface, and evaluation protocol. Specifically, we replace the original data-loading and model interfaces with our common client partitions and model architecture, align the training and evaluation pipeline with the remaining baselines, and report the same utility and forgetting metrics. No modification is made to the Fast-FedUL unlearning procedure itself.

Fast-FedUL is a training-free method and does not perform an iterative unlearning or recovery trajectory. Consequently, there is no departure round during unlearning and no unlearning-iteration budget to match. Instead, we evaluate the final unlearned model produced by Fast-FedUL from the same pre-unlearning model and experimental configuration used by the other methods.

Across methods, we match the data partition, client roles, attack setting and, when applicable, the communication topology. When a benchmark requires a different training procedure or information-access

assumption, we follow its prescribed setup while keeping the remaining experimental conditions matched as closely as possible.

E.1.5 HYPERPARAMETER SELECTION AND BASELINE TUNING

Hyperparameters are selected separately for each experimental configuration, as the reported experiments span different datasets, client data partitions, model architectures, and retained-client disconnection settings. For each configuration, candidate hyperparameter settings are evaluated jointly according to model utility, forgetting performance, and decentralized agreement, favoring high validation accuracy, low attack success rate (ASR), and low relative consensus error. Once a hyperparameter configuration is selected for a given experimental setting, it is fixed and used unchanged across all five independent runs reported for that setting; no per-run hyperparameter retuning is performed. We therefore do not enforce a single global configuration across all experiments. Since the selected values vary across configurations, Tables 2 and 3 report the search ranges and fixed values considered during tuning rather than a separate final hyperparameter configuration for every individual experiment.

We distinguish between shared optimization hyperparameters and method-specific unlearning hyperparameters. Shared optimization parameters are matched across methods within each experimental configuration whenever applicable. When an iterative unlearning method continues to perform gradient-based optimization, the corresponding optimization parameters are retained during unlearning unless the method explicitly requires a different procedure. In addition, all iterative unlearning methods are given the same number of unlearning iterations as our method in the corresponding configuration. The shared optimization ranges and fixed values are summarized in Table 2.

Table 2: Search ranges and fixed values for the shared optimization and training hyperparameters considered across the reported experiments.

Hyperparameter	Search range / fixed value
Learning rate	$\{0.03,\ 0.04,\ 0.05\}$; CIFAR-10 used predefined architecture-specific learning-rate schedules
Batch size	128
Local epochs	1
Optimizer	SGD; method-specific regularization applied when prescribed
Pre-unlearning training iterations	$\{350,\ 500,\ 800,\ 1000\}$
Unlearning iterations (iterative methods)	Matched to Ours within each configuration

For CIFAR-10, we use architecture-specific piecewise-constant learning-rate schedules. For ResNet-20, the learning rate is initialized at 0.04 and reduced to 0.02, 0.01, and 0.005 at iterations 401, 701, and 901, respectively. For the CNN, it is initialized at 0.05 and reduced to 0.03, 0.01, 0.005, and 0.001 at iterations 301, 501, 601, and 701, respectively.

The method-specific hyperparameter search spaces are summarized in Table 3. For Ours and FedRecover, the shared reconstruction and correction parameters include the number of initial exact rounds, the periodic exact-correction interval, the number of final exact rounds, the L-BFGS memory size, and the infinity-norm threshold scale used to determine when a reconstructed update is considered unreliable. Our method additionally tunes the damping coefficient and the number of ratio-consensus iterations used to reconstruct the missing mixing step of the disconnected retained client.

For TRACE-DU, the reported ϵ and δ values specify fixed certification parameters rather than ordinary optimization hyperparameters. For the remaining baselines, we tune the parameters specific to their respective unlearning procedures.

Whenever values recommended by the original paper or implementation are directly applicable, they are included among the candidate configurations. Additional values are considered when adaptation to our data partition, model architecture, or optimization setting is required.

Table 3: Search ranges and fixed values for method-specific hyperparameters. When multiple values are listed, tuning is performed separately for each experimental configuration.

