HETEROGENEOUS FEDERATED GRAPH LEARNING VIA OPTIMIZER SECOND-MOMENT COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph learning (FGL) has emerged as a promising approach to train graph neural networks on distributed graph data. However, existing methods often cause personalized learning degradation and convergence failure due to mismatch fixed second moments of gradients on different clients with heterogeneous graph dataset. To address this issue, we propose a novel federated graph learning framework, namely FedSeed, which leverage the second moment of Adam optimizer as statistical metric of multi-hop graphs feature. On the client side, FedSeed designs an adaptive gradient update mechanism to adapt local heterogeneity, which dynamically balance the local and global second moment on each client per round. On the server side, FedSeed proposes a similarity aggregation strategy to mitigate model drift by heterogeneity, which aggregates different local second moments by clustering. Extensive experiments demonstrate that FedSeed significantly outperforms state-of-the-art methods on accuracy over real-world graph datasets with statistical and topological heterogeneity settings.

1 INTRODUCTION

Federated graph learning (FGL) enables multiple clients to jointly train graph neural networks on local subgraphs without sharing raw data Liu et al. (2025). FGL also handles large-scale graphs through scalable local models Li et al. (2024). However, the graph data stored by different clients are often non-independent and identically distributed (non-IID). Such data appears as differences in graph structures and node feature distributions across clients Hsu et al. (2019); Li et al. (2020c). These differences slow convergence and degrade final accuracy. Overcoming them lets FGL make full use of distributed graph data and improve accuracy for every client.

Existing FGL methods address these constraints by three ways. Knowledge-transfer-based methods Zhang et al. (2021); Peng et al. (2023) complete missing neighbors or align client representations to improve collaboration. Topology-aware aggregation methods Zhu et al. (2024); Li et al. (2024) weight client contributions by structural similarity. Personalized methods Baek et al. (2023); Chen et al. (2026) learn client-specific components or elastic alignments for each local distribution.

However, existing methods still share a same global second moment across all clients. Both the mixing weight and the shared second moment stay fixed uniformly, regardless of client differences. We summarize two challenges as follows. i) How to derive the mixing weight from the optimizer's own statistics is the key challenge. Existing methods fix it as a constant, so clients with different graph structures are treated identically Liu et al. (2026); Chen et al. (2026). ii) How to decide how much clients share with each other is the important challenge. Existing similarity-weighted aggregation relies on handcrafted structural descriptors or on loss signals drifting during training. It reduces to hard partitioning. Such partitioning is unreliable with only a handful of clients per cluster Li et al. (2024); Xie et al. (2021).

This motivates our core idea: using the second moment of the local optimizer for both per-parameter mixing on the client and topology-aware aggregation on the server. Specifically, at the client level, we derive a per-parameter coefficient to mix the local second moment with the shared one. This per-parameter mixing keeps the gradient update step size of each parameter specific to the client, instead of a same step size shared by all clients. At the server level, we design an similarity aggregation strategy to build the shared second moment. It computes a topology-weighted average

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.

of client second moments. The weights are larger for topologically similar clients and smaller for topologically different clients.

The contributions of this work are summarized as follows:

- We propose FedSeed, a novel personalized graph learning framework for federated graph learning. Different client graph topologies produce different second moments of gradients. FedSeed keeps each client's gradient update step size specific to its own topology, while the server aggregates second moments from topologically similar clients.
- On the client side, FedSeed designs an adaptive gradient update rule. It derives a per-parameter coefficient to mix the local second moment with the shared one. This coefficient keeps each client's gradient update step size specific to its own topology, so personalized training is improved under topology heterogeneity.
- On the server side, FedSeed designs an similarity aggregation strategy. It aggregates each client's second moment across topologically similar clients,so similar clients share more and dissimilar clients share less. The elastic aggregation strategy improves coordination of clients under topology heterogeneity.
- We evaluate FedSeed under statistical and topological heterogeneity settings. Experiments and ablation studies show that FedSeed outperforms state-of-the-art methods, improving task accuracy by up to 8.0%.

2 Related Work

Existing FGL for graph heterogeneity fall into three categories: neighbor-completion, topology-aware aggregation, and personalization methods.

Neighbor-Completion Methods. These methods work at the data level and repair the subgraph structure destroyed by partitioning. FedSage+ Zhang et al. (2021) generates missing neighbors with a cross-client generator. FedNI Peng et al. (2023) performs network inpainting for missing nodes. FedGNN Wu et al. (2021) completes the local neighborhood for recommendation. They repair structure before training. In this way, they align local objectives and keep the aggregation rule unchanged. However, generation adds computation and communication. They also assume a faithful recovery of the missing structure. A partition often removes most cross-client edges, and then this assumption fails. More importantly, these methods act on graph data only. They never inspect the optimizer state from local training.

