FEDGAUG: DIAGNOSING LATE-ONSET INSTABILITY IN GRAPH-AUGMENTED FEDERATED CLINICAL PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) lets hospitals train clinical prediction models together without pooling their electronic health records (EHRs). When patient populations differ between sites, however, the aggregated global model can break down after many communication rounds. We present FedGAug, a federated clinical prediction system that pairs a parameter-efficient transformer with a per-patient knowledge graph encoder, where each local graph is enriched with entities retrieved from external biomedical sources. We train it on three non-IID clients with three random seeds each, using asynchronous, staleness-weighted FedAvg. Without regularization, the global model improves steadily to a mean AUROC of 0.907 at round 12 and shows no forgetting up to that point. It then collapses to 0.370 by round 15 and stays there until round 20. Evaluating the local and global models separately shows that the local models hold up much better (AUROC around 0.71 against 0.37 for the global model during the collapse), which points to aggregation rather than local overfitting as the source of the failure. We then test Elastic Weight Consolidation (EWC) and Knowledge Distillation (KD) as remedies. None of the regularized variants collapses. Over 29 rounds their peak-to-final forgetting stays at or below 0.005, compared with 0.169 for the unregularized model, and EWC with $\lambda = 0.5$ still reaches a mean AUROC of 0.856 at round 21, when the unregularized model is at 0.385. Our results show that late-onset degradation in non-IID federated clinical prediction can originate at the aggregation step, and that continual-learning regularization prevents it over long training runs.

1 INTRODUCTION

Electronic Health Records (EHRs) contain diagnosis codes, procedures, and medication orders that can be used to predict in-hospital mortality, length of stay, and readmission early. Pooling these records across a healthcare network is difficult, though, because regulations such as HIPAA and GDPR restrict how patient data can be shared (Nikolovski & Petreska, 2024).

Federated Learning (FL) addresses this by letting hospitals and clinics train a shared model while keeping their data on site (McMahan et al., 2017; Rieke et al., 2020). Only model updates are sent to a central server for aggregation. Clinical deployments still face two practical problems. First, clinical data are rarely identically distributed. Sites differ in demographics, disease prevalence, coding habits, and admission criteria, which leads to feature and label heterogeneity (Zhao et al., 2018). Second, sites have different hardware, so some clients finish their local updates much later than others. This pushes deployments toward asynchronous or staleness-weighted aggregation (Xie et al., 2019).

Both problems contribute to client drift. Each client moves toward a solution that fits its own data, and averaging these diverging solutions gives a global model that is less accurate and slower to converge (Karimireddy et al., 2020; McCloskey & Cohen, 1989; Kirkpatrick et al., 2017; Shoham et al., 2019).

Client drift has been studied extensively in computer vision and natural language processing (Karimireddy et al., 2020; Li et al., 2020; Reddi et al., 2021; Wang et al., 2020). Much less is known about

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.

how it behaves in parameter-efficient, graph-augmented clinical models. Most FL studies also train for only 5 to 10 rounds, which is too short to see failures that appear later in training (Kairouz et al., 2021).

In this paper we study how federated clinical models degrade over longer training. We introduce FedGAug (Federated Graph Augmentation), which combines a LoRA-adapted transformer (Hu et al., 2022; Sanh et al., 2019) with a per-patient knowledge graph (KG) processed by a graph encoder (Kipf & Welling, 2017; Choi et al., 2017), following the approach of GraphCare (Jiang et al., 2024).

We ask whether, in this kind of system trained on non-IID cohorts with asynchronous FedAvg, degradation happens gradually or as a sudden late event, and whether continual-learning regularizers can prevent it without stopping the clients from adapting to their own data.

We test two hypotheses. The first, *post-aggregation parameter interference*, is that the late drop in performance is caused mainly by the server averaging step and not by clients overfitting their local data. When client data differ, their updates can conflict, and combining them damages the global model. The second, *curvature-weighted stabilization*, is that penalizing local parameters for moving away from the global model, weighted by the curvature of the loss, prevents this collapse over many rounds while keeping the model's ability to rank patients, measured by AUROC and AUPRC.

We evaluate FedGAug on three non-IID clients with three random seeds per client and address four research questions:

- **RQ1 (Onset timing):** When does global model degradation appear, and would a standard short training run detect it?
- **RQ2 (Failure localization):** Is the degradation caused by local overfitting or by interference between client updates at the server?
- **RQ3 (Mitigation):** Can Elastic Weight Consolidation (EWC) and KD prevent the late collapse, and how does regularization strength interact with differences in client size?
- **RQ4 (Monitoring):** Which training signals give reliable early warning of global model failure when classes are imbalanced?

2 Related Work

2.1 Federated Learning in Healthcare

Hospitals rarely share raw patient data, and FL has become a common way to build models together in radiology, genomics, and structured EHR analysis (Noor et al., 2026; Nguyen et al., 2025). Multi-center studies on tumor segmentation have shown that federated averaging can work at scale (Nguyen et al., 2025). The harder problem is that clinical data are almost never IID. Disease prevalence varies between sites, which creates label skew, and each institution codes its records differently, which creates feature skew (Martinez, 2026). Shi et al. (2023) showed that this heterogeneity can collapse the global model's representations into a low-dimensional subspace, and Nguyen et al. (2023) found that starting from a pre-trained model closes much of the gap between IID and non-IID training. These findings motivate our use of a pre-trained backbone.

2.2 Client Drift in Heterogeneous and Asynchronous FL

When clients run several local epochs on skewed data, their parameters move away from the global solution. Recent methods address this on the client side and on the server side. FedImpro (Tang et al., 2024) reshapes local feature distributions so that client gradients agree more closely, and FedSMOO (Sun et al., 2023) and FedLESAM (Fan et al., 2024) use sharpness-aware optimization to steer local updates toward flat regions that also suit the global objective. On the server, FedDisco (Ye et al., 2023) weights clients by how far their label distribution is from the global one, and FedALA (Zhang et al., 2023) learns how much of the global model each client should adopt. Asynchronous training adds another difficulty, because the server aggregates updates computed from outdated models. CA2FL (Wang et al., 2024b) corrects for this by caching and reusing earlier client updates, and FADAS (Wang et al., 2024c) adapts adaptive server optimizers to the asynchronous

setting. Almost all of these methods were evaluated on vision benchmarks with full model training, so their behavior over long runs with only a small set of adapter parameters is largely unknown.

