ROBUST FEDERATED PRIMAL–DUAL LEARNING VIA DYNAMIC GRAPH TOPOLOGY CORRECTION UNDER SPARSE CLIENT PARTICIPATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated primal–dual methods rely on client-specific dual variables to correct drift, yet sparse participation leaves most of these variables without fresh updates. Mean-based compensation addresses this missing information by assigning every inactive client the same correction, overlooking differences in their relationships with active clients. We propose RFL-DFDC, which reconstructs missing dual increments from client relationships and couples these reconstructions to the server update. A dynamic graph determines how observed increments contribute to each inactive client's virtual dual. The same weights determine the corresponding contributions to the global dual, preserving its equality to the average of virtual client duals. Our analysis bounds reconstruction error, accounts for missing graph support, and characterizes how these errors propagate to nonconvex optimization and returning clients under stated regularity conditions. Experiments on heterogeneous federated benchmarks show higher accuracy and fewer communication rounds to target accuracy than mean compensation. A comparison with a simple cosine-based alternative further supports the value of the proposed relationship weights.

1 INTRODUCTION

Federated learning trains a shared model across clients whose data and availability vary (Li et al., 2023; Zhang et al., 2025). Primal–dual methods coordinate their local objectives through client-specific dual variables, which carry information about disagreement with the global model (Wang et al., 2022; Sun et al., 2024). These variables receive fresh updates when clients participate. When participation is sparse, the global model continues to evolve while most client states remain tied to earlier rounds. Maintaining useful dual states between client visits is therefore an essential part of learning under intermittent participation.

Virtual dual memories provide a way to update absent clients without requiring them to communicate. A-FedPD and A-FedPDSAM compensate for missing increments using the mean of the active clients' updates (Sun et al., 2024). This allows inactive states to evolve, but supplies the same surrogate to every inactive client. Under heterogeneous objectives, an observed update can be informative for one client and less relevant to another. Two inactive clients may, for example, align with different members of the active cohort. Uniform compensation cannot express this distinction. Figure 1 illustrates how relationships among clients can instead guide different reconstructions from the same observations.

Client-specific reconstruction must also account for the role of dual variables in the global update. An inactive client's virtual dual supplies the correction used when that client returns, while its contribution to the average dual enters the server update in the current round. Changing the virtual memory therefore changes both quantities. The reconstruction rule and the server correction must be designed together to preserve this relationship. This coupling is central to extending mean compensation to heterogeneous client relationships.

We introduce Robust Federated Primal–Dual Learning via Dynamic Graph Topology Correction (RFL-DFDC), which couples relation-aware dual completion with a consistent server update. The server maintains a graph of client relationships from model updates observed when clients partici-

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.

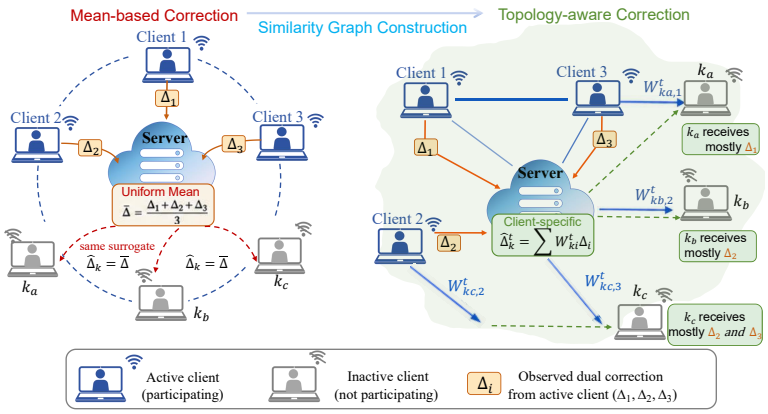


Figure 1: Schematic of client-specific dual completion. Mean compensation supplies the same surrogate to inactive clients. Relationship weights route observed corrections Δ_i^t into different reconstructions $\hat{\Delta}_k^t$; the server dual incorporates their average contribution.

pate together. This graph determines how active-client increments contribute to the virtual update of each inactive client, allowing compensation to depend on the client being reconstructed.

The key observation is that the reconstruction weights also determine the required server correction. Each row of the reconstruction kernel specifies an inactive client's update. Summing contributions along each column gives the influence of an active client on the completed population. Using these two views of the same operator preserves the global dual as the average of all virtual client duals exactly. Mean compensation follows as the uniform-weight special case; relationship weights allow the contributions to vary across clients while retaining the same averaging identity.

Our analysis separates this exact design property from the quality of the reconstructed increments. We derive a completion-error bound that includes clients without an observed connection to the active cohort. Under explicit regularity conditions, the analysis characterizes how reconstruction errors enter nonconvex stationarity and accumulate in the states supplied to returning clients. These results describe error propagation; the performance of the chosen relationship weights is evaluated empirically.

We evaluate RFL-DFDC on image classification and next-character prediction with heterogeneous client data and sparse participation. On the image benchmarks, the method improves accuracy and reaches target accuracies in fewer communication rounds than mean-based compensation. A comparison with compensation based on the last available updates further supports the proposed relationship weights. Component and correction-schedule studies examine the complementary roles of client and server updates and the benefit of continuing their correction throughout training.

2 Related Work

Federated consensus correction. Proximal regularization, control variates, and dynamic regularization address heterogeneous local updates (Li et al., 2020; Karimireddy et al., 2020; Acar et al., 2021). Primal–dual formulations also address hybrid data partitions (Overman et al., 2024) and partial client participation (Wang et al., 2022). A-FedPD and A-FedPDSAM are our closest algorithmic references: they maintain virtual duals for inactive clients and amplify the server correction using active-set means (Sun et al., 2024). RFL-DFDC retains this framework and replaces uniform missing increments with relationship-weighted, client-specific increments. Its influence coefficients follow from averaging the resulting virtual-memory changes.

Memory for missing updates. FedVARP uses stored client updates as surrogates for missing updates, while ClusterFedVARP reduces the memory requirement through clustering (Jhunjhunwala et al., 2022). MIFA also uses memorized updates to address device unavailability (Gu et al., 2021). These methods establish the broader role of server memory in compensating for inactive clients.

RFL-DFDC focuses on missing consensus-dual increments: each reconstruction changes both the state supplied to a returning client and the average dual used by the server.

Client relationships and local optimization. Client relationships support collaboration, selection, aggregation, and personalization (Ye et al., 2023; Chang et al., 2023; Wu et al., 2024; Chen et al., 2022; Jiao et al., 2025). Personalized methods also select client-specific parameters (Tamirisa et al., 2024) or calibrate feature extractors and classifiers (Tang et al., 2024). We use a relationship graph to complete dual increments while retaining a shared prediction model and the prescribed active cohort. The graph supplies a reconstruction prior from optimization signals, without requiring client labels or direct access to local data distributions. SAM and its federated variants modify local optimization (Foret et al., 2021; Sun et al., 2023a; Fan et al., 2024; Caldarola et al., 2025). Our implementation uses SAM, as does A-FedPDSAM. The proposed completion rule changes virtual memories and the server dual; its empirical evaluation is within this SAM-based implementation.

3 METHOD

3.1 PRELIMINARIES AND PROBLEM FORMULATION

We consider a FL system comprising N clients indexed by $\mathcal{V} = \{1, \dots, N\}$. Client k holds a private dataset $\mathcal{D}_k$ and minimizes a local objective $F_k(\cdot)$. The global learning objective is $\min_{\mathbf{w}} f(\mathbf{w}) \triangleq \frac{1}{N}\sum_{k=1}^{N} F_k(\mathbf{w})$. By introducing local model copies $\{\mathbf{w}_k\}_{k\in\mathcal{V}}$ and a global variable $\mathbf{w}_0$, we formulate the learning objective as the following problem:

$$\min_{\mathbf{w}_0,\{\mathbf{w}_k\}} \frac{1}{N}\sum_{k=1}^{N} F_k(\mathbf{w}_k) \quad \text{s.t.} \quad \mathbf{w}_k = \mathbf{w}_0,\ \forall k \in \mathcal{V}, \tag{1}$$

where $F_k(\mathbf{w}_k) = |\mathcal{D}_k|^{-1}\sum_{\xi_k\in\mathcal{D}_k} f_k(\mathbf{w}_k;\xi_k)$. We adopt the following sign convention for the augmented Lagrangian term:

$$\mathcal{L}_k(\mathbf{w}_0, \mathbf{w}_k, \boldsymbol{\lambda}_k) = F_k(\mathbf{w}_k) + \langle \boldsymbol{\lambda}_k, \mathbf{w}_0 - \mathbf{w}_k\rangle + \tfrac{\alpha}{2}\|\mathbf{w}_k - \mathbf{w}_0\|^2, \tag{2}$$

where $\alpha > 0$ is the penalty and $\boldsymbol{\lambda}_k$ is the client-specific consensus dual. This convention is equivalent to the standard convention under a reversal of the dual sign and makes the correction direction defined below explicit.

Active-client primal–dual update. At round t, the server samples a nonempty set of active clients $\mathcal{S}_t \subset \mathcal{V}$, where $P_t = |\mathcal{S}_t|$. Each active client i receives $(\mathbf{w}_0^t, \hat{\boldsymbol{\lambda}}_i^t)$ and approximately solves the corresponding local augmented subproblem to obtain $\mathbf{w}_i^{t+1}$. The client then updates the dual variable as follows:

$$\boldsymbol{\lambda}_i^{t+1} = \hat{\boldsymbol{\lambda}}_i^t + \alpha(\mathbf{w}_0^t - \mathbf{w}_i^{t+1}), \qquad \mathbf{z}_i^t \triangleq \boldsymbol{\lambda}_i^{t+1} - \hat{\boldsymbol{\lambda}}_i^t. \tag{3}$$

Thus the observed dual update satisfies the exact identity $\mathbf{z}_i^t = \alpha(\mathbf{w}_0^t - \mathbf{w}_i^{t+1})$.

Graph-theoretic view. Under non-IID data, clients with similar local distributions often produce correlated updates within related parameter subspaces. We represent this relationship at round t using a dynamic weighted graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t, \mathbf{W}^t)$, where $W_{i,j}^t \geq 0$ measures similarity between client i and j. We view the dual updates $\{\mathbf{z}_k^t\}$ as graph signals, where the updates of active clients are observed, while those of inactive clients are missing and must be reconstructed.

3.2 A SHARED OPERATOR FOR CLIENT AND SERVER CORRECTION

Figure 2 shows the information flow. A reconstruction kernel maps active dual increments to the virtual memories of inactive clients. Each row defines a client-specific reconstruction; the accumulated contributions along each column determine that active client's influence on the global dual. The server estimates this kernel from recent optimization signals and maintains both states using the same coefficients. Algorithm 1 specifies the complete update procedure.