Topology-Aware Aggregation Methods. These methods use structural signals during aggregation. FedGTA Li et al. (2024) weights client contributions by topology-aware smoothing confidence and mixed moments of neighbor features. FedTAD Zhu et al. (2024) decouples node variation from topology variation and reweights knowledge transfer by class-wise reliability. FedIA Zhou et al. (2025) selects important parameter coordinates and reweights clients by their contributions on the selected subspace. Clustered approaches Xie et al. (2021); Ghosh et al. (2020); Sattler et al. (2021) group clients by structural similarity and aggregate within each group. However, these methods need structural descriptors, and the descriptors are computed and transmitted in addition to the model. Other methods reduce to hard partitioning. Hard partitioning is unreliable with only a handful of clients per group. None of these methods derives its weighting from a quantity already carried by the federation.

Personalization Methods. These methods adapt the model or its training to each client. FedPub Baek et al. (2023) learns personalized subgraph models. GraphP-FL Chen et al. (2026) combines a client-side topology reconstruction with an elastic alignment at the server, and it uses the Fisher diagonal as an anisotropic penalty and as a per-client aggregation weight. Parameter-space remedies, including proximal terms Li et al. (2020b), control variates Karimireddy et al. (2020); Wang et al. (2020), and local steps Stich (2019), constrain update drift and leave the optimizer untouched. Server-side adaptive optimizers Reddi et al. (2021b) and FedAdamW Liu et al. (2026) act on the optimizer, but they replace the client's second moment wholesale. However, these methods either adjust the model parameters rather than the optimizer state, or fix the personalization degree by construction, as a single constant or an external strength. The normalization for a client's step

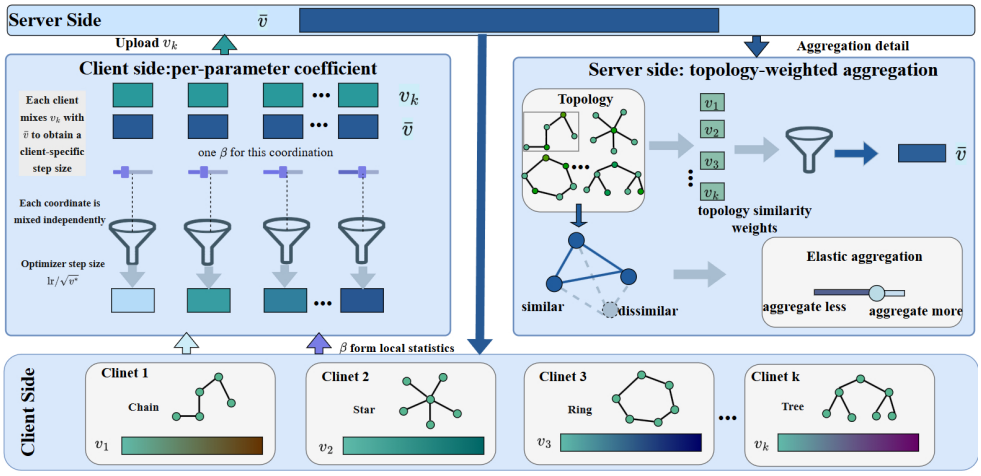


Figure 1: The Architecture of FedSeed

size is therefore never adjusted. No client can obtain a degree adapted to its own topology, and each client's step size still depends on the ignored second moment.

3 Methodology

3.1 Problem Statement

We consider a federated graph learning system with one central server and K clients. Client k holds a private subgraph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k, \mathbf{X}_k, \mathbf{Y}_k)$ partitioned from a global graph, with $n_k = |\mathcal{V}_k|$ nodes and row-normalized adjacency $\mathbf{A}_k$. Clients share a common architecture and collaboratively minimize

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k\big(f(\mathcal{G}_k;\boldsymbol{\theta}), \mathbf{Y}_k\big), \tag{1}$$

where $\mathcal{L}_k$ is the local cross-entropy loss, subject to a communication budget per round. Each round, every participating client receives the global model $\boldsymbol{\theta}^{(t)}$, trains locally with Adam for E epochs, and uploads its update together with a summary of its optimizer state. The budget forces that summary to be small: we allow at most $O(K)$ scalars per round against d model parameters.

Our starting point is that the local optimizer state is not a byproduct but a statistic of the client's graph. Let $\mathbf{v}_k$ denote the second moment Adam maintains for client k, and let $\bar{\mathbf{v}}_k$ denote the server-side second moment sent to client k. The quantity we act on is the mismatch

$$\Delta_k = \bar{\mathbf{v}}_k - \mathbf{v}_k. \tag{2}$$

Under topology heterogeneity this mismatch is systematic rather than incidental, because $\mathbf{v}_k$ is a function of $\mathbf{A}_k$; under partial participation it is time varying, because $\bar{\mathbf{v}}_k$ changes while an absent client's $\mathbf{v}_k$ does not. FedSeed keeps a server-side second moment for each client, returns it to that client together with the global model, and lets each client decide how much of it to mix into its local second moment.