2.3 Continual Learning Regularization

A model trained on tasks in sequence tends to overwrite what it learned earlier, a problem known as catastrophic forgetting (Wang et al., 2024a). Two families of regularizers are widely used against it. EWC estimates the importance of each parameter from the Fisher information and penalizes changes to important parameters. Knowledge distillation keeps the model's outputs close to those of a frozen teacher. Parameter-efficient tuning also helps somewhat on its own, since Biderman et al. (2024) found that LoRA forgets less than full fine-tuning, although it also learns less. Both families have been adapted to FL. GradMA (Luo et al., 2023) stores past gradients to correct local updates, Federated Orthogonal Training (Bakman et al., 2024) keeps new updates orthogonal to the subspace that holds earlier global knowledge, FedTGP (Zhang et al., 2024) ties heterogeneous clients to trainable global prototypes, and FedDAT (Chen et al., 2024) distills knowledge in both directions between a frozen and a trainable adapter. We use these ideas differently. Instead of protecting knowledge from earlier tasks, we use EWC and KD to keep each client's local update close to the current global model.

2.4 Parameter-Efficient Fine-Tuning in Clinical Domains

Fine-tuning all parameters of a large transformer requires a lot of memory and bandwidth, and few clinical sites have the hardware for it. Low-rank adaptation (LoRA) freezes the pre-trained weights and learns small low-rank updates to the attention layers instead. DoRA (Liu et al., 2024) separates the magnitude and direction of the weight update, and LoRA+ (Hayou et al., 2024) uses different learning rates for the two low-rank factors. In FL, sending only adapter weights reduces communication considerably, but averaging low-rank factors across clients is not equivalent to averaging full weights. FFA-LoRA (Sun et al., 2024) freezes one of the two factors to keep aggregation stable, and FedSA-LoRA (Guo et al., 2025) found that one factor mostly captures general knowledge and the other client-specific knowledge, so only the first needs to be shared. How such a constrained parameter space behaves under asynchronous aggregation is still an open question, and it is the question we study here. On the clinical side, GraphCare (Jiang et al., 2024) builds a knowledge graph for each patient to support EHR prediction, and KARE (Jiang et al., 2025) improves LLM-based clinical prediction by retrieving communities from a medical knowledge graph. We deliberately use a general-purpose backbone, DistilBERT, rather than a biomedical encoder. Appendix A gives a detailed comparison with these works.

3 Methodology

We treat federated clinical prediction as an asynchronous, regularized, non-convex optimization problem over non-IID clients. Each client builds a knowledge graph for every patient and enriches it with entities retrieved from public biomedical sources (Section 3.2). Each client then trains a LoRA and GNN model with EWC and KD regularization and exchanges only parameter updates with the server.

3.1 Problem Formulation

Let K be the number of clients. Client $k \in \{1, \dots, K\}$ holds a private dataset $\mathcal{D}_k$ of size $N_k = |\mathcal{D}_k|$, drawn from a client-specific distribution P_k over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $x \in \mathcal{X}$ is a clinical record and $y \in \{0, 1\}$ is the mortality outcome. With $N = \sum_{k=1}^{K} N_k$, we seek global parameters $\theta^* \in \Theta \subseteq \mathbb{R}^d$ that minimize the population risk:

$$\min_{\theta \in \Theta} \mathcal{F}(\theta) \coloneqq \sum_{k=1}^{K} p_k f_k(\theta), \quad \text{where } p_k = \frac{N_k}{N}, \quad f_k(\theta) \coloneqq \mathbb{E}_{(x,y)\sim P_k}\left[\ell(\hat{y}_\theta(x), y)\right], \tag{1}$$

where f_k is the local risk and ℓ is the binary cross-entropy loss.

3.2 GRAPH-AUGMENTED ENCODER

Each patient record $x = (S, \mathcal{G})$ consists of an ordered sequence of clinical codes $S = (c_1, c_2, \ldots, c_M)$ and a relational knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The nodes are the patient's diagnosis, procedure, and medication codes, and edges connect codes that occur in the same admission, as in GraphCare (Jiang et al., 2024; Murali et al., 2025).

Retrieval-augmented graph construction. The graph is not limited to codes in the patient's own record. Each client also runs similarity queries with its local entity embeddings against an external biomedical Linked Open Data source, retrieves entities and relations linked to the patient's codes, and adds these triples to $\mathcal{G}$ before the GNN processes it. Only entity identifiers and queries leave the client. Patient records are never sent.

Parameter-efficient transformer (LoRA). Let $W_0 \in \mathbb{R}^{d\times k}$ be a frozen pre-trained projection matrix in a multi-head attention layer. Following Hu et al. (2022), we learn a low-rank update:

$$W = W_0 + \Delta W = W_0 + \frac{\alpha_{\text{LoRA}}}{r} BA, \quad \text{where } B \in \mathbb{R}^{d\times r},\ A \in \mathbb{R}^{r\times k},\ \text{with } r \ll \min(d, k), \tag{2}$$

and α_{LoRA} is a constant scaling factor. The sequence representation $\mathbf{h}_{\text{seq}} \in \mathbb{R}^{d_{\text{model}}}$ is the final hidden state of a frozen DistilBERT backbone (Sanh et al., 2019):

$$\mathbf{h}_{\text{seq}} = \text{Transformer}_{\theta_{\text{adapter}}}(S). \tag{3}$$

Graph encoder. The graph structure captures co-occurrence and temporal relations between medical codes, in the spirit of graph convolutional networks (Kipf & Welling, 2017) and earlier code-graph models for clinical prediction (Choi et al., 2017; Jiang et al., 2024). Node embeddings are updated over L message-passing layers with mean aggregation, and a mean readout gives the graph embedding $\mathbf{h}_{\mathcal{G}}$. We concatenate $\mathbf{h}_{\mathcal{G}}$ with $\mathbf{h}_{\text{seq}}$, pass the result through a linear layer with ReLU to obtain $\mathbf{h}_{\text{patient}}$, and apply a sigmoid output layer to get $\hat{y}_\theta(x) = \sigma(\mathbf{z}_\theta(x))$, where $\mathbf{z}_\theta(x)$ is the logit. Appendix E gives the full equations. The trainable parameters exchanged during training are $\theta := \{\theta_{\text{adapter}}, \theta_{\text{GNN}}, \theta_{\text{fuse}}\} \in \mathbb{R}^d$. The frozen backbone is not part of θ and is not regularized.