Method	Hyperparameter	Search range / fixed value
Ours	Initial exact rounds T_{init}	$\{5, 6\}$
	Correction period T_{corr}	$\{10, 15, 20\}$
	Final exact rounds T_{final}	$\{0, 4, 6\}$
	L-BFGS memory	$\{3, 4\}$
	Infinity-norm threshold scale γ_∞	$[0.5, 1.2]$
	Ratio-consensus iterations K	$\{2, 5, 10, 15\}$
	Damping coefficient	$[0.1, 0.5]$
FedRecover	Initial exact rounds T_{init}	$\{5, 6\}$
	Correction period T_{corr}	$\{10, 15, 20\}$
	Final exact rounds T_{final}	$\{0, 4, 6\}$
	L-BFGS memory	$\{3, 4\}$
	Infinity-norm threshold scale γ_∞	$[0.5, 1.2]$
TRACE-DU	Certification parameter ϵ	50
	Certification parameter δ	10^{-5}
	Noise parameter σ	$[0.025, 0.200]$
Certified-DFU	ℓ_2-regularization coefficient λ	$[10^{-4}, 10^{-1}]$
	CG relative tolerance	$[10^{-4}, 10^{-3}]$
	CG maximum iterations	$\{400, 800\}$
Fast-FedUL	Expected saving / sampling size m	$\{5, 7, 10\}$
	Skew-mitigation coefficient α	$[0, 0.07]$
	Correction scale θ_Δ	$[0.01, 2]$

This tuning procedure is intended to evaluate each method under a competitive configuration while keeping the experimental conditions matched within each individual setting.

E.1.6 Numerical Stability Safeguards

We employ two complementary numerical safeguards.

L-BFGS curvature damping. We use damped L-BFGS to safeguard the stored curvature pairs by modifying curvature information when the curvature condition is insufficiently satisfied Nocedal & Wright (2006). Such damping is commonly used to improve the robustness of quasi-Newton approximations in stochastic and nonconvex optimization Wang et al. (2017).

Taylor-correction scaling. After client r disconnects, we additionally bound the second-order Taylor correction used by each ghost replica, following the general principle of constraining approximate second-order steps for improved robustness Steihaug (1983). Let

$$\widehat{\mathbf{h}}_{r,t}^{(i)} = \mathbf{H}_{r,t}\left(\bar{\mathbf{x}}_{r,t}^{(i)} - \mathbf{x}_{r,t}\right). \tag{54}$$

We use the same historical-scale threshold Γ_r employed by the ghost-gradient validity test and define

$$\kappa_{r,t}^{(i)} = \min\left\{1, \frac{\Gamma_r}{\|\widehat{\mathbf{h}}_{r,t}^{(i)}\|_\infty + \varepsilon_{\text{num}}}\right\}, \tag{55}$$

where $\varepsilon_{\text{num}} = 10^{-12}$ is used only for numerical stability. The resulting ghost-gradient estimate is

$$\widehat{\mathbf{g}}_{r,t}^{(i)} = \nabla f_r(\mathbf{x}_{r,t}) + \kappa_{r,t}^{(i)}\widehat{\mathbf{h}}_{r,t}^{(i)}. \tag{56}$$

Thus, corrections below the prescribed threshold are unchanged, while larger corrections are proportionally attenuated.

E.2 Additional Decentralized Experiments

The main paper reports the primary MNIST and CIFAR-10 results and illustrates the unlearning dynamics for a representative retained-client disconnection. Here, we extend the evaluation to additional data, network, disconnection, and model settings.

E.2.1 Robustness Across Network and Data Configurations

We first examine whether the behavior observed in the main MNIST experiment persists under variations in the data distribution and decentralized network configuration. We vary the heterogeneity of the client designated for unlearning and the retained client that disconnects, the number of participating clients, and the connectivity of the communication graph. Together, these experiments test robustness to changes in both client-specific information and network structure.

Forgotten-client heterogeneity. We further evaluate the main MNIST setting when the client designated for unlearning contains an 80% unique-label fraction. Table 4 reports the corresponding final test accuracy for the two departing-client heterogeneity settings considered in the main experiment.

Table 4: Additional final test accuracy (%) for MNIST with $N = 20$ clients over an ER graph ($p = 0.5$) when the forgotten client has an 80% unique-label component.

	Accuracy (%)	
	Departing: 100% Unique	**Departing: 60% Unique**
Ours	**86.08**	**87.51**
No Ghost	79.55	79.56
TRACE-DU	79.51	79.67
Certified-DFU	85.56	86.64
Retraining w. Departure	80.54	78.98
Oracle: Retraining w/o Departure	*87.73*	*87.48*

The additional results follow the same trend as the main MNIST experiment. When the forgotten client has an 80% unique-label component, Ours achieves 86.08% and 87.51% test accuracy for departing-client unique-label fractions of 100% and 60%, respectively. These results remain close to the corresponding oracle retraining accuracies of 87.73% and 87.48%. In contrast, No Ghost, TRACE-DU, and Retraining w. Departure exhibit larger accuracy drops. Certified-DFU performs better than these baselines, but still remains below Ours in both configurations.