3.2 Overview of FedSeed

As shown in Figure 1, FedSeed coordinates each client's second moment on two sides. On the client side, FedSeed derives a per-parameter coefficient from the client's own optimizer statistics. This coefficient mixes the local second moment with a server-side second moment. On the server side, FedSeed applies an similarity aggregation. The server updates one second moment per client as a topology-weighted average of the participating clients' second moments. The bandwidth of

this average follows the observed dispersion of client profiles. Both sides act on one quantity, the second moment, and exchange it on a single communication channel of about ten floats per round. Algorithm 1 summarizes one round. Section 3.3 describes the second moment and the channel, Section 3.4 the client-side mixing, and Section 3.5 the server-side aggregation.

3.3 The Second Moment and the Communication Channel

Why the second moment carries structure. Writing $\delta_{t,c}$ for the residual and $z_t^{(h)}$ for the h-th hop feature of node t, the gradient of the readout block $\mathbf{W}_{c,h}$ is $\partial\mathcal{L}/\partial\mathbf{W}_{c,h} = |T_k|^{-1}\sum_t \delta_{t,c} z_t^{(h)}$. Under residual–input separability, which holds to first order under a Taylor expansion of the residual in the features, and task-factor balance across output coordinates, the cross terms vanish and

$$v_{c,h} = \mathbb{E}\big[(\partial\mathcal{L}/\partial\mathbf{W}_{c,h})^2\big] \;\propto\; \frac{\tau^2\,\mathbb{E}\|\mathbf{z}^{(h)}\|^2}{|T_k|}. \tag{3}$$

Summing over c, the common factors cancel and the hop energy share of the second moment equals the client's multi-hop energy profile,

$$\frac{\sum_c v_{c,h}}{\sum_{c,h'} v_{c,h'}} \;\approx\; e_k(h) := \frac{\|\mathbf{A}_k^h \mathbf{X}_k\|_F^2}{\sum_{h'}\|\mathbf{A}_k^{h'}\mathbf{X}_k\|_F^2}, \tag{4}$$

which contains no n_k. Two consequences matter. First, the systematic part of the second moment is low dimensional, residing in the $K{+}1$ hop shares, while the per-parameter residual has zero mean conditional on the block; this makes the second moment worth transmitting and possible to compress. Second, the server-side second moment affects training only through the inverse square root of the local second moment, so its effect is bounded: comparing a mixed second moment v_s with the second moment v^* a client would use, the relative distortion of an Adam step obeys

$$D = \frac{\mathbb{E}\,|u(v_s) - u(v^*)|}{\mathbb{E}\,|u(v^*)|} \;\le\; \frac{1}{2}\,\mathbb{E}\left[\frac{|v_s - v^*|}{v^*}\right], \tag{5}$$

whenever $v_s \ge v^*/2$ per parameter, and becomes unbounded when the server-side second moment falls below that level on a set of parameters of positive measure. This failure condition is not a technicality: it is exactly the regime in which replacing a client's second moment by a structureless mean becomes destructive, and it is why the deployed rule must price each parameter individually rather than globally.

The global second moment. The only persistent state introduced by FedSeed is one second moment per client, an exponential moving average of the second moments uploaded by the participating clients:

$$\bar{\mathbf{v}}_k^{(t)} = \lambda\,\bar{\mathbf{v}}_k^{(t-1)} + (1-\lambda)\,\frac{\sum_{j\in S_t} w_{kj}\,\mathbf{v}_j^{(t)}}{\sum_{j\in S_t} w_{kj}}, \qquad \lambda = 0.9, \tag{6}$$

where S_t is the participating set in round t and w_{kj} is the similarity weight of Equation equation 15. Three properties of this update matter. The server keeps one second moment per client, so the variation the federation undergoes while a client is absent is recorded on the server. Only the second moments of participating clients enter the update, so the global second moment tracks the current federation rather than a historical snapshot. And because it is broadcast with the model, it costs no additional communication. The global second moment is also far less noisy than any single client's state: for near-uniform weights, averaging over the participating set reduces the variance of a per-client estimate Var_l to $\mathrm{Var}_l/|S_t|$, and the moving average adds a further average over an effective horizon $r_{\mathrm{eff}} = (1+\lambda)/(1-\lambda) = 19$ rounds, so

$$\mathrm{Var}_a = \frac{\mathrm{mean}_k \mathrm{Var}_{l,k}}{|S_t|\,r_{\mathrm{eff}}}, \tag{7}$$

up to a bias of order $1/r_{\mathrm{eff}}$ that decays geometrically. The reliability of the global second moment therefore improves with the size of the federation.