3.3 REGULARIZED LOCAL TRAINING

When client k receives the global parameters issued at server step $t_{\text{issued},k}$, it uses them as an anchor, $\theta^\star := \theta^{(t_{\text{issued},k})}$, and minimizes a regularized loss around this anchor.

Task loss. For a local batch $\mathcal{B}_k \subset \mathcal{D}_k$, the task loss is the binary cross-entropy:

$$\mathcal{L}_{\text{task}}(\theta_k; \mathcal{B}_k) = -\frac{1}{|\mathcal{B}_k|} \sum_{(x,y)\in\mathcal{B}_k} \left[y \log \hat{y}_{\theta_k}(x) + (1-y) \log(1 - \hat{y}_{\theta_k}(x)) \right]. \tag{4}$$

Curvature-weighted penalty (EWC). Following Kirkpatrick et al. (2017), we use a quadratic penalty weighted by the diagonal of a curvature matrix F_k. For each parameter $i \in \{1, \ldots, d\}$, we estimate $F_{k,i}$ from the squared gradient of the model's log-probability for the positive class at the anchor:

$$F_{k,i} = \frac{1}{|\mathcal{D}_k^{\text{val}}|} \sum_{x\in\mathcal{D}_k^{\text{val}}} \left(\frac{\partial \log \sigma(\mathbf{z}_\theta(x))}{\partial \theta_i}\bigg|_{\theta=\theta^\star} \right)^2. \tag{5}$$

This is a label-free approximation of the diagonal Fisher information. Unlike the empirical Fisher, it does not use the observed labels, so it measures how sensitive the model's own predictions are to each parameter. With $z = \mathbf{z}_\theta(x)$, the chain rule gives $\partial \log \sigma(z)/\partial\theta_i = (1 - \sigma(z))\, \partial z/\partial\theta_i$, which is non-zero whenever $\sigma(z) \neq 1$. The penalty is

$$\mathcal{R}_{\text{EWC}}(\theta_k; \theta^\star, F_k) = \frac{1}{2} \sum_{i=1}^{d} F_{k,i} (\theta_{k,i} - \theta_i^\star)^2. \tag{6}$$

Knowledge distillation penalty (KD). EWC keeps the parameters close to the anchor, but it does not directly constrain the predictions. We therefore add a distillation term (Hinton et al., 2015) that uses the anchor model as a teacher. For each example in the batch, we soften the predicted probabilities of both models with a temperature $T_{\text{distill}} \geq 1$, using $\sigma(\mathbf{z}_\theta(x)/T_{\text{distill}})$, and compute the KL divergence between the resulting Bernoulli distributions of the local model and the anchor. $\mathcal{R}_{\text{KD}}(\theta_k; \theta^\star, \mathcal{B}_k)$ is the batch average of this divergence, multiplied by T^2_{distill} as in Hinton et al. (2015).

Local objective. Client k minimizes

$$\mathcal{L}_{\text{local}}(\theta_k) = \mathcal{L}_{\text{task}}(\theta_k; \mathcal{B}_k) + \lambda_{\text{EWC}}\mathcal{R}_{\text{EWC}}(\theta_k; \theta^\star, F_k) + \lambda_{\text{KD}}\mathcal{R}_{\text{KD}}(\theta_k; \theta^\star, \mathcal{B}_k). \tag{7}$$

This objective is close to FedCurv (Shoham et al., 2019), which also adds a Fisher-weighted penalty toward a shared optimum, and to FedNTD (Lee et al., 2022), which distills from the global model. Our contribution is therefore not a new regularizer but a study of how it behaves over long training in a graph-augmented clinical setting.

3.4 Asynchronous Aggregation

Let t be the server clock, which advances by one each time the server merges an update. Client k trains for E local epochs starting from $\theta^{(t_{\text{issued},k})}$ and returns θ_k^{local}. The staleness of this update is $\tau = t - t_{\text{issued},k}$. Following FedAsync (Xie et al., 2019), the server scales the merge by a hyperbolic staleness discount $s(\tau)$:

$$\theta^{(t+1)} = (1 - \gamma_0\, s(\tau))\, \theta^{(t)} + \gamma_0\, s(\tau)\, \theta_k^{\text{local}}, \quad \text{where } s(\tau) = (1 + b\tau)^{-1}, \tag{8}$$

where $b > 0$ controls how fast the discount decays and $\gamma_0 \in (0, 1]$ is the base mixing rate. The merge does not weight updates by client dataset size, so every update counts equally regardless of how much data the client holds. We count one communication round as K server steps, so that each client contributes one update per round on average. Algorithm 1 in Appendix C gives the full protocol.

4 Experimental Setup

4.1 Data and Client Partitioning

We use structured EHRs generated with the Synthea simulator (Walonoski et al., 2018). The task is predicting in-hospital mortality ($y \in \{0, 1\}$). To simulate differences between institutions, we split the data across three clients with a Dirichlet distribution $\text{Dir}(\alpha_{\text{Dir}})$ over label prevalence, with $\alpha_{\text{Dir}} = 1.0$, following common practice in FL (Zhao et al., 2018). Client 3 is deliberately smaller and has a lower prevalence, to represent a clinic with limited resources. Table 1 shows the resulting split.