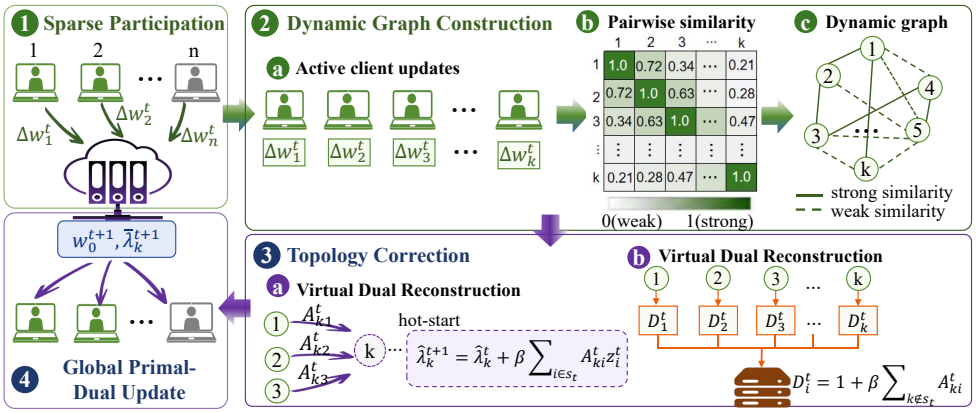


Figure 2: Framework of RFL-DFDC. Unlike mean-based correction, which assigns the same surrogate dual increment to all inactive clients, RFL-DFDC reconstructs client-specific missing dual increments through a dynamic client topology and uses the same topology to reweight the server-side dual update.

3.3 Dynamic Graph Topology Learning

Because raw data remain private, we learn $\mathbf{W}^t$ online from the model updates of active clients. Let $\Delta\mathbf{w}_i^t = \mathbf{w}_i^{t+1} - \mathbf{w}_0^t$. To ensure that the graph update remains well defined when an update has a small norm, we define the stabilized cosine similarity as follows:

$$\cos_\delta(\mathbf{u}, \mathbf{v}) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \, \|\mathbf{v}\| + \delta}, \qquad \delta > 0.$$

We use $\delta = 10^{-12}$ and initialize $\mathbf{W}^{-1} = \mathbf{I}$. For $i, j \in \mathcal{S}_t$, we update the graph weights using EMA:

$$W_{i,j}^t = (1-\gamma) W_{i,j}^{t-1} + \frac{\gamma}{2}\big(\cos_\delta(\Delta\mathbf{w}_i^t, \Delta\mathbf{w}_j^t) + 1\big), \tag{4}$$

with $\gamma \in (0, 1]$. The weights of unobserved client pairs retain their previous values. We further enforce symmetry by setting $W_{i,j}^t = W_{j,i}^t$.

3.4 Topology-Aware Virtual Dual Reconstruction

The server maintains a virtual dual memory $\hat{\boldsymbol{\lambda}}_k^t$ for each client. For an active client i, the observed state is stored:

$$\hat{\boldsymbol{\lambda}}_i^{t+1} \leftarrow \boldsymbol{\lambda}_i^{t+1}. \tag{5}$$

For an inactive client k, let $m_k^t = \sum_{j \in \mathcal{S}_t} W_{kj}^t$ denote its active-neighbor mass. We define the active-set diffusion kernel and the corresponding row mass as follows:

$$A_{ki}^t = \begin{cases} \dfrac{W_{ki}^t}{m_k^t}, & i \in \mathcal{S}_t \text{ and } m_k^t > \varepsilon_A, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $\varepsilon_A \geq 0$ is the fallback threshold used to identify unsupported rows. The corresponding row mass is $s_k^t \triangleq \sum_{i \in \mathcal{S}_t} A_{ki}^t \in \{0, 1\}$. RFL-DFDC reconstructs the missing dual increment and updates the virtual memory as follows:

$$\hat{\mathbf{z}}_k^t = \beta \sum_{i \in \mathcal{S}_t} A_{ki}^t \mathbf{z}_i^t, \qquad \hat{\boldsymbol{\lambda}}_k^{t+1} = \hat{\boldsymbol{\lambda}}_k^t + \hat{\mathbf{z}}_k^t, \tag{7}$$

where $\beta \in [0, 1]$. If the active-neighbor mass does not exceed ε_A, then $A_{ki}^t = 0$ for all i, $s_k^t = 0$, and the virtual memory of the inactive client remains unchanged. The analysis only requires nonnegative weights with $s_k^t \leq 1$, so it also covers additively stabilized or sparsified kernels.

Rejoin initialization. When client k next participates, the server sends $\hat{\boldsymbol{\lambda}}_k^t$ instead of the stale dual state last observed from that client. Relative to a recursively updated counterfactual memory, the rejoin error propagates through the local dual transition and accumulates the intervening reconstruction errors; Appendix B makes this dependence explicit.

3.5 Topology-Aware Global Dual and Server Updates

For each active client i, we define the topology influence weight as follows:

$$D_i^t = 1 + \beta \sum_{k \notin \mathcal{S}_t} A_{ki}^t. \tag{8}$$

The server dual variable is defined as the average of the virtual memories: $\boldsymbol{\lambda}_0^t \triangleq N^{-1} \sum_{k=1}^N \hat{\boldsymbol{\lambda}}_k^t$. Eqs. (5)–(8) imply the exact update

$$\boldsymbol{\lambda}_0^{t+1} - \boldsymbol{\lambda}_0^t = \frac{1}{N} \sum_{i \in \mathcal{S}_t} D_i^t \mathbf{z}_i^t = \frac{\alpha}{N} \sum_{i \in \mathcal{S}_t} D_i^t (\mathbf{w}_0^t - \mathbf{w}_i^{t+1}). \tag{9}$$

Thus, the global dual update is exactly the average change in the virtual memories. This algebraic identity holds for every admissible kernel, independently of reconstruction accuracy. Moreover, $\sum_{i \in \mathcal{S}_t} D_i^t = P_t + \beta \sum_{k \notin \mathcal{S}_t} s_k^t \le P_t + \beta(N - P_t)$. Let $\bar{\mathbf{w}}_{\mathcal{S}}^{t+1} = P_t^{-1} \sum_{i \in \mathcal{S}_t} \mathbf{w}_i^{t+1}$. Following the underlying A-FedPD-style method and the sign convention in Eq. (2), the server model is updated as follows:

$$\mathbf{w}_0^{t+1} = \bar{\mathbf{w}}_{\mathcal{S}}^{t+1} - \chi_t \boldsymbol{\lambda}_0^{t+1}, \tag{10}$$

where $\chi_t > 0$ controls the dual-to-primal scaling, with the canonical choice $\chi_t = 1/\alpha$. Any additional server-side damping can be incorporated into χ_t.

3.6 Consistency and Special Cases

Equation (9) explains why reconstruction and amplification are coupled. To see the effect of decoupling them, define the discrepancy $\mathbf{b}^t = \boldsymbol{\lambda}_0^t - N^{-1} \sum_k \hat{\boldsymbol{\lambda}}_k^t$. For arbitrary server coefficients D_i^t and applied inactive increments $\hat{\mathbf{z}}_k^t$, the update equations imply

$$\mathbf{b}^{t+1} - \mathbf{b}^t = \frac{1}{N} \sum_{i \in \mathcal{S}_t} (D_i^t - 1) \mathbf{z}_i^t - \frac{1}{N} \sum_{k \notin \mathcal{S}_t} \hat{\mathbf{z}}_k^t. \tag{11}$$

RFL-DFDC equates the two terms, preserving $\mathbf{b}^t = 0$ from zero initialization. Reconstruction alone omits inactive-memory changes from the server dual; amplification alone includes server contributions absent from those memories. The identity specifies how the two paths agree; Section 5.3 compares the training outcomes of the complete rule and these component interventions.

Relation to uniform compensation. For $A_{ki}^t = 1/P_t$, every inactive client receives β times the active mean, and $D_i^t = 1 + \beta(N - P_t)/P_t$. Thus uniform completion is a special case. Under full row support, the total influence is fixed at $\sum_i D_i^t = P_t + \beta(N - P_t)$, and the graph redistributes this influence among active clients. With incomplete support, the total becomes $P_t + \beta \sum_{k \notin \mathcal{S}_t} s_k^t$, tying amplification to the memories actually updated. Setting $\beta = 0$ recovers unchanged inactive memories and $D_i^t = 1$.

Local solver and implementation. The completion identities apply to the output of the local primal solver. The evaluated implementation uses SAM (Foret et al., 2021); its optimization error enters the host budget in Section 4. Graph construction uses the model updates already available to the server. The additional state consists of the graph and virtual dual memories, with no graph computation required at prediction time.

Implementation details. The complete procedure and computational cost are provided in Appendix E.

4 Completion Error and Optimization

The exact memory-average identity in Eq. (9) determines how reconstructed increments enter the server update. We next bound their completion error, including unsupported rows, and propagate this bound to stationarity under explicit host and graph-error conditions.

Let $\mathcal{F}_t$ contain the algorithmic history immediately before the round-t cohort is sampled, and write $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. For every inactive client $k \in \mathcal{I}_t \triangleq \mathcal{V} \setminus \mathcal{S}_t$, let $\mathbf{z}_k^t$ denote the one-step counterfactual dual increment obtained by applying the same local rule as Eq. (3) to the current state $(\mathbf{w}_0^t, \hat{\boldsymbol{\lambda}}_k^t)$. These oracle-completion objects are evaluated at the actual round-t state and do not define an alternative training trajectory.

Because the diffusion kernel can be sub-stochastic, we use the corrected graph energy

$$\overline{\mathcal{E}}_\beta^t \triangleq \sum_{k\in\mathcal{I}_t} \Big[\sum_{i\in\mathcal{S}_t} A_{ki}^t \|\beta \mathbf{z}_i^t - \mathbf{z}_k^t\|^2 + (1 - s_k^t)\|\mathbf{z}_k^t\|^2 \Big]. \tag{12}$$

The second term accounts for missing row mass. It vanishes for a normalized supported row and reduces to $\|\mathbf{z}_k^t\|^2$ for an unsupported row. Define the oracle-completed and topology-completed global dual increments by

$$\mathbf{g}^t = \frac{1}{N}\sum_{k=1}^N \mathbf{z}_k^t, \qquad \widetilde{\mathbf{g}}^t = \frac{1}{N}\sum_{i\in\mathcal{S}_t} D_i^t \mathbf{z}_i^t, \qquad \boldsymbol{\xi}^t = \widetilde{\mathbf{g}}^t - \mathbf{g}^t.$$

Using $\bar{\mathbf{w}}_{\mathcal{S}}^{t+1}$ from Eq. (10), define the oracle-completed host direction

$$\mathbf{h}^t = \chi_t^{-1}(\mathbf{w}_0^t - \bar{\mathbf{w}}_{\mathcal{S}}^{t+1}) + \boldsymbol{\lambda}_0^t + \mathbf{g}^t. \tag{13}$$

Equations (9) and (10) give the exact recursion

$$\mathbf{w}_0^{t+1} = \mathbf{w}_0^t - \chi_t(\mathbf{h}^t + \boldsymbol{\xi}^t). \tag{14}$$

Assumption 1 (Oracle-host regularity)**.** *The objective f is L-smooth and bounded below by $f^\star$. The oracle-completed host direction satisfies $\mathbf{h}^t = \nabla f(\mathbf{w}_0^t) + \mathbf{e}^t$, where $\mathbb{E}_t\|\mathbf{e}^t\|^2 \le v_t$ almost surely for a deterministic nonnegative sequence $\{v_t\}$. Appendix A gives sufficient local consensus and stationarity conditions. In particular, under conditionally uniform sampling given P_t, those conditions yield $v_t = 3(\chi_t^{-2} + L^2)c_t + 3r_t$, where c_t bounds the population mean squared local–global displacement and r_t bounds the mean squared stationarity residual of the augmented local subproblems.*

Assumption 2 (Corrected graph control)**.** *The reconstruction weights are nonnegative and satisfy $s_k^t \le 1$. There is a deterministic nonnegative sequence $\{q_{G,t}\}$ such that*

$$\mathbb{E}_t\left[\frac{N - P_t}{N^2}\overline{\mathcal{E}}_\beta^t\right] \le q_{G,t} \tag{15}$$

almost surely. No independence between the learned graph and the active-client updates is required. Under fixed-size sampling with $P_t = P$ and $\mathbb{E}_t[\overline{\mathcal{E}}_\beta^t] \le \varepsilon_{G,t}$, one may take $q_{G,t} = \frac{N-P}{N^2}\varepsilon_{G,t}$.