The compressed channel. Transmitting $\bar{\mathbf{v}}_k$ in full would cost d floats per round and violate the budget of Equation equation 1. We therefore transmit the second moment in a form that is constant within each hop block. Since the readout of a decoupled GNN is indexed by propagation hop, we

send one scalar per hop block, namely the root mean square of v^2 within the block, and one scalar per remaining tensor, for a total of about $(K{+}1) + L^- \approx 10$ floats. The server-side second moment is compressed in the same form, so the same channel carries both the mixed second moment and the model, and neither direction incurs extra cost. By the factorization of Equation equation 3, v is exactly constant within a hop block when the within-block feature second moments are isotropic, so the reconstruction error vanishes. More generally, writing $v = c_h(1+\varepsilon)$ within block h with zero-mean ε, the mean absolute relative deviation is controlled by the within-block anisotropy κ_Z of the feature second moment, and substituting this into Equation equation 5 gives

$$D(\hat{\mathbf{v}}) \le \frac{1}{2} \cdot \frac{\kappa_Z - 1}{\kappa_Z + 1}. \tag{8}$$

The bound depends only on within-block anisotropy and not on the ambient dimension, so the channel stays sufficient as the model grows. Block averaging additionally reduces the sampling noise of the empirical second moment in proportion to the within-block correlation length, so compression can be mildly beneficial rather than merely tolerable, and it is orthogonal to bit-level quantization.

3.4 Client-Side Mixing

At the end of each round, every participating client mixes its server-side second moment into its own second moment,

$$\mathbf{v}_k \leftarrow \beta \odot \mathbf{v}_k + (1-\beta) \odot \bar{\mathbf{v}}_k, \tag{9}$$

applied per parameter, with the first moment $\mathbf{m}_k$ left untouched. The difficulty lies entirely in β: no single value is correct everywhere. Forcing every client onto the server-side second moment distorts the step of a client with an accurate local second moment, while leaving every client alone forfeits coordination. We therefore derive β rather than set it, on two levels that serve different purposes.

Estimation-level optimum and the safety boundary. Treat the local moving average $\hat{\mathbf{v}}_k$ and the global second moment $\hat{\bar{\mathbf{v}}}_k$ as two estimators of the same true second moment $\mathbf{v}_k$. The local estimator is unbiased with variance $\mathrm{Var}_{l,k}$, whereas the server-side second moment has the smaller variance Var_a of Equation equation 7 but carries a bias $\Delta_k = \bar{\mathbf{v}}_k - \mathbf{v}_k$. Writing the mixture of Equation equation 9 as $v(\beta)$, its mean squared error and its minimizer are

$$\mathrm{MSE}_k(\beta) = \beta^2 \mathrm{Var}_{l,k} + (1-\beta)^2 \big(\mathrm{Var}_a + \Delta_k^2\big), \tag{10}$$

$$\beta_k^* = \frac{\mathrm{Var}_a + \Delta_k^2}{\mathrm{Var}_{l,k} + \mathrm{Var}_a + \Delta_k^2}. \tag{11}$$

The two limits identify the regimes of existing methods. When the states agree across the federation and the global second moment is well estimated, so that $\Delta_k = 0$ and $|S_t| r_{\mathrm{eff}} \gg 1$, the coefficient tends to zero and *replacing the local state with the global mean is optimal*; this is exactly the reset of FedAdamW Liu et al. (2026), recovered as one end of a continuous rule rather than as a competing design. When the mismatch dominates estimation noise, $\Delta_k^2 \gg \mathrm{Var}$, the coefficient tends to one and pure local protection is optimal. A fixed coefficient $\bar{\beta}$ is optimal only on the single surface $\mathrm{Var}_a + \Delta_k^2 = \frac{\bar{\beta}}{1-\bar{\beta}} \mathrm{Var}_{l,k}$, so a tuned constant is one point of that surface and no constant is uniformly correct. The role of this level is not to produce the gain but to locate the safe mixing.