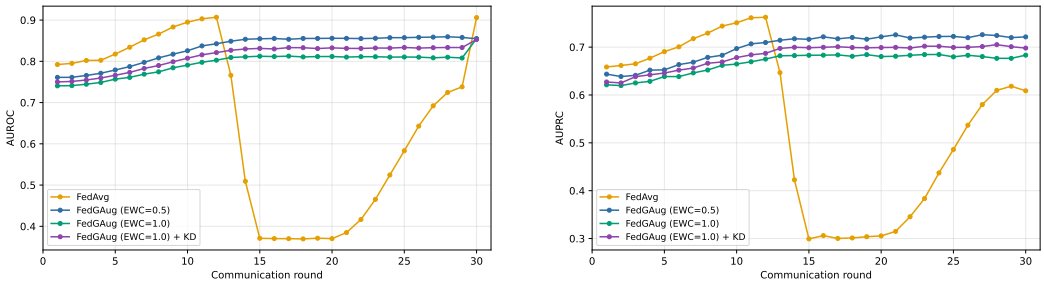
Table 1: Sample size and mortality prevalence for each client.

Client	Patient Records	Positive Labels (Mortality)	Prevalence (%)
Client 1	14,250	1,795	12.6
Client 2	11,800	1,510	12.8
Client 3 (resource-constrained)	2,150	180	8.4

4.2 Configurations

We compare four configurations on the same client splits, with three random seeds per client. All of them use the asynchronous aggregation rule in Eq. 8.

1. **FedGAug (unregularized):** the same sequence and graph encoder with $\lambda_{\text{EWC}} = 0$ and $\lambda_{\text{KD}} = 0$. We use this name rather than "FedAvg" because it differs from the other three configurations only in the absence of regularization, and it keeps the graph encoder and asynchronous aggregation.



(a) Global AUROC

(b) Global AUPRC

Figure 1: Global AUROC and AUPRC by communication round, averaged over clients and seeds.

2. **FedGAAug (EWC $\lambda = 0.5$):** moderate curvature-weighted regularization (Kirkpatrick et al., 2017), with $\lambda_{\text{EWC}} = 0.5$ and $\lambda_{\text{KD}} = 0$.
3. **FedGAAug (EWC $\lambda = 1.0$):** strong curvature-weighted regularization, with $\lambda_{\text{EWC}} = 1.0$ and $\lambda_{\text{KD}} = 0$.
4. **FedGAAug (EWC 1.0 + KD):** strong EWC combined with distillation (Hinton et al., 2015), with $\lambda_{\text{EWC}} = 1.0$ and $\lambda_{\text{KD}} = 1.0$.

4.3 Implementation Details

Local models are trained with AdamW, a learning rate of 2×10^{-4}, and a weight decay of 0.01. Each client runs $E = 3$ local epochs per round with a batch size of 32, for 29 communication rounds. The server runs one thread per client and merges updates in the order they arrive. A lock on the global model serializes concurrent arrivals, so two updates that arrive at the same time are merged one after the other. No client selection is performed, and staleness arises only from differences in client speed. All runs use the hyperbolic staleness discount of Eq. 8 with the values in Table 6. Staleness is not capped. Every configuration is trained with three random seeds per client. Unless stated otherwise, we average over seeds within each client, then average across clients, and report the standard deviation across the three clients. With three clients and three seeds per client, our sample is small. We therefore limit our comparisons to whether a configuration collapses or stays stable, and we do not make fine-grained claims about which regularizer or regularization strength is best. Table 6 in Appendix D lists all hyperparameters.

5 Results

5.1 RQ1: Late-Onset Instability

We focus first on AUROC and AUPRC, since these rank-based metrics remain reliable when classes are imbalanced.

Figure 1 shows the global model after each aggregation. Client drift is usually expected to cause a gradual decline. Without regularization, we instead see a sudden collapse late in training. During rounds 1 to 12, AUROC improves steadily and peaks at 0.907 in round 12. It then drops to 0.766 in round 13 and 0.509 in round 14, stays at 0.370 from round 15 to round 20, and partly recovers to 0.738 by round 29. AUPRC follows the same pattern and falls from 0.762 to 0.299.

Table 2 and Figure 2 show why training length matters. Over rounds 1 to 5 or 1 to 12, the unregularized model shows no forgetting at all. Over rounds 1 to 17, forgetting rises to 0.537 ± 0.057. A typical FL evaluation of 5 to 10 rounds (Kairouz et al., 2021) would miss this failure completely.

5.2 RQ2: Is the Failure Local or Caused by Aggregation?

To test the first hypothesis, we evaluate each client's model right after local training (θ_{local}) and the global model right after aggregation (θ_{global}) on the same fixed validation set. Without regularization,

Table 2: Peak-to-final forgetting of the global model (peak AUROC minus final AUROC) over windows of increasing length. Each client's trajectory is first averaged over its three seeds, and values are the mean and standard deviation across clients. n is the number of clients that reached the window.

Window	Method	n	Peak AUROC	Forgetting
Rounds 1 to 5	FedGAug	3	0.818	0.000 ± 0.000
	FedGAug (EWC 0.5)	3	0.779	0.000 ± 0.000
	FedGAug (EWC 1.0)	3	0.757	0.000 ± 0.000
	FedGAug (EWC 1.0 + KD)	3	0.766	0.000 ± 0.000
Rounds 1 to 12	FedGAug	3	0.907	0.000 ± 0.000
	FedGAug (EWC 0.5)	3	0.843	0.000 ± 0.000
	FedGAug (EWC 1.0)	3	0.803	0.000 ± 0.000
	FedGAug (EWC 1.0 + KD)	3	0.821	0.000 ± 0.000
Rounds 1 to 17	FedGAug	3	0.907	0.537 ± 0.057
	FedGAug (EWC 0.5)	3	0.855	0.002 ± 0.001
	FedGAug (EWC 1.0)	3	0.813	0.001 ± 0.001
	FedGAug (EWC 1.0 + KD)	3	0.833	0.000 ± 0.000
Rounds 1 to 29	FedGAug	3	0.907	0.169 ± 0.096
	FedGAug (EWC 0.5)	3	0.860	0.002 ± 0.001
	FedGAug (EWC 1.0)	3	0.813	0.005 ± 0.003
	FedGAug (EWC 1.0 + KD)	3	0.835	0.001 ± 0.001