Lemma 1 (Topology completion error)**.** *Pathwise,*

$$\|\boldsymbol{\xi}^t\|^2 \le \frac{N - P_t}{N^2}\overline{\mathcal{E}}_\beta^t. \tag{16}$$

Consequently,

$$\mathbb{E}_t\|\boldsymbol{\xi}^t\|^2 \le q_{G,t}, \qquad \|\mathbb{E}_t\boldsymbol{\xi}^t\|^2 \le q_{G,t}. \tag{17}$$

Theorem 1 (Conditional nonconvex stationarity)**.** *Suppose Assumptions 1–2 hold and $0 < \chi_t \le 1/(4L)$. Let $S_T = \sum_{t=0}^{T-1}\chi_t$ and $\mathcal{G}_T = S_T^{-1}\sum_{t=0}^{T-1}\chi_t\mathbb{E}\|\nabla f(\mathbf{w}_0^t)\|^2$. Then*

$$\mathcal{G}_T \le \frac{2(f(\mathbf{w}_0^0) - f^\star)}{S_T} + \frac{4}{S_T}\sum_{t=0}^{T-1}\chi_t(1 + L\chi_t)\big(v_t + q_{G,t}\big). \tag{18}$$

If $\chi_t = \chi$ is constant, define $\bar{v}_T = T^{-1}\sum_{t<T} v_t$ and $\bar{q}_{G,T} = T^{-1}\sum_{t<T} q_{G,t}$. Then

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|\nabla f(\mathbf{w}_0^t)\|^2 \le \frac{2(f(\mathbf{w}_0^0) - f^\star)}{\chi T} + 4(1 + L\chi)(\bar{v}_T + \bar{q}_{G,T}). \tag{19}$$

Interpretation and applicability. For constant χ, Eq. (19) separates optimization progress from the accumulated host and graph-error budgets. Nonvanishing budgets leave an error floor. Applying its step-size condition to $\chi = 1/\alpha$ requires $\alpha \geq 4L$. The pathwise reconstruction bound applies to any cohort; the sufficient host budget additionally uses conditional uniform sampling. Persistent participation bias requires a corresponding host analysis.

Appendix A compares graph and mean error bounds conditionally on an energy inequality at a matched state; it does not establish that the chosen cosine–EMA graph satisfies this inequality. Performance advantages are evaluated in Tables 1–4. Appendix B traces reconstruction errors into rejoining client states and gives conditional stability of the complete state, including the graph and virtual memories.

5 Experiments

We evaluate how relation-aware dual completion affects learning when most clients are absent from each round. Alongside benchmark comparisons, we examine the contributions of client and server correction, the choice of relationship weights, and how often the correction should be applied.

5.1 Experimental Setup

We use CIFAR-10/100 (Krizhevsky et al., 2009), Tiny-ImageNet (Le & Yang, 2015), and Shakespeare (Caldas et al., 2018). Vision data are partitioned among 500 clients using a Dirichlet distribution with $\alpha_{\mathrm{Dir}} \in \{0.1, 0.6\}$. We train LeNet-style networks on CIFAR, ResNet-18 with GroupNorm on Tiny-ImageNet, and a two-layer LSTM on Shakespeare's 280 naturally partitioned clients. Unless otherwise specified, vision experiments run for 1600 rounds with five local epochs and Bernoulli participation at nominal rate $p = 0.02$, resampling empty cohorts. RFL-DFDC uses SAM with radius 0.1 and $\alpha = 0.1$, $\beta = 0.2$, and $\gamma = 0.1$.

We compare with FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), SCAFFOLD (Karimireddy et al., 2020), FedDyn (Acar et al., 2021), FedDC (Gao et al., 2022), A-FedPD/A-FedPDSAM (Sun et al., 2024), FedSpeed (Sun et al., 2023b), FedSMOO (Sun et al., 2023a), FedLESAM-D (Fan et al., 2024), and FedGloSS (Caldarola et al., 2025). A-FedPDSAM provides the closest comparison, combining mean-based dual compensation with SAM. We report final-round accuracy as mean±std over three runs and average learning curves pointwise. Appendix C gives the full training settings and metric definitions.

	CIFAR-10		CIFAR-100		Tiny-ImageNet	
Method	Acc↑	$R_{75}\downarrow$	Acc↑	$R_{40}\downarrow$	Acc↑	$R_{30}\downarrow$
FedAvg	75.13±0.12	–	28.56±0.04	–	28.31±0.05	1474
FedProx	74.78±0.14	–	29.01±0.05	–	28.23±0.06	–
SCAFFOLD	77.90±0.08	677	32.21±0.02	–	33.47±0.13	567
FedDyn	77.43±0.11	724	31.67±0.08	–	25.16±0.07	–
FedDC	78.30±0.16	727	39.49±0.04	–	31.54±0.04	654
A-FedPD	77.85±0.43	319	38.05±0.16	–	22.38±0.31	–
FedSpeed	81.69±0.08	473	42.44±0.02	1149	35.24±0.14	858
FedSMOO	81.81±0.06	461	47.25±0.02	795	34.69±0.07	693
FedLESAM-D	77.57±0.09	729	31.11±0.10	–	34.75±0.13	927
A-FedPDSAM	82.89±0.13	340	47.84±0.03	627	34.81±0.26	822
FedGloss	77.62±0.13	1078	38.23±0.02	–	35.20±0.18	913
RFL-DFDC	**84.37±0.09**	**235**	**51.81±0.01**	**547**	**39.27±0.04**	**542**

Table 1: Performance with 500 clients, nominal 2% participation, Dirichlet-0.6 data, and 1600 training rounds. Acc is final-round accuracy (mean±std over three runs, %). R_τ is the mean first round reaching τ% accuracy; "–" indicates that at least one run does not reach the target.

Correction	Acc↑	R_{40} ↓
Neither	38.46±1.04	–
Server only	41.72±0.61	1374
Virtual memory only	48.25±0.06	675
Both (RFL-DFDC)	51.81±0.01	547

Table 2: Client and server correction on CIFAR-100/Dirichlet-0.6. All variants retain active-client dual updates.

Weight source	Acc↑	R_{40} ↓
Mean ($\beta = 0.2$)	46.16±0.02	715
Last-update cosine	48.46±0.03	681
Random graph	37.29±0.20	–
Shuffled graph	41.74±0.11	1486
Frozen after 10	45.27±0.05	1064
Dynamic graph	51.81±0.01	547

Table 3: Relationship weights on CIFAR-100/Dirichlet-0.6. Both correction paths use $\beta = 0.2$ throughout. The frozen graph uses the first ten rounds of observations.

5.2 Performance under Sparse Participation

RFL-DFDC achieves the highest accuracy on all three image benchmarks in Table 1. The gain over A-FedPDSAM is larger on CIFAR-100 and Tiny-ImageNet than on CIFAR-10, exceeding three percentage points on both datasets. On Tiny-ImageNet, RFL-DFDC also improves on FedSpeed, the strongest baseline, by about four points. The advantage thus extends from the LeNet models to ResNet-18.

The accuracy improvement is accompanied by fewer communication rounds to reach each target. Relative to A-FedPDSAM, RFL-DFDC reduces the required rounds by approximately 31% on CIFAR-10 and 13% on CIFAR-100. It also reaches the Tiny-ImageNet target before SCAFFOLD, which is the fastest baseline for that target despite its lower final accuracy.

5.3 Client and Server Correction

Updating virtual memories provides the larger individual gain in Table 2. These memories supply returning clients with a correction accumulated during their absence. Adding the matching server update further improves accuracy by 3.56 percentage points and reduces the rounds needed to reach the target, allowing the reconstructed increments to contribute to the global dual immediately.

The two paths act on different parts of the training state. Retaining just one breaks the average-memory identity in Eq. (11); applying both gives the best performance. The comparison favors correcting both the states supplied to returning clients and the global dual used between their visits.

5.4 Effect of Relationship Weights

With both correction paths enabled, we compare alternative weighting rules at the same compensation strength (Table 3). Using cosine similarities between the last available client updates already improves on uniform weights. The dynamic graph yields a further 3.35 percentage-point gain while reaching the target in fewer rounds. Here, Mean uses $\beta = 0.2$, matching RFL-DFDC; the native A-FedPDSAM comparison appears in Table 1.

Randomizing neighborhoods or shuffling client identities reduces performance substantially. In particular, shuffling preserves the distribution of learned weights but changes which clients they connect, indicating that their assignment to clients matters. Continuing to collect co-participation observations also outperforms freezing the graph after its first ten rounds.

5.5 Correction Frequency and Client Availability

Applying the coupled correction throughout training gives the highest accuracy and the fewest rounds to target (Table 4). Intermittent correction reduces the work per round but requires more communication to reach the same accuracy. Stopping after a brief warm-up performs worse than either periodic schedule, showing that the benefit of correction extends beyond initialization.

The advantage persists as participation becomes sparser (Figures 4 and 5(b)). At nominal 0.5% participation, RFL-DFDC retains 36.4% accuracy on CIFAR-100, whereas A-FedPDSAM reaches

Correction schedule	Acc↑	R_{40} ↓	T/R (s)↓
Stop after 5 rounds	40.48±0.66	–	2.34
Stop after 10 rounds	40.33±0.79	–	2.42
Update every 10 rounds	45.54±0.72	1118	3.09
Update every 5 rounds	48.22±0.38	941	3.72
Update every round	51.81±0.01	547	5.05

Table 4: Application frequency of the complete correction on CIFAR-100 under nominal 2% participation and Dirichlet-0.6. Reconstruction and its matching server correction are scheduled together. Acc is final-round mean±std across runs; R_{40} and T/R are run means. "–" means that at least one run does not reach 40% within 1,600 rounds.

3.9%. On Shakespeare, RFL-DFDC also achieves the highest next-character accuracy among the compared methods (Figure 5(a)), extending the evaluation to naturally partitioned language data.