Update-level risk and the exact coefficient. Estimating v more accurately does not by itself improve training, because the mixed second moment is also the denominator of the Adam step. The gain appears in update space. Consider one parameter and compare the step taken under the mixed second moment, with normalizer $u(\beta) = v(\beta)^{-1/2}$, with the step the client would take under its own second moment, $u_k = v_k^{-1/2}$. Writing m for the first moment and $\hat{\sigma}^2 = \max(v_k - m^2, 0)$ for the plug-in estimate of the gradient noise variance, the mean squared deviation of the two steps decomposes exactly, as an identity rather than an approximation,

$$R_k(\beta) = \underbrace{\Big(\frac{m}{\sqrt{v(\beta)}} - \frac{m}{\sqrt{v_k}}\Big)^2}_{\text{distortion}} + \underbrace{\frac{\hat{\sigma}^2}{v(\beta)}}_{\text{noise floor}}, \tag{12}$$

and the deployed coefficient minimizes this risk per parameter, $\beta_k^{\star\star} = \arg\min_{\beta \in [0,1]} R_k(\beta)$. The two terms oppose each other. The first term penalizes the distortion caused by replacing the client's

own second moment: where the client carries a distinct and strong signal, moving the denominator harms it, so this term pushes the coefficient toward one and protects the local state. It is the online form of the failure condition of Equation equation 5. The second term penalizes noise amplification: the variance contributed to the update by gradient noise is $\hat{\sigma}^2/v(\beta)$, which grows without bound as the denominator shrinks. For a parameter with a small local second moment, normalizing against it magnifies noise, and the global second moment supplies a larger and more stable floor, so this term pushes the coefficient toward zero. This second term is the source of the gain from coordination.

All four inputs, m, v_k, $\hat{\sigma}^2$ and $\bar{\mathbf{v}}_k$, already exist: the first three are local optimizer state and the fourth arrives on the same channel as the model. The rule therefore introduces no hyperparameter. Because $v(\beta)$ is affine in β and the risk depends on it only through $u = v(\beta)^{-1/2}$, the minimization admits an exact solution. With $s = m^2$, the distortion term equals $s(u - u_k)^2$, so Equation equation 12 is a strictly convex quadratic in u with unconstrained minimizer $u^\star = s\,u_k/(s + \hat{\sigma}^2)$. Since $\beta \mapsto v(\beta) \mapsto u$ is a strictly monotone bijection onto the interval between $u_0 = \bar{v}_k^{-1/2}$ and u_k, the constrained minimizer is the projection of $u^\star$ onto that interval,

$$u^\dagger = \mathrm{clip}\big(u^\star, \min(u_0, u_k), \max(u_0, u_k)\big), \tag{13}$$

$$\beta_k^{\star\star} = \frac{u^{\dagger -2} - \bar{v}_k}{v_k - \bar{v}_k}. \tag{14}$$

Four behaviours follow directly. Where the local state already agrees with the server-side second moment, the mixture degenerates to a point and mixing is an exact no-op. Where the global second moment is below the local second moment, $u^\star$ lies beyond u_k, the projection returns $\beta_k^{\star\star} = 1$, and the local second moment is left unchanged; the rule therefore never lowers a parameter's second moment below its own value. Where the global second moment is above and the parameter has a small local second moment, the second moment is raised and the noise floor is lowered, which is the gain mechanism. And where the parameter carries no signal, the distortion term vanishes identically and the rule takes the larger of the two second moments.

3.5 Server-Side Similarity Aggregation

A uniform mean needs no information about how clients relate to one another. For exactly that reason, it is the wrong choice here. We have established that the second moment carries the client's topology. A uniform mean therefore compares every client against peers of possibly different topology. The bias it introduces is a systematic offset, not noise reduced by averaging. The same channel as the model already tells us which clients are comparable, so we use it.

The weights in Equation equation 6 are similarity-weighted,

$$w_{kj} \propto \exp\Big(- \frac{\|\mathbf{p}_k - \mathbf{p}_j\|^2}{2h^2} \Big), \tag{15}$$

where $\mathbf{p}_k$ is the hop energy profile of client k and h is the bandwidth of the aggregation. Three properties distinguish this construction.

It requires no extra information. The profile $\mathbf{p}_k$ is already on the same channel as the model, so the weighting requires no descriptor to be designed, no extra statistic to be computed, and no additional byte to be transmitted. Methods that weight aggregation by structure must construct and transmit a structural summary; the summary is the quantity the federation already exchanges for compression.

It is continuous. Because the weights are normalized, the aggregate is unbiased in client noise, and its error splits into a structural bias controlled by the bandwidth and an estimation variance controlled by the effective sample size. When client profiles are well separated, the aggregate is strictly better than the uniform mean over a wide range of bandwidths. When they coincide, the kernel weights become uniform and the aggregate reduces exactly to the uniform mean. A hard partition has no such limit: it commits each client to a group, and a misassignment corrupts the aggregate abruptly instead of degrading it gracefully, because the number of clients per group is small in realistic federations. This aggregate interpolates between the two extremes of a uniform mean and a hard partition, and the data determines the position.