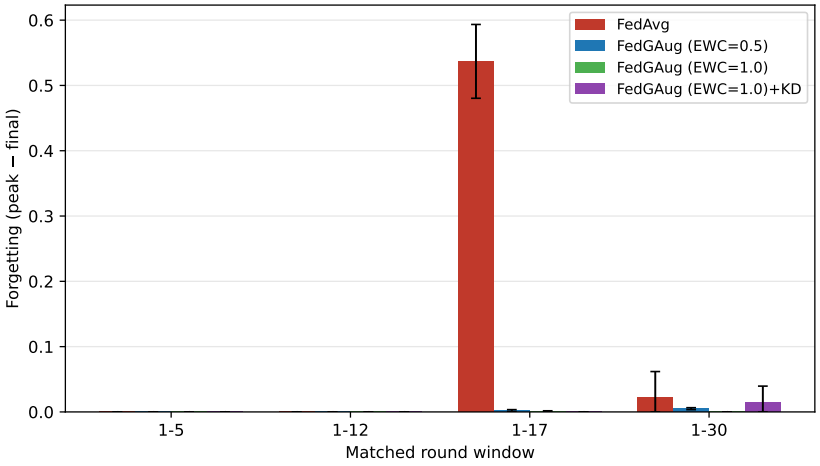


Figure 2: Peak-to-final forgetting of the global model over windows of increasing length (mean and standard deviation across clients).

both drop together when the collapse begins (round 14: 0.526 local, 0.509 global), but they separate clearly afterwards. From round 15 to round 20, the local models stay at an AUROC of about 0.71 while the global model stays at 0.370. By round 29 the local models reach 0.990 and the global model 0.738 (Figure 3). The regularized variants behave very differently. For all three, the gap between local and global AUROC stays between 0.008 and 0.022 in every round, so aggregation neither harms nor noticeably improves on the local models.

Because the local models keep working while the global model fails, the loss of discrimination appears to arise when the updates are combined, not during local training. Let $\Delta_k = \theta_k^{\text{local}} - \theta^{(t_{\text{issued},k})}$ be client k's update. By Eq. 8, each merge moves the global model by $\gamma_0\, s(\tau)\, \Delta_k$, so over one round

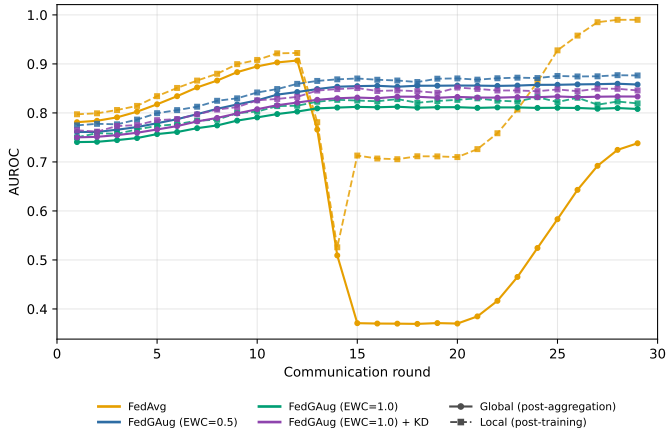


Figure 3: AUROC of the local models after training (dashed lines, squares) and of the global model after aggregation (solid lines, circles), averaged over clients and seeds. For the unregularized model the two diverge sharply after round 14. Under all FedGAug variants they stay within 0.022 of each other.

the global model moves roughly by the sum $\sum_k \gamma_0 \, s(\tau_k) \, \Delta_k$. When clients train on different data, their updates can point in opposing directions, that is, $\langle \Delta_k, \Delta_j \rangle < 0$ for $k \neq j$. In that case the combined step can increase the loss of the global model even though each update lowers the loss of its own client. We did not measure these inner products directly, so the gap between local and global performance is consistent with this explanation but does not prove it.

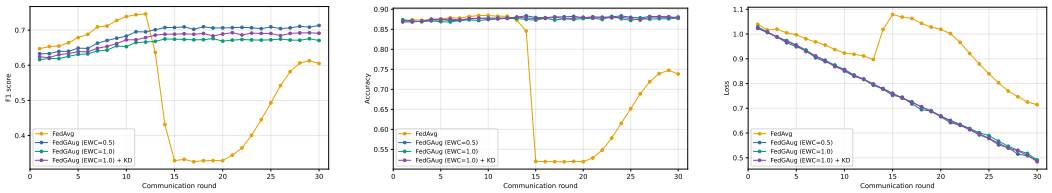
5.3 RQ3: Does Regularization Prevent the Collapse?

Regularization prevents the collapse over the full 29 rounds, which supports the second hypothesis. None of the three regularized variants shows a late drop (Figure 1a). Their forgetting stays at or below 0.005 in every window. Over 29 rounds it is 0.002 ± 0.001 for EWC $\lambda = 0.5$, 0.005 ± 0.003 for EWC $\lambda = 1.0$, and 0.001 ± 0.001 for EWC 1.0 + KD, against 0.169 ± 0.096 for the unregularized model (Table 2). The differences between the regularized variants are within one standard deviation, so we do not claim that KD adds stability beyond EWC.

This stability has a cost. The unregularized model reaches the highest peak AUROC (0.907), while the regularized models level off lower, and the stronger the regularization, the lower the plateau: 0.860 for EWC $\lambda = 0.5$, 0.835 for EWC 1.0 + KD, and 0.813 for EWC $\lambda = 1.0$. At the final round, however, every regularized variant is better than the unregularized model (Table 3). The per-client results also show how regularization interacts with client size. Under all FedGAug variants, the small, low-prevalence Client 3 has the lowest AUROC. Without regularization, it is the only client that recovers from the collapse (0.859, against 0.672 and 0.684 for the other two). We do not have a firm explanation for this. Because the merge does not weight updates by dataset size, Client 3's updates count as much as those of the two larger clients each time they arrive, even though it holds only about 8% of the data. How this interacts with its lower prevalence is a question we leave to future work.

Table 3: Global model AUROC for each client at the final round (round 29). Client columns show the mean and standard deviation over three seeds. The last column shows the mean and standard deviation across clients.