5.6 Computation and Parameter Sensitivity

The additional server computation has a small effect on round time in the CIFAR-100 experiment. RFL-DFDC takes about 1.6% longer per round than A-FedPDSAM, while its lower round count gives a shorter estimated time to target (Table 7). Appendix C describes the NVIDIA L40 setup and the estimate obtained from mean round times and target-round counts.

Virtual dual memories dominate server storage, and reconstructing them accounts for most of the correction time (Table 8). The principal scaling cost therefore comes from maintaining a full model-sized state for each client. Graph storage contributes comparatively little at the evaluated scale.

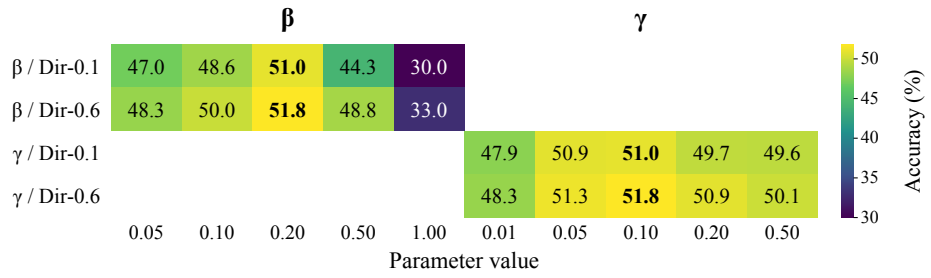


Figure 3: Sensitivity to β and γ on CIFAR-100. The β sweep fixes $\gamma = 0.1$, and the γ sweep fixes $\beta = 0.2$. Accuracies are rounded to one decimal place.

Figure 3 shows that moderate reconstruction strength performs best under both heterogeneity settings. Increasing β to full strength is less effective, while intermediate values of γ give similar accuracies. This favors using the inferred updates as a partial correction, with performance less sensitive to the precise EMA factor in the middle of the tested range.

6 Conclusion

RFL-DFDC couples relation-aware completion of missing dual increments with a consistent server update. The reconstruction kernel determines both inactive-client memory changes and active-client influence, preserving the global dual as the average of virtual client memories by construction. The analysis bounds completion error, including unsupported rows, and describes its propagation under explicit regularity conditions. Experiments demonstrate higher accuracy and fewer rounds to target in the evaluated sparse-participation settings. Component interventions favor using both correction paths, relationship-weight comparisons show gains over mean and last-update cosine compensation, and correction-frequency controls support continued application throughout training. These results support relation-aware dual completion with a matching server correction under sparse participation.

AI USE STATEMENT

Generative AI tools assisted with revision planning, literature and consistency checks, LaTeX migration, manuscript editing and restructuring around existing results, and preparation of experiment-reporting templates. AI assistance did not generate experimental measurements. The authors are responsible for validating the analysis, experimental evidence, and final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix E describe the method and algorithm. Section 5.1 and Appendix C document the datasets, client partitions, model architectures, optimization settings, random seeds, and evaluation metrics. Appendices A and B provide the assumptions and proofs for the theoretical results.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Debora Caldarola, Pietro Cagnasso, Barbara Caputo, and Marco Ciccone. Beyond local sharpness: Communication-efficient global sharpness-aware minimization for federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25187–25197, 2025.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Lei Chang, Peiyuan Li, Rui Wang, Jiandong Yu, Chenchen Wang, Jianping Wu, and Di Wu. Graphcs: Graph-based client selection for heterogeneous federated learning. *J. Parallel Distrib. Comput.*, 180:84–96, 2023. doi: 10.1016/j.jpdc.2023.06.003. URL https://doi.org/10.1016/j.jpdc.2023.06.003.

F Chen, G Long, Z Wu, T Zhou, and J Jiang. Personalized federated learning with a graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2022.

Ziqing Fan, Shengchao Hu, Jiangchao Yao, Gang Niu, Ya Zhang, Masashi Sugiyama, and Yanfeng Wang. Locally estimated global perturbations are better than local perturbations for federated sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 12858–12881. PMLR, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10112–10121, 2022.

Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/64be20f6dd1dd46adf110cf871e3ed35-Abstract.html.

Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. FedVARP: Tackling the variance due to partial client participation in federated learning. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180, pp. 906–916. PMLR, 2022. URL https://proceedings.mlr.press/v180/jhunjhunwala22a.html.

Pengfei Jiao, Zian Zhou, Meiting Xue, Huijun Tang, Zhidong Zhao, and Huaming Wu. Fedcch: Automatic personalized graph federated learning for inter-client and intra-client heterogeneity. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2991–2999, 2025.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html.

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

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

Peichun Li, Guoliang Cheng, Xumin Huang, Jiawen Kang, Rong Yu, Yuan Wu, and Miao Pan. Anycostfl: Efficient on-demand federated learning over heterogeneous edge devices. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

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

Tom Overman, Garrett Blum, and Diego Klabjan. A primal-dual algorithm for hybrid federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 14482–14489, 2024.

Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International conference on machine learning*, pp. 32991–33013. PMLR, 2023a.

Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. In *The Eleventh International Conference on Learning Representations*, 2023b.

Yan Sun, Li Shen, and Dacheng Tao. A-fedpd: Aligning dual-drift is all federated primal-dual learning needs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Zhou, Ron Arel, and Aviv Shamsian. Fedselect: Personalized federated learning with customized selection of parameters for fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23985–23994, 2024.

Xiaoli Tang, Han Yu, Run Tang, Chao Ren, Anran Li, and Xiaoxiao Li. Dual calibration-based personalised federated learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4982–4990, 2024.

Han Wang, Siddartha Marella, and James Anderson. FedADMM: A federated primal-dual algorithm allowing partial participation. *arXiv preprint arXiv:2203.15104*, 2022. URL https://arxiv.org/abs/2203.15104.

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

Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. Personalized federated learning with inferred collaboration graphs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39801–39817. PMLR, 2023. URL https://proceedings.mlr.press/v202/ye23b.html.

Hao Zhang, Chenglin Li, Wenrui Dai, Ziyang Zheng, Junni Zou, and Hongkai Xiong. Stabilizing and accelerating federated learning on heterogeneous data with partial client participation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):67–83, 2025.

SUPPLEMENTARY MATERIAL (APPENDICES)

A CONVERGENCE ANALYSIS

This appendix proves the theoretical results for the explicit server update in Eq. (10) of the main paper. All conditional expectations are taken with respect to the pre-sampling history $\mathcal{F}_t$ defined in the main text; they include cohort sampling and any local-solver randomness at round t.

A.1 EXACT TOPOLOGY-COMPLETION BOUND

For $k \in \mathcal{I}_t$, write

$$\hat{\mathbf{z}}_k^t = \beta \sum_{i \in \mathcal{S}_t} A_{ki}^t \mathbf{z}_i^t, \qquad \mathbf{d}_k^t = \hat{\mathbf{z}}_k^t - \mathbf{z}_k^t.$$

The influence weights in Eq. (8) of the main paper imply the exact identity

$$\widetilde{\mathbf{g}}^t = \frac{1}{N}\left(\sum_{i \in \mathcal{S}_t} \mathbf{z}_i^t + \sum_{k \in \mathcal{I}_t} \hat{\mathbf{z}}_k^t\right), \qquad \boldsymbol{\xi}^t = \frac{1}{N}\sum_{k \in \mathcal{I}_t} \mathbf{d}_k^t. \tag{20}$$

Proof of Main-Paper Lemma 1. By Cauchy–Schwarz and $|\mathcal{I}_t| = N - P_t$,

$$\|\boldsymbol{\xi}^t\|^2 \le \frac{N - P_t}{N^2} \sum_{k \in \mathcal{I}_t} \|\mathbf{d}_k^t\|^2. \tag{21}$$

For each inactive client,

$$\mathbf{d}_k^t = \sum_{i \in \mathcal{S}_t} A_{ki}^t (\beta \mathbf{z}_i^t - \mathbf{z}_k^t) - (1 - s_k^t)\mathbf{z}_k^t. \tag{22}$$

The coefficients $\{A_{ki}^t : i \in \mathcal{S}_t\} \cup \{1 - s_k^t\}$ are nonnegative and sum to one. Convexity of the squared norm therefore gives

$$\|\mathbf{d}_k^t\|^2 \le \sum_{i \in \mathcal{S}_t} A_{ki}^t \|\beta \mathbf{z}_i^t - \mathbf{z}_k^t\|^2 + (1 - s_k^t)\|\mathbf{z}_k^t\|^2. \tag{23}$$

Summing Eq. (23) over $k \in \mathcal{I}_t$ and substituting it into Eq. (21) proves the pathwise bound (16) in the main paper. Taking $\mathbb{E}_t$, applying Assumption 2 of the main paper, and then using conditional Jensen yields both inequalities in Eq. (17) of the main paper. □

A.2 A SUFFICIENT ORACLE-HOST ERROR BUDGET

For this subsection, let $\breve{\mathbf{w}}_k^{t+1}$ denote the one-step oracle output obtained by running the same local solver for client k from $(\mathbf{w}_0^t, \hat{\boldsymbol{\lambda}}_k^t)$. For active clients, $\breve{\mathbf{w}}_i^{t+1} = \mathbf{w}_i^{t+1}$. Define the stationarity residual using the sign convention of Eq. (2) of the main paper:

$$\mathbf{u}_k^t \triangleq \nabla F_k(\breve{\mathbf{w}}_k^{t+1}) - \hat{\boldsymbol{\lambda}}_k^t + \alpha(\breve{\mathbf{w}}_k^{t+1} - \mathbf{w}_0^t). \tag{24}$$

The associated oracle dual increment is

$$\mathbf{z}_k^t = \alpha(\mathbf{w}_0^t - \breve{\mathbf{w}}_k^{t+1}) = \nabla F_k(\breve{\mathbf{w}}_k^{t+1}) - \hat{\boldsymbol{\lambda}}_k^t - \mathbf{u}_k^t. \tag{25}$$

Lemma 2 (Sufficient oracle-host regularity)**.** *Assume each F_k is L-smooth. Conditional on P_t, suppose $\mathcal{S}_t$ is uniform among all subsets of size P_t and is conditionally independent of the oracle local-solver randomness. Suppose deterministic sequences $c_t, r_t \ge 0$ satisfy*

$$\mathbb{E}_t\left[\frac{1}{N}\sum_{k=1}^N \|\breve{\mathbf{w}}_k^{t+1} - \mathbf{w}_0^t\|^2\right] \le c_t, \qquad \mathbb{E}_t\left[\frac{1}{N}\sum_{k=1}^N \|\mathbf{u}_k^t\|^2\right] \le r_t. \tag{26}$$