It is not tuned. The bandwidth is set by the observed dispersion σ_w of the client profiles, estimated online from their pairwise distances in the current round. The optimal bandwidth scales with σ_w,

Algorithm 1 FedSeed: one communication round

Require: global model $\boldsymbol{\theta}^{(t)}$; per-client global second moment $\{\bar{\mathbf{v}}_k^{(t)}\}_{k=1}^K$; participating set S_t; local epochs E

Ensure: updated model $\boldsymbol{\theta}^{(t+1)}$; updated per-client second moment $\{\bar{\mathbf{v}}_k^{(t+1)}\}_{k=1}^K$

Server: send $\boldsymbol{\theta}^{(t)}$ and $\bar{\mathbf{v}}_k^{(t)}$ to each client k
for each client $k \in S_t$ in parallel **do**
  compute $\beta_k^{\star\star}$ from m_k, v_k, $\hat{\sigma}_k^2$ and $\bar{\mathbf{v}}_k^{(t)}$ by Eq. equation 14
  mix the local second moment: $\mathbf{v}_k \leftarrow \beta_k^{\star\star} \odot \mathbf{v}_k + (1 - \beta_k^{\star\star}) \odot \bar{\mathbf{v}}_k^{(t)}$
  for $e = 1, \ldots, E$ **do**
    update $\boldsymbol{\theta}_k$ by one Adam step on $\mathcal{G}_k$
  end for
  compute the hop energy profile $\mathbf{p}_k$ from $\mathbf{v}_k$
  send $\Delta\boldsymbol{\theta}_k$ and the compressed $\mathbf{v}_k$ to the server
end for
Server: estimate the profile dispersion σ_w from $\{\mathbf{p}_k\}_{k \in S_t}$
Server: set the bandwidth $h \propto \sigma_w$
Server: update $\bar{\mathbf{v}}_k^{(t+1)}$ for every k over S_t by Eq. equation 6
Server: aggregate $\{\Delta\boldsymbol{\theta}_k\}_{k \in S_t}$ into $\boldsymbol{\theta}^{(t+1)}$
Server: send $\boldsymbol{\theta}^{(t+1)}$ and $\bar{\mathbf{v}}_k^{(t+1)}$ to each client k

so the construction is calibrated rather than searched, and it adapts as the structural spread of the federation changes.

4 Experiments

4.1 Experimental Setup

Datasets. We use: Cora, CiteSeer, and PubMed Yang et al. (2016); Amazon Photo and Amazon Computers Shchur et al. (2018); Ogbn-Arxiv Hu et al. (2020); and AGNews Zhang et al. (2015).

Backbones. We use GCN Kipf & Welling (2017) and GAMLP Zhang et al. (2022) as local models.

Partition strategies. Metis partitions the global graph into structurally balanced subgraphs. The Dirichlet distribution with concentration parameter α controls statistical heterogeneity, and smaller α means stronger heterogeneity. We use $\alpha \in \{0.1, 0.5, 10.0\}$.

Baselines. We compare FedSeed with FedAvg McMahan et al. (2017), FedProx Li et al. (2020a), Scaffold Karimireddy et al. (2020), FedAdam Reddi et al. (2021a), FedAWA Shi et al. (2025), and FedSOL Lee et al. (2024).

Protocol. The hidden dimension is 128, the dropout rate is 0.5, and the weight decay is 5×10^{-4}. Each round runs 5 local epochs with a batch size of 128 and a learning rate of 0.01 using Adam, for 200 communication rounds. All clients participate in each round unless stated otherwise. We report the mean over three random seeds with standard deviation.

4.2 Main Results across Client Scales (Q1)

Tables 1 and 2 report accuracy on Cora, Ogbn-Arxiv, and Amazon Photo with 5, 10, and 20 clients under GCN and GAMLP, using the Metis partition. FedSeed ranks first in all 18 configurations. The largest gain is on Ogbn-Arxiv with GAMLP at 20 clients, where FedSeed reaches 64.55% and exceeds the strongest baseline by 8.0%. On Cora with GCN at 10 clients and Ogbn-Arxiv with GCN at 20 clients, FedSeed improves the strongest baseline by 3.7% and 5.7%, respectively. Across all configurations, FedSeed improves the strongest baseline by 0.5% to 8.0%.

FedSeed performs well under both GCN and GAMLP, so the second moment is not tied to one architecture. On Amazon Photo with GAMLP at 10 clients, FedSeed reaches 94.14%. The gain over the strongest baseline grows with the number of clients: on Ogbn-Arxiv with GCN, it rises

Table 1: Accuracy comparison of federated learning methods across different datasets with 5, 10, and 20 clients under GCN model (%)