Network size. We next examine whether the behavior observed in the main MNIST experiment persists for a smaller decentralized network. Table 5 reports the final test accuracy for $N = 10$ clients under two levels of client-specific information at the disconnected retained client and three data configurations for the forgotten client.

Table 5: Final test accuracy (%) for MNIST with $N = 10$ clients over an ER graph ($p = 0.5$). We consider two levels of client-specific information at the disconnected retained client. Within each group, the forgotten client is either IID or contains a 60% or 80% unique-label component.

	Accuracy (%)					
	Departing: 100% Unique			**Departing: 60% Unique**		
Method	**IID**	**60%**	**80%**	**IID**	**60%**	**80%**
Ours	**97.01**	**85.11**	**85.72**	**97.83**	**86.75**	**88.10**
No Ghost	88.69	79.40	79.43	89.46	79.49	79.58
TRACE-DU	89.52	79.56	79.65	89.76	80.34	80.34
Certified-DFU	95.54	79.17	83.55	95.27	84.60	84.52
Retraining w. Departure	88.80	79.70	79.78	89.49	80.86	80.84
Oracle: Retraining w/o Departure	*97.99*	*85.84*	*86.90*	*98.44*	*88.16*	*88.25*

The $N = 10$ results exhibit the same qualitative behavior as the larger MNIST network reported in the main paper. Across all considered unique-label fractions, retained-client departure consistently reduces final test accuracy, whereas the proposed ghost reconstruction recovers most of this degradation and remains close to retraining without client departure.

For example, with an IID client to be forgotten and a retained client consisting entirely of a unique label, Ours achieves 97.01% accuracy, compared with 88.69% for No Ghost and 97.99% for the oracle retraining reference. The same pattern holds when the forgotten client contains a substantial unique-label component.

TRACE-DU remains close to the methods that continue after disconnection without reconstructing the missing retained client, since in our adaptation client r is simply removed upon disconnection, with no mechanism to reconstruct its missing contribution. Certified-DFU attains relatively high test accuracy in several settings. However, its residual ASR remains non-negligible in several unique-label configurations, reaching approximately 2.6%. Its accuracy should therefore be interpreted jointly with this residual backdoor effect and its different information-access assumptions.

Communication-graph connectivity. We next examine the sensitivity of the proposed method to reduced communication-graph connectivity. While the remaining MNIST experiments use an ER topology with edge probability $p = 0.5$, here we consider a sparser graph with $p = 0.3$ and $N = 10$ clients. In this experiment, the client designated for unlearning is IID and contains no unique-label component, while the departing retained client contains an 80% unique-label component. The remaining experimental settings are kept unchanged.

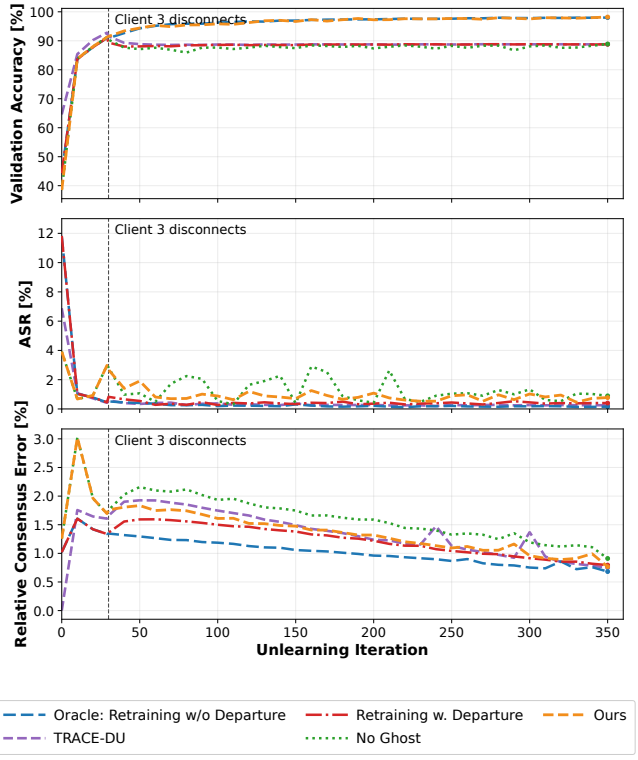

Figure 3: Validation accuracy, ASR, and relative consensus error throughout unlearning on MNIST with $N = 10$ clients over an ER communication graph with edge probability $p = 0.3$. The forgotten client is IID, while the departing retained client contains an 80% unique-label component.