Then Assumption 1 of the main paper holds with

$$v_t = 3(\chi_t^{-2} + L^2)c_t + 3r_t. \tag{27}$$

Proof. Because $\boldsymbol{\lambda}_0^t = N^{-1}\sum_k \hat{\boldsymbol{\lambda}}_k^t$, averaging Eq. (25) gives

$$\boldsymbol{\lambda}_0^t + \mathbf{g}^t = \frac{1}{N}\sum_{k=1}^N \nabla F_k(\breve{\mathbf{w}}_k^{t+1}) - \bar{\mathbf{u}}^t, \qquad \bar{\mathbf{u}}^t \triangleq \frac{1}{N}\sum_{k=1}^N \mathbf{u}_k^t. \tag{28}$$

Substituting Eq. (28) into Eq. (13) of the main paper yields the exact decomposition

$$\begin{aligned}\mathbf{e}^t = \chi_t^{-1}&\left(\mathbf{w}_0^t - \frac{1}{P_t}\sum_{i\in\mathcal{S}_t}\breve{\mathbf{w}}_i^{t+1}\right)\\ &+ \frac{1}{N}\sum_{k=1}^N\left(\nabla F_k(\breve{\mathbf{w}}_k^{t+1}) - \nabla F_k(\mathbf{w}_0^t)\right) - \bar{\mathbf{u}}^t.\end{aligned} \tag{29}$$

Using $\|\mathbf{a}+\mathbf{b}+\mathbf{c}\|^2 \le 3(\|\mathbf{a}\|^2+\|\mathbf{b}\|^2+\|\mathbf{c}\|^2)$, Jensen's inequality, and L-smoothness, we obtain

$$\begin{aligned}\mathbb{E}_t\|\mathbf{e}^t\|^2 \le{}& 3\chi_t^{-2}\mathbb{E}_t\left[\frac{1}{P_t}\sum_{i\in\mathcal{S}_t}\|\breve{\mathbf{w}}_i^{t+1}-\mathbf{w}_0^t\|^2\right]\\ &+ 3L^2\mathbb{E}_t\left[\frac{1}{N}\sum_{k=1}^N\|\breve{\mathbf{w}}_k^{t+1}-\mathbf{w}_0^t\|^2\right] + 3\mathbb{E}_t\left[\frac{1}{N}\sum_{k=1}^N\|\mathbf{u}_k^t\|^2\right].\end{aligned}$$

Conditional uniformity of $\mathcal{S}_t$ makes the first sample average unbiased for the population average. Applying Eq. (26) proves Eq. (27). □

Effect of SAM. Lemma 2 treats the local solver only through r_t. To make the SAM dependence explicit, suppose the normalized perturbation has radius ρ_{sam} and the residual computed with the perturbed gradient is $\widetilde{\mathbf{u}}_k^t$. If

$$\mathbf{u}_k^t = \widetilde{\mathbf{u}}_k^t + \nabla F_k(\breve{\mathbf{w}}_k^{t+1}) - \nabla F_k(\breve{\mathbf{w}}_k^{t+1}+\mathbf{p}_k^t), \qquad \|\mathbf{p}_k^t\| \le \rho_{\mathrm{sam}},$$

then L-smoothness gives

$$\frac{1}{N}\sum_k \mathbb{E}_t\|\mathbf{u}_k^t\|^2 \le 2\frac{1}{N}\sum_k \mathbb{E}_t\|\widetilde{\mathbf{u}}_k^t\|^2 + 2L^2\rho_{\mathrm{sam}}^2. \tag{30}$$

Thus SAM contributes through the local stationarity budget; Eq. (30) does not require treating the SAM update as an exact minimizer of the unperturbed subproblem.

A.3 Proof of the Nonconvex Rate

Proof of Main-Paper Theorem 1. Let $\mathbf{a}^t = \nabla f(\mathbf{w}_0^t)$ and $\boldsymbol{\delta}^t = \mathbf{e}^t + \boldsymbol{\xi}^t$. The exact recursion (14) of the main paper becomes $\mathbf{w}_0^{t+1} = \mathbf{w}_0^t - \chi_t(\mathbf{a}^t + \boldsymbol{\delta}^t)$. By L-smoothness,

$$f(\mathbf{w}_0^{t+1}) \le f(\mathbf{w}_0^t) - \chi_t\|\mathbf{a}^t\|^2 - \chi_t\langle\mathbf{a}^t,\boldsymbol{\delta}^t\rangle + \frac{L\chi_t^2}{2}\|\mathbf{a}^t+\boldsymbol{\delta}^t\|^2. \tag{31}$$

Young's inequality and $\|\mathbf{a}+\mathbf{b}\|^2 \le 2\|\mathbf{a}\|^2+2\|\mathbf{b}\|^2$ give

$$-\langle\mathbf{a}^t,\boldsymbol{\delta}^t\rangle \le \tfrac{1}{4}\|\mathbf{a}^t\|^2 + \|\boldsymbol{\delta}^t\|^2, \qquad \frac{L\chi_t^2}{2}\|\mathbf{a}^t+\boldsymbol{\delta}^t\|^2 \le L\chi_t^2(\|\mathbf{a}^t\|^2+\|\boldsymbol{\delta}^t\|^2).$$

Since $L\chi_t \le 1/4$, Eq. (31) implies

$$f(\mathbf{w}_0^{t+1}) \le f(\mathbf{w}_0^t) - \frac{\chi_t}{2}\|\nabla f(\mathbf{w}_0^t)\|^2 + \chi_t(1+L\chi_t)\|\boldsymbol{\delta}^t\|^2. \tag{32}$$

No independence between $\mathbf{e}^t$ and $\boldsymbol{\xi}^t$ is needed: by $\|\mathbf{e}^t+\boldsymbol{\xi}^t\|^2 \le 2\|\mathbf{e}^t\|^2+2\|\boldsymbol{\xi}^t\|^2$, Assumptions 1–2 and Lemma 1 of the main paper yield

$$\mathbb{E}_t\|\boldsymbol{\delta}^t\|^2 \le 2(v_t + q_{G,t}).$$

Taking conditional and then total expectation in Eq. (32), summing from $t=0$ to $T-1$, and using $f(\mathbf{w}_0^T) \ge f^\star$ gives

$$\frac{1}{2}\sum_{t=0}^{T-1}\chi_t\mathbb{E}\|\nabla f(\mathbf{w}_0^t)\|^2 \le f(\mathbf{w}_0^0) - f^\star + 2\sum_{t=0}^{T-1}\chi_t(1+L\chi_t)(v_t+q_{G,t}).$$

Dividing by $S_T/2$ proves Eq. (18) of the main paper. The constant-χ statement follows by substituting $S_T = \chi T$ and the definitions of $\bar{v}_T$ and $\bar{q}_{G,T}$. □

A.4 Conditional Comparison to Mean Completion

Corollary 1 (Conditional comparison of error bounds)**.** *For the matched round-t state, let $A^t_{\mathrm{mean,ki}} = 1/P_t$ for $i \in \mathcal{S}_t$. If, for some $\rho \in [0, 1)$,*

$$\mathbb{E}_t\Big[(N - P_t)\overline{\mathcal{E}}^t_\beta(A_{\mathrm{graph}})\Big] \le \rho\, \mathbb{E}_t\Big[(N - P_t)\overline{\mathcal{E}}^t_\beta(A_{\mathrm{mean}})\Big],$$

then the topology-error contribution in the one-step descent bound, and hence its contribution to Theorem 1 with the host term held fixed, is at most ρ times the corresponding mean-correction contribution. This comparison assumes the displayed energy inequality and holds the host term fixed; the inequality is not derived for the chosen cosine–EMA graph.

Proof of Corollary 1. At a matched roundwise state, the graph and mean kernels share P_t. Lemma 1 bounds the expected squared completion error by $N^{-2}\mathbb{E}_t[(N - P_t)\overline{\mathcal{E}}^t_\beta]$. Dividing the assumed weighted-energy comparison by N^2 and substituting the resulting error-budget bounds into Eq. (32) proves the componentwise comparison of these bounds. The random factor $N - P_t$ must remain inside the expectation for variable-size cohorts. For fixed $P_t = P < N$, it cancels from the premise. The argument holds the host-error budget fixed and makes no claim about independently trained trajectories. □

A.5 Exact Energy Decomposition

For a supported row, let $\boldsymbol{\mu}^t_k = \sum_i A^t_{ki}\mathbf{z}^t_i$. The exact identity

$$\sum_i A^t_{ki}\|\beta\mathbf{z}^t_i - \mathbf{z}^t_k\|^2 = \|\beta\boldsymbol{\mu}^t_k - \mathbf{z}^t_k\|^2 + \beta^2 \sum_i A^t_{ki}\|\mathbf{z}^t_i - \boldsymbol{\mu}^t_k\|^2 \tag{33}$$

separates the squared prediction error from within-neighbor dispersion. Thus a smaller graph energy alone does not establish smaller prediction error, and neither quantity alone establishes higher test accuracy. The energy is used here as an analytical error bound; the empirical evidence in Section 5 comes from training outcomes and controlled algorithm interventions.

For a supported row $\sum_i A_{ki} = 1$, write $\boldsymbol{\mu}_k = \sum_i A_{ki}\mathbf{z}_i$. Expanding $\beta\mathbf{z}_i - \mathbf{z}_k = \beta(\mathbf{z}_i - \boldsymbol{\mu}_k) + (\beta\boldsymbol{\mu}_k - \mathbf{z}_k)$ and summing cancels the cross term because $\sum_i A_{ki}(\mathbf{z}_i - \boldsymbol{\mu}_k) = 0$. This proves Eq. (33). For a sub-stochastic row, the corrected energy has the exact form

$$\sum_i A_{ki}\|\beta\mathbf{z}_i - \mathbf{z}_k\|^2 + (1 - s_k)\|\mathbf{z}_k\|^2 = \|\beta\boldsymbol{\mu}_k - \mathbf{z}_k\|^2 + \beta^2\left(\sum_i A_{ki}\|\mathbf{z}_i\|^2 - \|\boldsymbol{\mu}_k\|^2\right).$$

The last bracket is nonnegative by Jensen after adding a zero-valued atom with mass $1 - s_k$. For a zero row it vanishes, leaving exactly the squared norm of the missing oracle increment.

B Rejoin-State and Full-State Stability

This appendix separates two stability questions. The first compares the reconstructed virtual dual memory with a continuously refreshed exact local transition along the realized global-model path. The second studies perturbations of the complete algorithmic state, including the learned graph and all virtual memories.

B.1 Exact Local Transition and Rejoin Error

Fix a client k and round t. For a dual input $\boldsymbol{\lambda}$, define the exact local subproblem and its dual transition by

$$\mathbf{w}_{k,t}(\boldsymbol{\lambda}) \triangleq \arg\min_{\mathbf{w}} \left\{F_k(\mathbf{w}) + \langle\boldsymbol{\lambda}, \mathbf{w}^t_0 - \mathbf{w}\rangle + \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}^t_0\|^2\right\}, \tag{34}$$

$$\mathcal{T}_{k,t}(\boldsymbol{\lambda}) \triangleq \boldsymbol{\lambda} + \alpha\big(\mathbf{w}^t_0 - \mathbf{w}_{k,t}(\boldsymbol{\lambda})\big). \tag{35}$$

Lemma 3 (Lipschitz exact transition)**.** *Suppose F_k is differentiable and L-smooth, and let $\alpha > L$. Then the minimizer in Eq.* (34) *is unique and*

$$\|\mathbf{w}_{k,t}(\boldsymbol{\lambda}) - \mathbf{w}_{k,t}(\boldsymbol{\lambda}')\| \le \frac{1}{\alpha - L}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|, \tag{36}$$

$$\|\mathcal{T}_{k,t}(\boldsymbol{\lambda}) - \mathcal{T}_{k,t}(\boldsymbol{\lambda}')\| \le q\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|, \qquad q \triangleq \frac{L}{\alpha - L}. \tag{37}$$

Hence the dual transition is contractive when $\alpha > 2L$.