	Cora			Ogbn-Arxiv			Amazon Photo		
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients
Global	85.31±0.20	84.98±0.35	85.23±0.23	52.08±0.26	52.35±0.32	51.97±0.27	93.53±0.15	93.73±0.14	93.36±0.18
FedAvg	82.30±0.09	77.17±0.85	69.31±0.0	62.64±0.23	61.77±0.06	59.39±0.36	92.81±0.03	89.25±0.65	90.50±0.05
FedProx	82.91±0.19	81.56±0.20	81.18±0.13	62.91±0.27	62.24±0.14	59.63±0.37	92.63±0.33	90.30±0.55	91.79±0.22
FedAWA	81.75±0.20	77.24±0.83	69.55±0.33	62.64±0.22	62.17±0.11	59.40±0.36	92.55±0.28	90.45±2.54	88.98±1.54
FedAdam	80.93±0.99	74.01±0.26	66.50±0.70	62.47±0.58	61.65±0.29	56.91±1.04	90.71±1.26	80.43±1.64	69.12±1.21
FedSOL	81.73±0.43	76.60±0.74	68.12±0.62	62.53±0.43	61.61±0.48	58.98±0.65	90.85±0.73	84.21±0.77	89.92±1.43
Scaffold	81.38±0.35	81.65±0.20	81.92±0.23	63.34±0.08	61.67±0.18	59.77±0.11	92.22±0.30	91.14±0.25	90.07±0.59
FedSeed (ours)	**83.73±0.14** (↑1.0%)	**84.70±0.17** (↑3.7%)	**84.62±0.13** (↑3.3%)	**66.67±0.15** (↑5.3%)	**64.05±0.21** (↑2.9%)	**63.16±0.26** (↑5.7%)	**93.96±0.19** (↑1.2%)	**93.78±0.13** (↑2.9%)	**93.06±0.11** (↑1.4%)

Table 2: Accuracy comparison of federated learning methods across different datasets with 5, 10, and 20 clients under GAMLP model (%)

	Cora			Ogbn-Arxiv			Amazon Photo		
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients
Global	85.79±0.54	82.43±0.55	85.97±0.30	53.38±0.51	57.21±0.11	54.62±0.34	93.83±0.17	93.36±0.55	93.68±0.16
FedAvg	82.36±0.00	76.95±0.06	69.40±0.12	62.64±0.24	62.22±0.11	59.39±0.36	92.01±0.08	91.19±0.16	90.70±0.12
FedProx	82.90±0.19	80.96±0.21	81.18±0.13	62.89±0.25	62.32±0.14	59.63±0.38	92.48±0.32	91.93±0.45	91.80±0.29
FedAWA	81.75±0.20	77.00±0.22	69.54±0.34	62.64±0.22	62.17±0.11	59.40±0.37	92.73±0.26	91.29±0.56	89.30±1.15
FedAdam	80.93±0.99	74.00±2.26	66.54±2.45	62.46±0.59	61.65±0.35	56.91±1.04	89.78±1.12	81.49±1.63	76.15±1.50
FedSOL	81.73±0.42	76.03±0.55	68.11±0.61	62.53±0.43	61.61±0.48	58.98±0.65	90.58±0.76	90.88±0.67	89.20±0.72
Scaffold	82.47±0.31	81.83±0.25	81.18±0.81	63.29±0.12	62.86±0.22	59.75±0.07	92.34±1.62	91.94±0.30	91.54±0.36
FedSeed (ours)	**84.53±0.23** (↑2.0%)	**83.31±0.24** (↑1.8%)	**84.95±0.15** (↑4.6%)	**66.37±0.07** (↑4.9%)	**65.21±0.22** (↑3.7%)	**64.55±0.08** (↑8.0%)	**93.16±0.58** (↑0.5%)	**94.14±0.45** (↑2.4%)	**92.33±0.16** (↑0.6%)

from 5.3% at 5 clients to 5.7% at 20 clients. More clients mean more fragmented local subgraphs, and FedSeed benefits from this fragmentation.

Figure 2 extends the evaluation to extreme non-IID conditions across four additional datasets: Computers, PubMed, CiteSeer, and AGNews. FedSeed achieves strong accuracy and fast convergence on these datasets. On Computers, it exceeds the centralized Global model. FedAdam and FedSOL degrade sharply on CiteSeer and AGNews, while FedSeed remains stable.

4.3 Robustness under Statistical Heterogeneity (Q2)

We evaluate FedSeed on AGNews with GAMLP at 20 clients across three Dirichlet levels. As Figure 3 shows, FedSeed attains the highest accuracy at every level. The advantage is largest under extreme heterogeneity, where FedSeed reaches 85.5% and exceeds FedProx by 2.0%. FedAdam varies by 19.7 percentage points across the spectrum, while FedSeed varies by only 4.4 points. This stability supports the use of FedSeed in deployments with unpredictable data distributions.