Method	**Client 1**	**Client 2**	**Client 3**	**Mean**
FedGAug	0.672 ± 0.002	0.684 ± 0.001	0.859 ± 0.003	0.738 ± 0.105
FedGAug (EWC 0.5)	0.880 ± 0.013	0.869 ± 0.017	0.824 ± 0.021	0.858 ± 0.030
FedGAug (EWC 1.0)	0.828 ± 0.021	0.813 ± 0.024	0.783 ± 0.030	0.808 ± 0.023
FedGAug (EWC 1.0 + KD)	0.853 ± 0.014	0.845 ± 0.020	0.803 ± 0.024	0.833 ± 0.027



(a) Global F1 (b) Global accuracy (c) Global training loss

Figure 4: Global F1, accuracy, and training loss by communication round.

5.4 RQ4: Which Training Signals Can Be Trusted?

We next check whether common training signals reflect the health of the global model when fewer than 13% of patients are positive.

Figures 4b and 4c show two ways these signals mislead:

1. **Training loss.** Without regularization, the training loss reaches its lowest value (0.897) in round 13, the same round in which global AUROC has already dropped from 0.907 to 0.766. The loss only rises one round later. For the regularized variants, the loss keeps falling until round 29, although AUROC stops improving around round 14. A decreasing loss therefore does not mean that discrimination is still improving.
2. **Accuracy.** Accuracy stays high while the collapse is starting (0.873 in round 13 and 0.846 in round 14, when AUROC is already 0.509), because most patients belong to the negative class. It only falls to 0.520 in round 15.

Federated clinical systems should therefore track threshold-independent metrics such as AUROC and AUPRC on the global model after each aggregation.

6 Conclusion

We studied late-onset degradation in parameter-efficient, graph-augmented federated clinical models. Local training remains largely stable, and the failure appears when updates from non-IID clients are aggregated. Continual-learning regularization prevents the late collapse, at the cost of a lower peak AUROC that decreases as regularization gets stronger. Training loss and accuracy are unreliable indicators of model health under class imbalance, so the global model should be monitored with threshold-independent metrics after every aggregation.

AI use statement

Generative AI tools were used for editing, LaTeX formatting, and improving the structure of the text.

Ethics statement

This study uses synthetic EHRs generated with the open-source Synthea framework (Walonoski et al., 2018). The data mimic real clinical distributions but contain no information about real patients, so the work raises no privacy or human-subjects concerns.

Reproducibility statement

Section 3 and Section 4 describe the model architecture and evaluation protocol. The source code, data generation scripts, and evaluation scripts are attached as supplemantry material to preserve anonymity.

REFERENCES

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Yahya H. Ezzeldin, and Salman Avestimehr. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2309.01289.

Dan Biderman et al. Lora learns less and forgets less. *Transactions on Machine Learning Research (TMLR)*, 2024. doi: 10.48550/arXiv.2405.09673.

Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11285–11293, 2024. doi: 10.1609/aaai.v38i10.29007.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: Graph-based attention model for healthcare representation learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 787–795, 2017. doi: 10.1145/3097983.3098126. DOI: https://doi.org/10.1145/3097983.3098126.

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 *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 12858–12881, 2024. doi: 10.48550/arXiv.2405.18890.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.01463.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2402.12354.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. doi: 10.48550/arXiv.1503.02531. DOI: https://doi.org/10.48550/arXiv.1503.02531.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2106.09685. DOI: https://doi.org/10.48550/arXiv.2106.09685.

Pengcheng Jiang, Cao Xiao, Adam Cross, and Jimeng Sun. Graphcare: Enhancing healthcare predictions with personalized knowledge graphs. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2305.12788.

Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. In *International Conference on Learning Representations (ICLR)*, 2025. doi: 10.48550/arXiv.2410.04585.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. DOI: https://doi.org/10.1561/2200000083.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, pp. 5132–5143, 2020. doi: 10.48550/arXiv.1910.06378. PDF: http://proceedings.mlr.press/v119/karimireddy20a/karimireddy20a.pdf.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. doi: 10.48550/arXiv.1609.02907. DOI: https://doi.org/10.48550/arXiv.1609.02907.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al.

Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. DOI: https://doi.org/10.1073/pnas.1611835114.

Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. doi: 10.5555/3600270.3603057. DOI: https://doi.org/10.5555/3600270.3603057.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems (MLSys)*, 2020. doi: 10.48550/arXiv.1812.06127. PDF: https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf,.

Shih-Yang Liu et al. Dora: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2402.09353.

Kangyang Luo, Xiang Li, Yunshi Lan, and Ming Gao. Gradma: A gradient-memory-based accelerated federated learning with alleviated catastrophic forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3708–3717, 2023. doi: 10.48550/arXiv.2302.14307.

Sophia Martinez Sophia Martinez. Autonomous healthcare analytics via graph neural networks and generative ai. *The American Online Journal of Science and Engineering (AOJSE)*, 4(02), 2026. URL https://aojse.org/index.php/aojse/article/view/51.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989. doi: 10.1016/S0079-7421(08)60536-8. DOI: https://doi.org/10.1016/S0079-7421(08)60536-8.

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 (AISTATS)*, pp. 1273–1282, 2017. doi: 10.48550/arXiv.1602.05629. PDF: https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf.

Lino Murali, G Gopakumar, Daleesha M Viswanathan, Raghu Raman, and Prema Nedungadi. Integrating llms and knowledge graphs for medical ai: Advances, challenges, and future directions. *IEEE journal of biomedical and health informatics*, 2025. doi: 10.1109/JBHI.2025.3622058.

D.-T. Nguyen, M. Imami, L.-M. Zhao, J. Wu, A. Borhani, A. Mohseni, M. Khunte, Z. Zhong, V. Shi, S. Yao, Y. Wang, N. Loizou, A.C. Silva, P.J. Zhang, Z. Zhang, Z. Jiao, I. Kamel, W.-H. Liao, and H. Bai. Federated learning for renal tumor segmentation and classification on multi-center MRI dataset. *Journal of Magnetic Resonance Imaging*, 2025. doi: 10.1002/jmri.29819.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2206.15387.