Figure 3 shows that the same qualitative behavior observed in the main experiment persists under reduced graph connectivity. Following the retained-client disconnection, Ours remains close to the oracle retraining trajectory, whereas No Ghost, TRACE-DU, and Retraining with Departure exhibit a larger loss in validation accuracy. As expected, the sparser communication graph leads to

a somewhat higher relative consensus error, reflecting slower information propagation across the network. Nevertheless, the active clients remain close to consensus and the proposed method maintains low ASR. These results indicate that the benefit of ghost reconstruction persists under reduced communication-graph connectivity.

Retained-client heterogeneity. We further examine a more heterogeneous retained-client setting in which the departing client contains two client-specific labels. Together, these two unique labels account for 80% of the departing client's local data, while the client designated for unlearning is IID and contains no unique-label component.

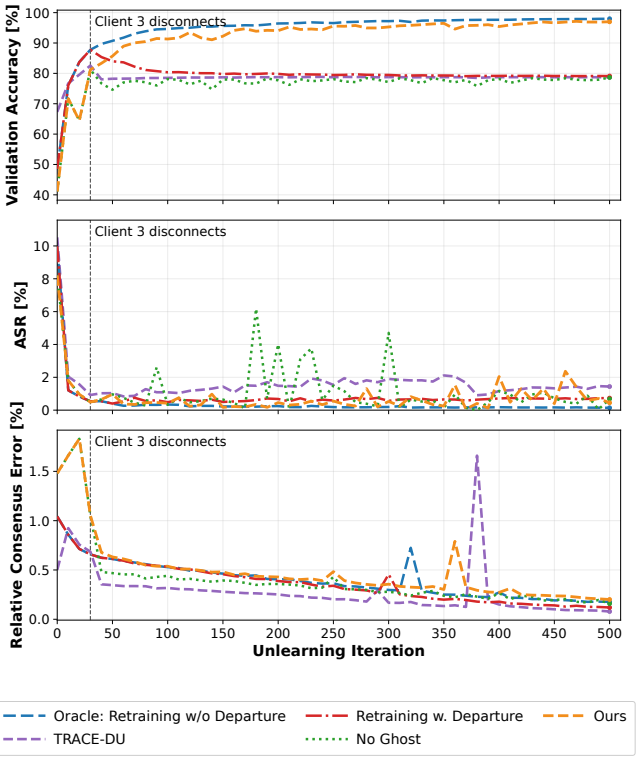

Figure 4: Validation accuracy, ASR, and relative consensus error throughout unlearning when the departing retained client contains two unique labels that together account for 80% of its local data. The forgotten client is IID.

Figure 4 shows that the proposed method remains close to the oracle retraining trajectory even when the unavailable retained client contributes information across multiple client-specific labels. In contrast, methods that do not reconstruct the departing client exhibit a pronounced reduction in validation accuracy after disconnection, reflecting the loss of information that is unavailable elsewhere in the network. Ours largely recovers this lost contribution while maintaining low ASR and close agreement among the active clients.

E.2.2 Robustness to Disconnection Dynamics

We next vary the disconnection process itself. We consider both multiple retained-client disconnections within the same unlearning trajectory and the timing of a retained-client disconnection. These experiments test whether the proposed reconstruction remains effective when client availability changes more substantially over the course of unlearning.

Multiple retained-client disconnections. We further evaluate the proposed method under multiple retained-client disconnections during unlearning. In this experiment, the client designated for unlearning is IID and contains no unique-label component, while two retained clients each contain an

80% unique-label component. The two retained clients disconnect sequentially during the unlearning process: client 3 disconnects at iteration 30, followed by client 1 at iteration 100.