Proof. The function in Eq. (34) is $(\alpha - L)$-strongly convex because an L-smooth function is L-weakly convex. Its first-order condition is

$$\nabla F_k(\mathbf{w}_{k,t}(\boldsymbol{\lambda})) - \boldsymbol{\lambda} + \alpha(\mathbf{w}_{k,t}(\boldsymbol{\lambda}) - \mathbf{w}_0^t) = \mathbf{0}. \tag{38}$$

Let $\mathbf{d} = \mathbf{w}_{k,t}(\boldsymbol{\lambda}) - \mathbf{w}_{k,t}(\boldsymbol{\lambda}')$. Subtracting the two optimality conditions and taking the inner product with $\mathbf{d}$ gives

$$\langle \boldsymbol{\lambda} - \boldsymbol{\lambda}', \mathbf{d}\rangle = \langle \nabla F_k(\mathbf{w}_{k,t}(\boldsymbol{\lambda})) - \nabla F_k(\mathbf{w}_{k,t}(\boldsymbol{\lambda}')), \mathbf{d}\rangle + \alpha\|\mathbf{d}\|^2 \ge (\alpha - L)\|\mathbf{d}\|^2.$$

Cauchy–Schwarz proves Eq. (36). Moreover, Eq. (38) implies $\mathcal{T}_{k,t}(\boldsymbol{\lambda}) = \nabla F_k(\mathbf{w}_{k,t}(\boldsymbol{\lambda}))$. Applying L-smoothness and Eq. (36) proves Eq. (37). □

Consider an interval $\{a, \dots, b-1\}$ during which client k remains inactive, starting from a directly refreshed memory at the beginning of round a and ending when the client rejoins at round b. Define the continuously refreshed comparator along the realized server path by

$$\boldsymbol{\lambda}_k^{\circ,a} = \hat{\boldsymbol{\lambda}}_k^a, \qquad \boldsymbol{\lambda}_k^{\circ,t+1} = \mathcal{T}_{k,t}(\boldsymbol{\lambda}_k^{\circ,t}), \quad a \le t < b. \tag{39}$$

For the exact local rule, the counterfactual increment used in the main theoretical analysis satisfies

$$\mathbf{z}_k^t = \mathcal{T}_{k,t}(\hat{\boldsymbol{\lambda}}_k^t) - \hat{\boldsymbol{\lambda}}_k^t.$$

Let $r_k^t \triangleq \|\hat{\mathbf{z}}_k^t - \mathbf{z}_k^t\|$ denote the one-step reconstruction error.

Lemma 4 (Accumulated rejoin-state error)**.** *Under the conditions of Lemma 3,*

$$\|\hat{\boldsymbol{\lambda}}_k^b - \boldsymbol{\lambda}_k^{\circ,b}\| \le \sum_{t=a}^{b-1} q^{\,b-1-t} r_k^t. \tag{40}$$

If $\alpha > 2L$, then $q < 1$ and

$$\|\hat{\boldsymbol{\lambda}}_k^b - \boldsymbol{\lambda}_k^{\circ,b}\|^2 \le \frac{1}{1-q}\sum_{t=a}^{b-1} q^{\,b-1-t}\left[\sum_{i\in\mathcal{S}_t} A_{ki}^t\|\beta\mathbf{z}_i^t - \mathbf{z}_k^t\|^2 + (1 - s_k^t)\|\mathbf{z}_k^t\|^2\right]. \tag{41}$$

The corresponding local-minimizer perturbation at rejoining satisfies

$$\|\mathbf{w}_{k,b}(\hat{\boldsymbol{\lambda}}_k^b) - \mathbf{w}_{k,b}(\boldsymbol{\lambda}_k^{\circ,b})\|^2 \le \frac{1}{(\alpha - L)^2}\|\hat{\boldsymbol{\lambda}}_k^b - \boldsymbol{\lambda}_k^{\circ,b}\|^2. \tag{42}$$

Proof. Using the virtual-memory update and adding and subtracting $\mathcal{T}_{k,t}(\hat{\boldsymbol{\lambda}}_k^t)$ gives

$$\begin{aligned}\|\hat{\boldsymbol{\lambda}}_k^{t+1} - \boldsymbol{\lambda}_k^{\circ,t+1}\| &\le \|\hat{\mathbf{z}}_k^t - \mathbf{z}_k^t\| + \|\mathcal{T}_{k,t}(\hat{\boldsymbol{\lambda}}_k^t) - \mathcal{T}_{k,t}(\boldsymbol{\lambda}_k^{\circ,t})\| \\ &\le r_k^t + q\|\hat{\boldsymbol{\lambda}}_k^t - \boldsymbol{\lambda}_k^{\circ,t}\|.\end{aligned}$$

Unrolling this recursion from the common initialization in Eq. (39) proves Eq. (40). When $q < 1$, weighted Cauchy–Schwarz yields

$$\left(\sum_{t=a}^{b-1} q^{b-1-t} r_k^t\right)^2 \le \left(\sum_{t=a}^{b-1} q^{b-1-t}\right)\left(\sum_{t=a}^{b-1} q^{b-1-t}(r_k^t)^2\right) \le \frac{1}{1-q}\sum_{t=a}^{b-1} q^{b-1-t}(r_k^t)^2.$$

Equation (23) bounds $(r_k^t)^2$ by the bracketed corrected-energy term, proving Eq. (41). Equation (42) follows from Eq. (36). □

The exact-transition result isolates the effect of topology completion. For an inexact local solver, including finite-step SAM, the same recursion contains an additional local-transition error and remains valid after adding that error to r_k^t.

B.2 Conditional Full-State Stability

The hard zero-row fallback in Eq. (6) of the main paper is discontinuous when a row crosses the threshold. A global Lipschitz claim is therefore not valid without a support-margin condition. We state a conditional result that makes this requirement explicit.

Let the pre-round state be

$$\mathbf{X}^t = (\mathbf{w}_0^t, \hat{\mathbf{\Lambda}}^t, \mathbf{W}^{t-1}), \qquad \hat{\mathbf{\Lambda}}^t = (\hat{\boldsymbol{\lambda}}_1^t, \dots, \hat{\boldsymbol{\lambda}}_N^t),$$

equipped with a weighted product norm

$$\|\mathbf{X}\|_{\mathsf{X}} = \|\mathbf{w}_0\| + c_\lambda \|\hat{\mathbf{\Lambda}}\|_{\mathrm{F}} + c_W \|\mathbf{W}\|_{\mathrm{F}}, \qquad c_\lambda, c_W > 0.$$

The server dual is omitted because it is the average of $\hat{\mathbf{\Lambda}}^t$. For a fixed cohort and coupled local randomness, let $\Psi_t(\mathbf{X}; \mathcal{C})$ denote the complete one-round map consisting of the local solver, graph refresh, diffusion, virtual-memory update, global-dual update, and server-model update.

Assumption 3 (Full-state map regularity)**.** *Consider two coupled runs on neighboring federated datasets $\mathcal{C}$ and $\widehat{\mathcal{C}}$ that differ in one sample and use the same cohorts and random bits. Assume that: (i) the local solver map is Lipschitz in the server model, client dual input, and affected empirical objective; (ii) the stabilized-cosine graph refresh is evaluated with the same $\delta > 0$ and bounded update norms; and (iii) for every normalized row in the two runs, the active-neighbor mass is at least $m_{\min} > 0$, while unsupported rows remain unsupported in both runs. Under these conditions, suppose deterministic constants $a_t, b_t \geq 0$ satisfy the following inequalities uniformly over admissible coupled states and neighboring dataset pairs:*

$$\|\Psi_t(\mathbf{X}; \mathcal{C}) - \Psi_t(\mathbf{X}'; \mathcal{C})\|_{\mathsf{X}} \leq a_t \|\mathbf{X} - \mathbf{X}'\|_{\mathsf{X}}, \tag{43}$$

$$\|\Psi_t(\mathbf{X}; \mathcal{C}) - \Psi_t(\mathbf{X}; \widehat{\mathcal{C}})\|_{\mathsf{X}} \leq b_t. \tag{44}$$

The conditions above apply to the complete state rather than only the server model. The implemented stabilized cosine is Lipschitz on bounded update sets for $\delta > 0$, and row normalization is Lipschitz on masses bounded below by $m_{\min}$. When the perturbation of the affected empirical objective is also bounded, finite a_t and b_t can be obtained by composing the stated local, graph, diffusion, memory, and server maps. The support-margin condition is necessary for the hard fallback used in the implementation.

Theorem 2 (Conditional full-state stability)**.** *Under Assumption 3, let $\Delta_t = \mathbb{E}\|\mathbf{X}^t - \widehat{\mathbf{X}}^t\|_{\mathsf{X}}$. Then*

$$\Delta_{t+1} \leq a_t \Delta_t + b_t, \tag{45}$$

and therefore

$$\Delta_T \leq \Delta_0 \prod_{j=0}^{T-1} a_j + \sum_{t=0}^{T-1} b_t \prod_{j=t+1}^{T-1} a_j. \tag{46}$$

For common initialization, the first term vanishes. If the evaluation loss is G-Lipschitz in the server model, the resulting uniform-stability contribution is at most

$$\epsilon_T \leq G \sum_{t=0}^{T-1} b_t \prod_{j=t+1}^{T-1} a_j. \tag{47}$$

Proof. Add and subtract $\Psi_t(\widehat{\mathbf{X}}^t; \mathcal{C})$:

$$\begin{aligned}\|\mathbf{X}^{t+1} - \widehat{\mathbf{X}}^{t+1}\|_{\mathsf{X}} \leq & \|\Psi_t(\mathbf{X}^t; \mathcal{C}) - \Psi_t(\widehat{\mathbf{X}}^t; \mathcal{C})\|_{\mathsf{X}} \\ & + \|\Psi_t(\widehat{\mathbf{X}}^t; \mathcal{C}) - \Psi_t(\widehat{\mathbf{X}}^t; \widehat{\mathcal{C}})\|_{\mathsf{X}}.\end{aligned}$$

Taking expectation and applying Eqs. (43)–(44) proves Eq. (45). Induction gives Eq. (46). The server-model difference is bounded by the product-state norm, so G-Lipschitzness of the loss and the standard neighboring-dataset coupling give Eq. (47). □

Theorem 2 applies within a fixed graph-support regime and does not cover changes in graph support or threshold crossings. The one-round neighboring-data sensitivity b_t remains explicit because its scaling depends on client dataset sizes, local sampling, and participation of the affected client. Under the fixed-support condition, the bound accounts for perturbations of the learned graph and virtual memories through the complete state map.