Saso Nikolovski and Anita Petreska. Evaluating gdpr and hipaa in the integration of ml/ai for future-proofing healthcare. In *Serbian International Conference on Applied Artificial Intelligence*, pp. 289–313. Springer, 2024. doi: 10.1007/978-3-031-99201-8_30.

Abtahi Noor, Ummay Maimona Chaman, and Md Saif Kabir. Federated learning in healthcare: A comprehensive survey on privacy, scalability and clinical applications. *ICT Express*, 12(4): 825–839, 2026. doi: 10.1016/j.icte.2026.05.011.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021. doi: 10.48550/arXiv.2003.00295. DOI: https://doi.org/10.48550/arXiv.2003.00295.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *npj Digital Medicine*, 3(1):119, 2020. doi: 10.1038/s41746-020-00323-1. DOI: https://doi.org/10.1038/s41746-020-00323-1.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. doi: 10.48550/arXiv.1910.01108. DOI: https://doi.org/10.48550/arXiv.1910.01108.

Yujun Shi, Jian Liang, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In *International Conference on Learning Representations (ICLR)*, 2023. doi: 10.48550/arXiv.2210.00226.

Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019. doi: 10.48550/arXiv.1910.07796. DOI: https://doi.org/10.48550/arXiv.1910.07796.

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 *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 32991–33013, 2023. doi: 10.48550/arXiv.2305.11584.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2403.12313.

Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xinmei Tian, Tongliang Liu, Bo Han, and Xiaowen Chu. Fedimpro: Measuring and improving client update in federated learning. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2402.07011.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018. doi: 10.1093/jamia/ocx079. DOI: https://doi.org/10.1093/jamia/ocx079.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. doi: 10.5555/3495724.3496362. DOI: https://doi.org/10.5555/3495724.3496362.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024a. doi: 10.1109/TPAMI.2024.3367329.

Yujia Wang, Yuanpu Cao, Jingcheng Wu, Ruoyu Chen, and Jinghui Chen. Tackling the data heterogeneity in asynchronous federated learning with cached update calibration. In *International Conference on Learning Representations (ICLR)*, 2024b. URL https://openreview.net/forum?id=4aywmeb97I.

Yujia Wang, Shiqiang Wang, Songtao Lu, and Jinghui Chen. Fadas: Towards federated adaptive asynchronous optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 51701–51733, 2024c. doi: 10.48550/arXiv.2407.18365.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019. doi: 10.48550/arXiv.1903.03934. DOI: https://doi.org/10.48550/arXiv.1903.03934.

Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. Feddisco: Federated learning with discrepancy-aware collaboration. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 39879–39902, 2023. doi: 10.48550/arXiv.2305.19229.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11237–11244, 2023. doi: 10.1609/aaai.v37i9.26330.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16768–16776, 2024. doi: 10.1609/aaai.v38i15.29617.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. doi: 10.48550/arXiv.1806.00582. DOI: https://doi.org/10.48550/arXiv.1806.00582.

A Detailed Comparison with Related Work

Table 4: Comparison with related work. **FL**: federated learning. **Non-IID**: heterogeneous client data. **Async**: asynchronous aggregation. **PEFT**: parameter-efficient fine-tuning. **Forget.**: addresses catastrophic forgetting. **Clin.**: clinical data. **RAG**: retrieval augmentation from an external knowledge source. **Edge graph**: graph data built on the client. **Edge RAG+G**: graph construction and retrieval both on the client.

Method	FL	Non-IID	Async	PEFT	Forget.	Clin.	RAG	Edge graph	Edge RAG+G
Heterogeneity in FL									
FedDecorr (Shi et al., 2023)	✓	✓	×	×	×	×	×	×	×
Nguyen et al. (2023)	✓	✓	×	×	×	×	×	×	×
Client drift									
FedImpro (Tang et al., 2024)	✓	✓	×	×	×	×	×	×	×
FedSMOO (Sun et al., 2023)	✓	✓	×	×	×	×	×	×	×
FedLESAM (Fan et al., 2024)	✓	✓	×	×	×	×	×	×	×
FedDisco (Ye et al., 2023)	✓	✓	×	×	×	×	×	×	×
FedALA (Zhang et al., 2023)	✓	✓	×	×	×	×	×	×	×
CA2FL (Wang et al., 2024b)	✓	✓	✓	×	×	×	×	×	×
FADAS (Wang et al., 2024c)	✓	✓	✓	×	×	×	×	×	×
Forgetting and regularization									
Wang et al. (2024a)	×	×	×	×	✓	×	×	×	×
Biderman et al. (2024)	×	×	×	✓	✓	×	×	×	×
GradMA (Luo et al., 2023)	✓	✓	×	×	✓	×	×	×	×
FOT (Bakman et al., 2024)	✓	✓	×	×	✓	×	×	×	×
FedTGP (Zhang et al., 2024)	✓	✓	×	×	×	×	×	×	×
FedDAT (Chen et al., 2024)	✓	✓	×	✓	×	×	×	×	×
PEFT and clinical models									
DoRA (Liu et al., 2024)	×	×	×	✓	×	×	×	×	×
LoRA+ (Hayou et al., 2024)	×	×	×	✓	×	×	×	×	×
FFA-LoRA (Sun et al., 2024)	✓	✓	×	✓	×	×	×	×	×
FedSA-LoRA (Guo et al., 2025)	✓	✓	×	✓	×	×	×	×	×
GraphCare (Jiang et al., 2024)	×	×	×	×	×	✓	×	×	×
KARE (Jiang et al., 2025)	×	×	×	×	×	✓	✓	×	×
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓

B Reported Baseline Numbers

Table 5: Improvements of FedProx and SCAFFOLD over FedAvg as reported in the original papers.

Method	Reported setting	FedAvg	Method
FedProx (Li et al., 2020)	Synthetic, MNIST, FEMNIST, Shakespeare, Sent140, with 90% stragglers	+22% absolute test accuracy	
SCAFFOLD (Karimireddy et al., 2020)	EMNIST, 2-layer FC network, 1k rounds, 0% client similarity	0.787	0.801
SCAFFOLD (Karimireddy et al., 2020)	EMNIST, 2-layer FC network, 1k rounds, 10% client similarity	0.828	0.842

C Training Protocol

Algorithm 1 describes the asynchronous training procedure. The server and the clients run concurrently. Setting $\lambda_{\text{EWC}} = \lambda_{\text{KD}} = 0$ gives the unregularized configuration.