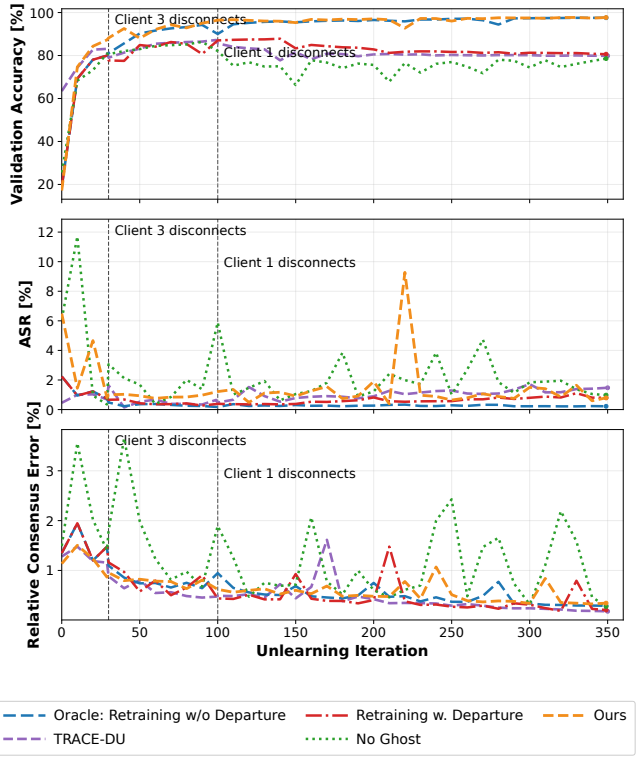

Figure 5: Validation accuracy, ASR, and relative consensus error throughout unlearning with two sequential retained-client disconnections. Client 3 disconnects at iteration 30 and client 1 disconnects at iteration 100. Each departing retained client contains an 80% unique-label component, while the forgotten client is IID.

Figure 5 shows that the proposed method remains effective under multiple sequential retained-client disconnections. Following the first departure, Ours remains close to the oracle retraining trajectory, whereas No Ghost, TRACE-DU, and Retraining with Departure begin to lose validation accuracy. After the second departure, this separation becomes larger, reflecting the additional loss of retained client-specific information in methods that do not reconstruct the unavailable clients.

Ours continues to preserve most of this contribution while maintaining low ASR and close agreement among the active clients. This experiment shows that the ghost mechanism can be maintained for multiple sequentially disconnected retained clients simultaneously, rather than being restricted to a single departure.

Disconnection time. We examine the sensitivity of the proposed method to the retained-client disconnection time.

Table 6: Effect of the retained-client disconnection round on final test accuracy (%). The forgotten client is IID, the retained client holds 60% of a unique label, and its disconnection round τ_r is varied over a 350-round unlearning process while all other settings are kept fixed.

	Accuracy (%)					
Method	$\tau_r = 30$	$\tau_r = 50$	$\tau_r = 100$	$\tau_r = 150$	$\tau_r = 200$	$\tau_r = 300$
Ours	**98.35**	**98.32**	**98.58**	**98.58**	**98.62**	**98.45**
No Ghost	89.42	89.59	89.59	89.61	89.89	96.26
TRACE-DU	88.83	89.02	90.09	92.14	94.21	96.84
Retraining w. Departure	89.12	89.85	91.33	93.91	94.92	96.96
Oracle: Retraining w/o Departure	*98.49*					

Ours is nearly insensitive to the disconnection time, maintaining a test accuracy between 98.32% and 98.62% across all departure rounds. Even when the retained client disconnects after only 30 of the 350 unlearning rounds, Ours remains within 0.17 percentage points of the oracle retraining reference.

Retraining w. Departure and TRACE-DU both improve as the disconnection is delayed, since the retained client's contribution remains available for a larger portion of the unlearning process. No Ghost shows a similar dependence, remaining near 89.4%–89.9% for disconnection rounds 30–200 and improving only when departure is delayed to round 300.

Overall, these results indicate that Ours preserves the unavailable client's contribution throughout the remaining unlearning iterations and closely tracks the oracle across all evaluated disconnection times. Certified-DFU is omitted from this sensitivity analysis because its unlearning procedure does not expose an analogous iterative disconnection-time parameter.

E.2.3 Sensitivity to Method-Inherent Parameters

We continue by studying sensitivity to the ratio-consensus budget, a method-specific parameter that controls the amount of additional communication used to reconstruct the missing mixing step after a retained-client disconnection.