C Experimental Details and Additional Results

C.1 Datasets and Client Construction

Table 5 summarizes the benchmark tasks and client construction procedures. CIFAR-10 and CIFAR-100 each contain 50,000 training and 10,000 test images at a resolution of 32×32, with 10 and 100 classes, respectively (Krizhevsky et al., 2009). Tiny-ImageNet contains 200 classes with 100,000 training and 10,000 validation images at 64×64 resolution (Le & Yang, 2015). We use the standard validation split for evaluation. For these vision datasets, we sample class proportions from a Dirichlet distribution and assign the resulting training examples to 500 clients. The main experiments use $\alpha_{\mathrm{Dir}} = 0.6$, while the stronger heterogeneity stress tests use $\alpha_{\mathrm{Dir}} = 0.1$. The centralized test or validation sets are used only for evaluation and are excluded from client training. For Shakespeare, we follow the naturally partitioned LEAF benchmark (Caldas et al., 2018), in which speaking roles define clients. We retain 280 clients with more than 2,000 local sequence examples each and evaluate next-character prediction. Unlike the image benchmarks, Shakespeare is not repartitioned with a synthetic Dirichlet split.

Benchmark	Task	Classes	Resolution / sequence	Clients	Client construction and evaluation split
CIFAR-10	Image classification	10	32×32	500	50,000 training images are assigned by class-wise Dirichlet proportions; the 10,000-image test set is shared for evaluation.
CIFAR-100	Image classification	100	32×32	500	Same protocol as CIFAR-10; $\alpha_{\mathrm{Dir}} \in \{0.1, 0.6\}$.
Tiny-ImageNet	Image classification	200	64×64	500	100,000 training images are Dirichlet-partitioned; the 10,000-image validation split is used for evaluation.
Shakespeare	Next-character prediction	Character vocabulary	Length 80	280	Natural LEAF client identities; clients with more than 2,000 local examples are retained.

Table 5: Benchmark and client-partition details. The Dirichlet parameter controls label heterogeneity only for the vision datasets; smaller α_{Dir} indicates stronger class skew.

C.2 Models, Optimization, and Hyperparameters

For CIFAR-10/100, we use the LeNet-style implementation adopted by the FedDyn-family codebase. For Tiny-ImageNet, we use ResNet-18 with BatchNorm replaced by GroupNorm. For Shakespeare, we use a two-layer LSTM with an eight-dimensional character embedding and a sequence length of 80. The baselines retain their native update rule and optimizer, while RFL-DFDC and the SAM-based baselines use SAM with the same base learning-rate schedule.

The implementation samples each client independently with probability p and resamples only when the resulting cohort is empty. Thus, p is a nominal participation probability and the realized cohort size $P_t = |\mathcal{S}_t|$ may vary across rounds. The default value is $p = 0.02$, while the participation studies use $p \in \{0.005, 0.01, 0.02, 0.05, 0.5\}$. The main vision experiments, component ablations, and graph-source controls use a budget of 1,600 rounds. The main and ablation experiments on CIFAR-100 use a mini-batch size of 50. For the other benchmarks, we use the corresponding released baseline data loaders while retaining five local epochs for each selected client.

The dynamic graph is initialized as an identity matrix, with all off-diagonal similarities set to zero, and is updated only for client pairs that participate in the same round. In the implementation, rows with active-neighbor mass above $\varepsilon_A = 10^{-12}$ are normalized exactly, whereas the remaining rows are set to zero, as specified in Eq. (6).

Item	Default / reported setting	Search range or alternatives
Reported random seeds	$\{20, 21, 22\}$	Fixed across reported three-run summaries
Vision communication rounds	1,600	Same budget for main, participation and ablation studies
Client sampling	Bernoulli, reject empty cohort	$p \in \{0.005, 0.01, 0.02, 0.05, 0.5\}$; default $p = 0.02$
Local epochs	5	Fixed
CIFAR-100 mini-batch size	50	Fixed in main and ablation runs
Initial local learning rate	0.1	Roundwise multiplier 0.998
Weight decay	10^{-3}	Fixed
FedPD penalty α	0.1	$\{0.01, 0.05, 0.1, 0.2, 0.5\}$
SAM radius ρ	0.1	$\{0.01, 0.05, 0.1, 0.2, 0.5\}$
Diffusion coefficient β	0.2	$\{0.05, 0.1, 0.2, 0.5, 1.0\}$
Graph EMA factor γ	0.1	$\{0.01, 0.05, 0.1, 0.2, 0.5\}$
Graph normalization	$\varepsilon_A = 10^{-12}$	Exact normalization above the threshold; zero-row fallback otherwise
Graph representation	Dense $N \times N$ FP32 matrix	No top-k pruning in reported experiments
Virtual-dual representation	Dense FP32 vector per client	No quantization or low-rank compression
Software / accelerator	PyTorch on NVIDIA L40 GPUs	Same accelerator class for timing comparisons

Table 6: Optimization settings, search grids, and implementation details.

C.3 Metrics and Timing

Unless stated otherwise, Acc is the accuracy at the final communication round, reported as mean$\pm$std across three runs with seeds $\{20, 21, 22\}$. Each run contributes one terminal accuracy; the standard deviation summarizes variation across these run-level values. Learning curves show the pointwise mean accuracy across runs. Evaluation uses the CIFAR test split, Tiny-ImageNet validation split, or Shakespeare next-character task.

For target τ, R_τ denotes the first target-reaching round, averaged across runs when all runs reach the target; otherwise the entry is "–". Target attainment records the first crossing even if accuracy subsequently falls below the target. T/R is the mean time per round on the NVIDIA L40 setup. We estimate time to target as

$$\widehat{T}_\tau = R_\tau \times (\mathrm{T/R})/60 \quad \text{minutes.} \tag{48}$$

This estimate multiplies aggregate round counts and round times rather than measuring cumulative elapsed time. Round times, including the large variation across methods on Tiny-ImageNet, depend on the execution setup. We use communication rounds as the primary efficiency measure.

Client sampling across runs. Cohort sampling is reseeded from the round index, run seed, and empty-cohort retry, so adjacent run seeds can produce shifted, overlapping cohort sequences. Standard deviations describe variation across the three runs under this sampling scheme. We do not treat them as uncertainty estimates from independent participation sequences or report independence-based confidence intervals or paired significance tests.

C.4 Observability of the Relationship Graph

The graph begins at the identity and observes only co-participating pairs. Under ideal independent Bernoulli rounds conditioned on a nonempty cohort, a fixed pair is observed with probability $q = p^2/[1-(1-p)^N]$. Its observation probability by round T is $1-(1-q)^T$. For $N = 500, p = 0.02$, this gives about 0.40% after ten rounds and 47.28% after 1600 rounds. These are analytic reference values, not measurements of the archived sampling traces. Pair coverage also differs from the fraction of inactive clients that have any active supported neighbor.

The round-10 frozen control retains the graph from this early observation stage and therefore compares continued learning with a graph built from few observations. It does not test freezing a mature

graph. With a fixed singleton cohort and identity initialization, off-diagonal relationships are never observed and inactive reconstruction remains zero. EMA decay is measured in joint observations of a client pair, with $\gamma = 0.1$ giving a half-life of about 6.58 such observations. Unsupported rows receive a zero update, and their missing increments enter the theoretical error bound.

C.5 Ablation and Control Definitions

Component switches. Table 2 of the main paper uses the clean learned graph and varies only the two correction mechanisms. Without topology correction, the inactive-client memories remain unchanged and $D_i^t = 1$. The reconstruction-only variant updates inactive-client memories using Eq. (7) of the main paper but sets $D_i^t = 1$. The amplification-only variant keeps inactive-client memories fixed but applies Eq. (8) of the main paper. Both single-component variants are diagnostic counterfactuals and do not preserve the exact average-memory identity in Eq. (9) of the main paper. Full RFL-DFDC applies both Eq. (7) and Eq. (8) of the main paper. In all variants, active clients continue to refresh their observed memories according to Eq. (5) of the main paper.

Graph-source controls. Table 3 of the main paper keeps the complete correction path active with $\beta = 0.2$ and varies only the source graph $\widetilde{\mathbf{W}}^t$. Given a source graph, let $\widetilde{m}_k^t = \sum_{j \in \mathcal{S}_t} \widetilde{W}_{kj}^t$. The active-set kernel is

$$A_{ki}^t(\widetilde{\mathbf{W}}) = \begin{cases} \dfrac{\widetilde{W}_{ki}^t}{\widetilde{m}_k^t}, & i \in \mathcal{S}_t \text{ and } \widetilde{m}_k^t > \varepsilon_A, \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

Mean ($\beta = 0.2$) uses $A_{ki}^{t,\mathrm{mean}} = \mathbf{1}\{i \in \mathcal{S}_t\}/P_t$, giving inactive increments $\hat{\mathbf{z}}_k^t = (0.2/P_t)\sum_{i \in \mathcal{S}_t} \mathbf{z}_i^t$ and matching server coefficients $D_i^t = 1 + 0.2(N - P_t)/P_t$. This is the uniform-weight control within RFL-DFDC; the native A-FedPDSAM baseline is reported separately in Table 1.

Last-update cosine caches each client's most recently uploaded model update. For an inactive client k with a cached update and an active client i, it uses $\widetilde{W}_{ki}^t = [1 + \cos_\delta(\Delta\mathbf{w}_k^{\tau_k}, \Delta\mathbf{w}_i^t)]/2$, where $\tau_k < t$ is the last round in which k participated. It uses no EMA of pairwise weights and applies the same active-set normalization and zero-row fallback in Eq. (49); clients with no observed update have zero rows. The resulting kernel is used in both virtual-memory reconstruction and its matching server correction. The random-graph control preserves the number of active neighbors in each row of the clean graph but assigns those neighbors uniformly at random. The shuffled learned graph applies a seed-fixed client permutation Π:

$$\widetilde{\mathbf{W}}^{t,\mathrm{shuf}} = \Pi^\top \mathbf{W}^t \Pi, \tag{50}$$

This transformation preserves the distribution of learned weights while breaking their alignment with client identities. The static learned graph uses the online graph through the warm-up round $t_s = 10$ and then fixes $\widetilde{\mathbf{W}}^t = \mathbf{W}^{t_s}$. The clean dynamic graph uses the EMA-updated $\mathbf{W}^t$ throughout training. These controls compare complete weighting rules and their observation histories, so the difference between last-update cosine and the dynamic graph does not isolate EMA alone.

Refresh-schedule controls. Table 4 uses the clean graph but varies the rounds in which inactive-client reconstruction and amplification are applied. The Stop after 5 and Stop after 10 variants apply the complete correction only during the first 5 or 10 rounds, respectively. Thereafter, inactive-client memories remain unchanged between direct refreshes by active clients. The Update every m variant applies the complete correction only when $t \equiv 0 \pmod m$. The full method applies both correction mechanisms in every round.