Algorithm 1 FedGAug asynchronous training

Require: Clients $\mathcal{K} = \{1, \dots, K\}$, number of server steps T_{max}, local epochs E, learning rate η, weights $\lambda_{\text{EWC}}, \lambda_{\text{KD}}$, temperature T_{distill}, staleness decay b, base mixing rate γ_0.

Server:
Initialize $t \leftarrow 0$, $\theta^{(0)} = \{\theta^{(0)}_{\text{adapter}}, \theta^{(0)}_{\text{GNN}}, \theta^{(0)}_{\text{fuse}}\}$.
Send $\theta^{(0)}$ to every client and set $t_{\text{issued},k} \leftarrow 0$ for all k.
while $t < T_{\text{max}}$ **do**
  Wait for an update θ^{local}_k from any client k.
  $\tau \leftarrow t - t_{\text{issued},k}, \quad s(\tau) \leftarrow (1 + b\tau)^{-1}$.
  Merge the update with Eq. 8: $\theta^{(t+1)} \leftarrow (1 - \gamma_0\, s(\tau))\, \theta^{(t)} + \gamma_0\, s(\tau)\, \theta^{\text{local}}_k$.
  $t \leftarrow t + 1$.
  Send $\theta^{(t)}$ to client k and set $t_{\text{issued},k} \leftarrow t$.
end while

Client k (each time it receives $\theta^{(t_{\text{issued},k})}$):
Set the anchor $\theta^\star \leftarrow \theta^{(t_{\text{issued},k})}$ and initialize $\theta_k \leftarrow \theta^\star$.
Compute the diagonal Fisher estimate F_k at $\theta^\star$ with Eq. 5.
for epoch $e = 1, \dots, E$ **do**
  for each mini-batch $\mathcal{B}_k \subset \mathcal{D}_k$ **do**
    Compute $\mathcal{L}_{\text{task}}$, $\mathcal{R}_{\text{EWC}}$ (Eq. 6), and $\mathcal{R}_{\text{KD}}$ (Section 3).
    Combine them into $\mathcal{L}_{\text{local}}(\theta_k)$ with Eq. 7.
    Update θ_k with one AdamW step on $\mathcal{L}_{\text{local}}$ with learning rate η.
  end for
end for
Send $\theta^{\text{local}}_k \leftarrow \theta_k$ to the server.

D Hyperparameters and Per-Client Forgetting

Table 6: Hyperparameters used for FedGAug.

Category	Hyperparameter	Value
Federated setup	Number of clients (K)	3
	Local epochs per round (E)	3
	Communication rounds	29
	Base mixing rate (γ_0)	0.6
	Staleness decay (b)	0.5
	Staleness discount	$s(\tau) = (1 + b\tau)^{-1}$, not capped
	Client scheduling	Asynchronous, merged on arrival
Transformer backbone	Base architecture	DistilBERT (base, uncased)
	LoRA rank (r)	4
	LoRA scaling factor (α_{LoRA})	16
	Target attention layers	Query (W_q), Value (W_v)
Graph encoder	Message-passing encoder	2-layer GCN
	Hidden dimension	128
	Pooling	Mean readout
Local optimizer	Optimizer	AdamW
	Learning rate (η)	2×10^{-4}
	Weight decay	0.01
	Batch size ($\lvert\mathcal{B}_k\rvert$)	32
Regularization	EWC weight (λ_{EWC})	$\{0.0, 0.5, 1.0\}$
	KD weight (λ_{KD})	$\{0.0, 1.0\}$
	Distillation temperature (T_{distill})	2.0

Table 7: Peak-to-final forgetting of global AUROC for each client over rounds 1 to 29. Each client's trajectory is averaged over its three seeds first. The last column shows the mean and standard deviation across clients.

Method	**Client 1**	**Client 2**	**Client 3 (small)**	**Mean ± Std**
FedGAug (unregularized)	0.223	0.225	0.058	0.169 ± 0.096
FedGAug (EWC 0.5)	0.003	0.002	0.000	0.002 ± 0.001
FedGAug (EWC 1.0)	0.004	0.008	0.003	0.005 ± 0.003
FedGAug (EWC 1.0 + KD)	0.001	0.001	0.002	0.001 ± 0.001

E Graph Encoder Architecture

Initial node embeddings $\mathbf{h}_v^{(0)} \in \mathbb{R}^{d_{\text{node}}}$ for each code $v \in \mathcal{V}$ are updated over L layers:

$$\mathbf{h}_v^{(\ell+1)} = \text{ReLU}\Big(W_{\text{GNN}}^{(\ell)} \cdot \text{AGG}\Big(\Big\{\mathbf{h}_u^{(\ell)} : u \in \mathcal{N}(v) \cup \{v\}\Big\}\Big)\Big), \quad \ell = 0, \ldots, L-1, \tag{9}$$

where $\mathcal{N}(v)$ is the set of neighbors of v and AGG is the mean. The graph embedding is the mean over nodes, $\mathbf{h}_{\mathcal{G}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{h}_v^{(L)}$.

The sequence and graph representations are concatenated and passed through the fusion layer $\theta_{\text{fuse}} = \{W_f, \mathbf{b}_f\}$:

$$\mathbf{h}_{\text{patient}} = \text{ReLU}(W_f[\mathbf{h}_{\text{seq}} \,\|\, \mathbf{h}_{\mathcal{G}}] + \mathbf{b}_f), \quad \mathbf{z}_\theta(x) = \mathbf{w}_{\text{out}}^\top \mathbf{h}_{\text{patient}} + b_{\text{out}}, \quad \hat{y}_\theta(x) = \sigma(\mathbf{z}_\theta(x)). \tag{10}$$

F Limitations

We use synthetic Synthea data, a single prediction task, and three clients, and two of the clients have very similar label prevalence, so the degree of heterogeneity is modest.