Ratio-consensus budget. We vary the ratio-consensus budget K used to reconstruct the missing mixing step after a retained-client disconnection. We consider ER communication graphs with edge probabilities $p \in \{0.3, 0.5, 0.8\}$ and vary the ratio-consensus budget over $K \in \{2, 5, 10, 15\}$. For each connectivity level, we repeat the experiment over five graph realizations and report the mean and standard deviation. This averaging is particularly relevant since the reconstruction quality depends not only on the overall graph connectivity, but also on the position of the disconnected client and on the connectivity among its former neighbors in the remaining active graph. All other experimental settings are kept fixed.

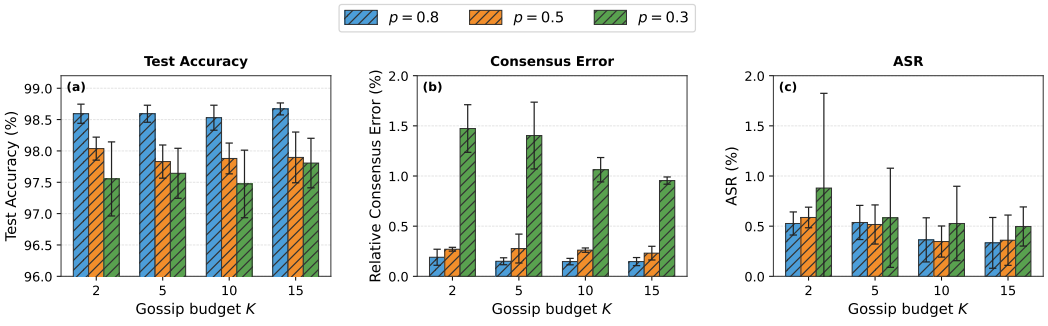

Figure 6: Effect of the ratio-consensus budget K under different communication-graph connectivities. Bars report the mean over five graph realizations and error bars indicate one standard deviation. We report test accuracy (left), relative consensus error (center), and ASR (right) for ER graphs with $p \in \{0.3, 0.5, 0.8\}$.

Figure 6 shows that a small ratio-consensus budget is already sufficient to achieve high test accuracy and low ASR across the considered connectivity levels. The effect of increasing K is more pronounced in the relative consensus error, particularly for the sparsest graphs. For $p = 0.3$, the consensus error decreases as additional ratio-consensus iterations are used, whereas for the better-connected graphs it is already small at low K and changes only modestly thereafter.

For $p = 0.3$ and $K = 2$, we also observe a somewhat higher mean ASR and substantially larger variability across graph realizations. This suggests that when both graph connectivity and the ratio-consensus budget are limited, the reconstruction is more sensitive to the particular graph realization. This variability decreases as additional ratio-consensus iterations are used.

Overall, this behavior is consistent with the role of the ratio-consensus procedure: additional iterations improve the propagation of the distributed contributions used to reconstruct the missing mixing step, with the largest benefit when information mixes more slowly over the remaining active graph. These results indicate that a large ratio-consensus budget is not required to maintain effective unlearning and model utility in the considered settings. Increasing K primarily improves agreement among the active clients and reduces sensitivity to the graph realization for less connected communication graphs, illustrating the communication–reconstruction tradeoff predicted by the analysis.

E.2.4 Generalization Across Model Architectures

To examine whether the observed behavior generalizes beyond the convolutional architecture used in the main experiments, we additionally evaluate the methods on CIFAR-10 using ResNet-20 with $N = 10$ clients. We consider two levels of client-specific information at the disconnected retained client, with 80% and 60% unique-label fractions. For each setting, the forgotten client is either IID or contains a 60% unique-label component. Unlike in the other experiments, in this setting the initial ASR is at least 91.03% across all four configurations.

Table 7: Final test accuracy (%) for CIFAR-10 with ResNet-20 and $N = 10$ clients. We consider two levels of client-specific information at the disconnected retained client. Within each group, the forgotten client is either IID or contains a 60% unique-label component.

	Accuracy (%)			
	Departing: 80% Unique		**Departing: 60% Unique**	
Method	**IID**	**60%**	**IID**	**60%**
Ours	**75.52**	**67.64**	**76.73**	**68.2**
No Ghost	74.38	67.18	74.49	66.69
TRACE-DU	74.44	66.92	74.3	66.94
Certified-DFU	63.6	60.02	65.1	62.49
Retraining w. Departure	74.45	66.79	74.13	66.99
Oracle: Retraining w/o Departure	*76.61*	*68.81*	*77.11*	*69.7*

As shown in Table 7, our method achieves the highest final test accuracy among the practical unlearning methods across all four configurations and remains close to the oracle obtained by retraining without a retained-client departure. The advantage over No Ghost, TRACE-DU, and Certified-DFU is preserved across both levels of client-specific information, indicating that the proposed approach remains effective when evaluated with the ResNet-20 architecture. For Certified-DFU, the final ASR remains relatively low, but exceeds 2% in some configurations and reaches up to 2.68%.