D Additional Experimental Results

D.1 Complete Benchmark and Timing Results

Table 7 includes per-round execution time and estimated time to target for the benchmark comparison.

Method	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Acc↑	$R_{75}\downarrow$	T/R↓	$\widehat{T}_\tau\downarrow$	Acc↑	$R_{40}\downarrow$	T/R↓	$\widehat{T}_\tau\downarrow$	Acc↑	$R_{30}\downarrow$	T/R↓	$\widehat{T}_\tau\downarrow$
FedAvg	75.13±0.12	–	2.10	–	28.56±0.04	–	2.95	–	28.31±0.05	1474	24.93	612.4
FedProx	74.78±0.14	–	2.09	–	29.01±0.05	–	2.21	–	28.23±0.06	–	27.95	–
SCAFFOLD	77.90±0.08	677	2.66	30.0	32.21±0.02	–	4.61	–	33.47±0.13	567	77.28	730.3
FedDyn	77.43±0.11	724	2.73	32.9	31.67±0.08	–	3.46	–	25.16±0.07	–	43.53	–
FedDC	78.30±0.16	727	4.28	51.9	39.49±0.04	–	6.22	–	31.54±0.04	654	125.25	1365.2
A-FedPD	77.85±0.43	319	4.44	23.6	38.05±0.16	–	4.11	–	22.38±0.31	–	50.37	–
FedSpeed	81.69±0.08	473	4.75	37.4	42.44±0.02	1149	4.00	76.6	35.24±0.14	858	40.82	583.7
FedSMOO	81.81±0.06	461	5.76	44.3	47.25±0.02	795	5.07	67.2	34.69±0.07	693	63.71	735.9
FedLESAM-D	77.57±0.09	729	4.92	59.8	31.11±0.10	–	7.89	–	34.75±0.13	927	96.71	1494.2
A-FedPDSAM	82.89±0.13	340	5.10	28.9	47.84±0.03	627	4.97	51.9	34.81±0.26	822	141.06	1932.5
FedGloss	77.62±0.13	1078	4.25	76.4	38.23±0.02	–	4.12	–	35.20±0.18	913	52.75	802.7
RFL-DFDC	**84.37**±0.09	**235**	5.01	**19.6**	**51.81**±0.01	**547**	5.05	**46.0**	**39.27**±0.04	**542**	51.69	**466.9**

Table 7: Benchmark accuracy and timing with Dirichlet-0.6, $N = 500$, and nominal $p = 0.02$. Acc is final-round mean±std across runs (%). R_τ is rounds to target, and T/R is seconds per round. $\widehat{T}_\tau = R_\tau(\mathrm{T/R})/60$ estimates time to target in minutes from these averages. "–" indicates that at least one run does not reach the target.

Component	Complexity	Share of extra correction time (%)	Persistent extra memory (MiB)
Graph update	$O(P_t^2 d)$	2.95	–
Virtual-dual reconstruction	$O(NP_t d)$	96.85	–
Topology-aware weighting	$O(NP_t)$	0.20	–
Dense graph storage	$O(N^2)$	–	0.95
Virtual-dual memory	$O(Nd)$	–	1555.12
Total additional state	–	**100.0**	**1556.08**

Table 8: Server-side overhead microbenchmark for CIFAR-100/LeNet with $N = 500$, dense FP32 state, and nominal 2% participation. Time shares are normalized within the additional topology-correction operations, not the full communication round. Memory values use MiB (2^{20} bytes); the virtual-dual memory dominates the persistent state.

D.2 Server-Side Overhead

The virtual dual states dominate both storage and correction time in Table 8. Each client requires a full-dimensional state, making the $O(Nd)$ virtual memory the main scaling cost; the graph itself occupies less than one MiB at this scale. Sparsifying the graph could reduce edge storage and diffusion work, but would leave the virtual-memory cost unchanged. The dense implementation is therefore best suited to moderate-scale, stateful deployments with persistent client identities and a history of co-participation.

D.3 Additional Task and Extreme-Participation Results

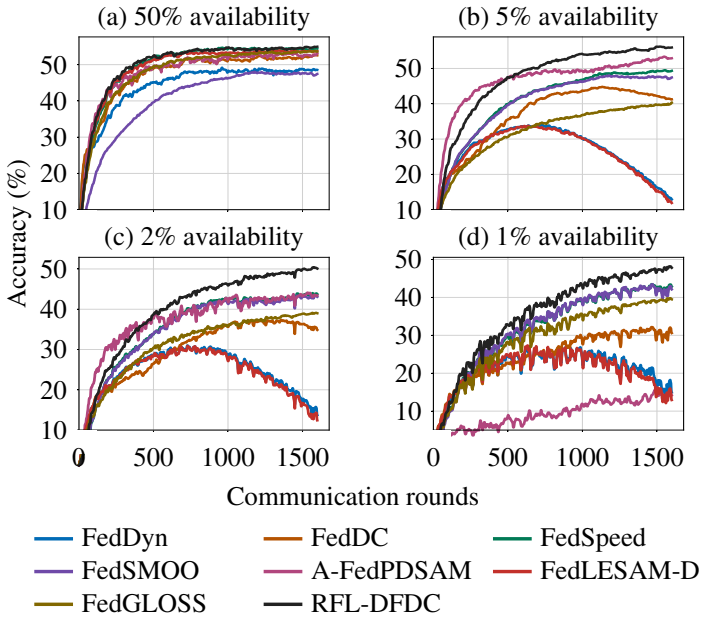
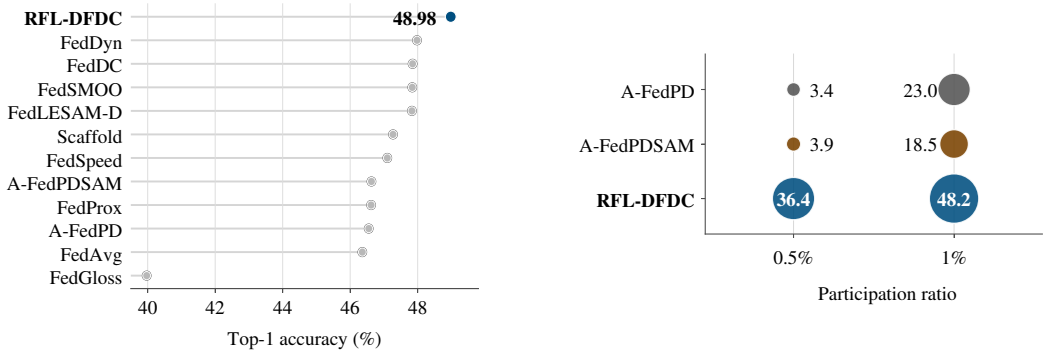


Figure 4: CIFAR-100 accuracy trajectories averaged across runs, with Dirichlet-0.1 at nominal random-participation rates of (a) 50%, (b) 5%, (c) 2%, and (d) 1%. The cohort size varies under Bernoulli sampling.



(a) Shakespeare next-character accuracy.

(b) CIFAR-100 at nominal $p \in \{0.005, 0.01\}$.

Figure 5: Additional sparse-participation results. (a) Points and horizontal error bars report final-round mean±std over three runs on naturally partitioned Shakespeare; RFL-DFDC is highlighted in blue. (b) Bubble area and labels encode final-round Top-1 accuracy averaged over three runs on CIFAR-100 with Dirichlet-0.1.

RFL-DFDC achieves the highest next-character accuracy on naturally partitioned Shakespeare (Figure 5(a)). On CIFAR-100, Figure 5(b) shows a substantial advantage over mean-based compensation at both extreme participation rates. Accuracy remains above 35% even when the nominal participation rate falls to 0.5%. The RFL-DFDC result at 1% corresponds to the trajectory in Figure 4(d).

E Algorithm and Computational Cost

Algorithm 1 RFL-DFDC

Require: Clients $\mathcal{V}$, rounds T, penalty α, scales $\{\chi_t\}$, local solver, EMA factor γ, diffusion rate β, cosine floor δ, row threshold ε_A

Initialize $\mathbf{w}_0^0$, $\hat{\boldsymbol{\lambda}}_k^0 \leftarrow \mathbf{0}$, $\boldsymbol{\lambda}_0^0 \leftarrow N^{-1}\sum_k \hat{\boldsymbol{\lambda}}_k^0$, and $\mathbf{W}^{-1} \leftarrow \mathbf{I}$
for $t = 0$ to $T-1$ **do**
  Sample a nonempty active set $\mathcal{S}_t$ and set $P_t \leftarrow |\mathcal{S}_t|$
  for all $i \in \mathcal{S}_t$ **in parallel do**
    Send $(\mathbf{w}_0^t, \hat{\boldsymbol{\lambda}}_i^t)$; compute $\mathbf{w}_i^{t+1}$ by the local primal solver (SAM in the evaluated implementation)
    $\mathbf{z}_i^t \leftarrow \alpha(\mathbf{w}_0^t - \mathbf{w}_i^{t+1})$; $\boldsymbol{\lambda}_i^{t+1} \leftarrow \hat{\boldsymbol{\lambda}}_i^t + \mathbf{z}_i^t$
    Upload $(\boldsymbol{\lambda}_i^{t+1}, \Delta\mathbf{w}_i^t)$
  end for
  Update $\mathbf{W}^t$ by Eq. (4); form A_{ki}^t by Eq. (6)
  for all $k \in \mathcal{V}$ **do**
    if $k \in \mathcal{S}_t$ **then**
      $\hat{\boldsymbol{\lambda}}_k^{t+1} \leftarrow \boldsymbol{\lambda}_k^{t+1}$
    else
      $\hat{\boldsymbol{\lambda}}_k^{t+1} \leftarrow \hat{\boldsymbol{\lambda}}_k^t + \beta \sum_{i\in\mathcal{S}_t} A_{ki}^t \mathbf{z}_i^t$
    end if
  end for
  Compute D_i^t by Eq. (8) and update $\boldsymbol{\lambda}_0^{t+1}$ by Eq. (9)
  $\bar{\mathbf{w}}_{\mathcal{S}}^{t+1} \leftarrow P_t^{-1}\sum_{i\in\mathcal{S}_t} \mathbf{w}_i^{t+1}$; $\mathbf{w}_0^{t+1} \leftarrow \bar{\mathbf{w}}_{\mathcal{S}}^{t+1} - \chi_t \boldsymbol{\lambda}_0^{t+1}$
end for
return $\mathbf{w}_0^T$

E.1 Complexity and Communication Cost

All additional computation is performed on the server. Updating pairwise similarities among the P_t active clients requires $O(P_t^2 d)$ operations, where d denotes the update dimension. Dense virtual-dual reconstruction requires $O(NP_t d)$ computation per round, while dense graph storage requires $O(N^2)$ memory. Using a sparse top-k graph reduces the edge-storage cost to $O(kN)$ and the diffusion cost to $O(kNd)$. Storing the N full-dimensional virtual dual states requires $O(Nd)$ memory, and computing the influence weights costs $O(NP_t)$. RFL-DFDC communicates the same model and dual-update tensors as the underlying primal–dual method. Its main trade-off is the additional server state and graph-diffusion computation.