E.3 Centralized Experiments

We finally consider the centralized setting. Here, retained-client disconnection does not create a missing network-mixing operation, but the client's future local computations remain unavailable even though its contribution should remain part of the retained-client objective. These experiments therefore isolate the reconstruction of the unavailable client's local contribution from the additional decentralized challenge of reconstructing its missing mixing step.

Table 8: Final test accuracy (%) for Fashion-MNIST with $N = 10$ clients in the centralized setting. We consider two levels of client-specific information at the disconnected retained client. Within each group, the forgotten client is either IID or contains a 60% or 80% unique-label component.

	Departing: 100% Unique			Departing: 80% Unique		
Method	**IID**	**60%**	**80%**	**IID**	**60%**	**80%**
Ours	**87.61**	**78.18**	**78.66**	**88.60**	**78.14**	**78.01**
FedRecover	81.68	71.22	71.10	81.56	71.60	71.34
Fast-FedUL	72.92	63.98	57.57	84.16	63.78	66.23
Retraining w. Departure	81.85	72.41	72.48	82.12	72.49	72.47
Oracle: Retraining w/o Departure	*90.39*	*79.62*	*79.62*	*90.14*	*79.73*	*79.55*

The centralized results exhibit the same overall trend. Across the six data configurations, Ours improves final test accuracy over FedRecover by 5.93–7.56 percentage points, over Fast-FedUL by 4.44–21.09 points, and over Retraining w. Departure by 5.54–6.48 points, while remaining within 0.96–2.79 points of the oracle retraining reference. Fast-FedUL achieves competitive test accuracy in some configurations, but its attack-removal performance is less consistent, with the final mean ASR reaching 29.70% in the worst case.

F Limitations and Scope

The proposed framework enables decentralized unlearning even when the desired counterfactual trajectory can no longer be reproduced by direct retraining after a retained client disconnects. This capability entails several resource and modeling tradeoffs, which we summarize below.

Memory Overhead. The reconstruction procedure relies on optimization information retained from the original training trajectory, introducing additional memory requirements relative to standard decentralized training. Importantly, this information consists of model-gradient histories rather than private training data. The overhead can be reduced using memory-efficient storage, checkpointing, or compression techniques developed for distributed unlearning Lang et al. (2026).

Additional Communication. After a retained client disconnects, reconstructing its missing mixing step requires K auxiliary consensus iterations among the active clients. This introduces a communication-accuracy tradeoff: increasing K improves the reconstruction of the unavailable mixing operation, as reflected by our analysis, at the cost of additional message exchanges. Our sensitivity study in Appendix E.2.3 further illustrates this tradeoff across different communication-graph connectivities, showing that relatively small values of K are sufficient in the considered settings. This overhead is incurred only after disconnection and is complementary to the computational savings obtained by replacing many direct gradient evaluations with trajectory reconstruction.

Dependence on Trajectory Information. The quality of reconstruction depends on how informative the stored training trajectory is for the counterfactual states encountered during unlearning. Highly noisy or rapidly varying optimization dynamics can therefore degrade the Taylor-L-BFGS approximation, particularly after disconnection, when exact updates of the unavailable retained client can no longer be obtained. The validity test and frozen-update safeguard in (11) limit the use of unreliable ghost gradients, but cannot completely eliminate this dependence. Our formulation also assumes that the retained client disconnects after a final successful handoff of the information required to initialize its ghost replicas; handling abrupt failures without such a handoff is left for future work.

Scope of the Theoretical Guarantees. Our analysis characterizes the deviation of the proposed dynamics from counterfactual decentralized retraining under standard assumptions used in convergence studies, including smooth strongly convex local objectives, a connected communication graph, and bounded reconstruction errors. These conditions enable an explicit trajectory-tracking guarantee but do not cover the full nonconvex and stochastic setting used in modern neural-network training. The experiments evaluate the method with stochastic optimization and nonconvex models, indicating

that the proposed reconstruction remains effective beyond the formal regime considered in the theory. Moreover, the analysis provides an optimization-dynamics guarantee rather than a certified-unlearning guarantee based on statistical indistinguishability.