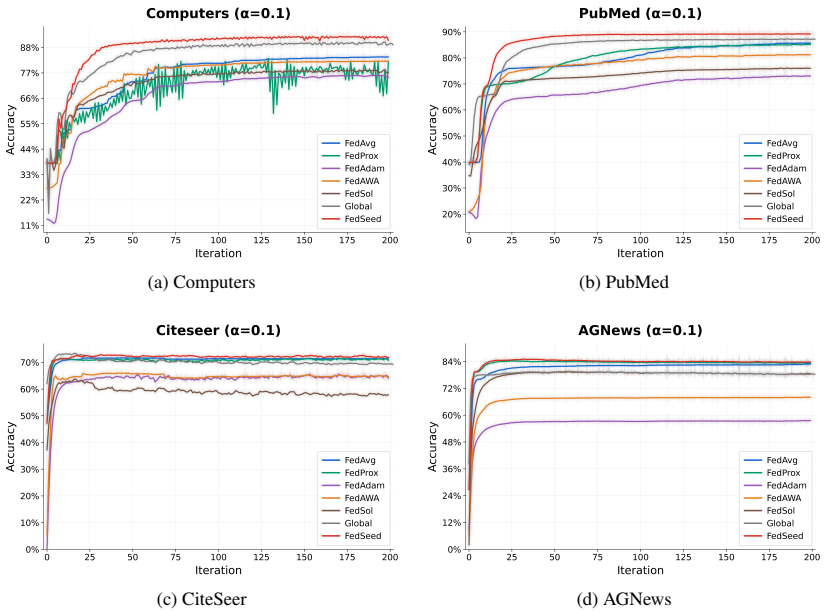
4.4 Ablation Study (Q3)

We ablate the two components on CiteSeer with GCN at 10 clients. Figure 4 reports FedAvg, the two single-component variants, and the full model.

Client-side coefficient. FedSeed w/o client-side coefficient keeps the server-side elastic aggregation and replaces the per-parameter coefficient by a fixed mixing weight. It reaches 69.41% and improves FedAvg (68.78%) by 0.63%. The server-side aggregation alone therefore gives a modest gain.

Server-side similarity aggregation. FedSeed w/o server-side similarity aggregation keeps the client-side coefficient and replaces the topology-weighted average by a uniform mean. It reaches 70.81% and improves FedAvg by 2.03%.

Complementary gains. The full FedSeed reaches 72.36%. Adding the server-side aggregation on top of the client-side coefficient improves the result by 1.55% (70.81% to 72.36%), and adding the



(a) Computers (b) PubMed

(c) CiteSeer (d) AGNews

Figure 2: Accuracy curves under extreme non-IID conditions (α=0.1, K=20, GCN).

client-side coefficient on top of the aggregation improves it by 2.95% (69.41% to 72.36%). The gain of FedAvg is 3.58%, which exceeds the sum of the two individual gains (2.66%). The two components therefore reinforce each other: the server aggregates second moments by topology, and the client returns this information to each local optimizer.

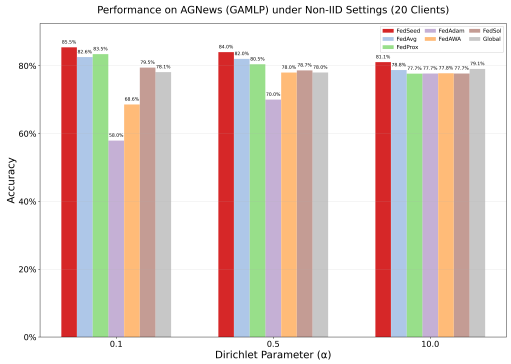


Figure 3: Accuracy on AGNews under varying α (GAMLP, K=20).

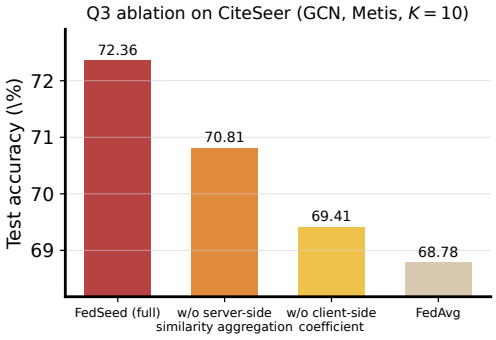


Figure 4: Ablation study on CiteSeer (Metis, K=10, GCN).

5 Conclusion

We proposed FedSeed, a second-moment framework for federated graph learning. On the client side, a per-parameter coefficient mixes the local second moment with the global second moment. On the server side, a similarity-weighted aggregation builds each client's global second moment from topologically similar clients. A single compressed channel carries the second moment in both directions. Experiments on graph datasets with statistical and topological heterogeneity show that FedSeed consistently outperforms state-of-the-art baselines across client sizes and models.

REFERENCES

Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1396–1415. PMLR, 2023. URL https://proceedings.mlr.press/v202/baek23a.html.

Haoyu Chen, Zening Zhao, Jinsong Wang, Kai Shi, Zongpu Wei, and Jianhao Li. GraphP-FL: Personalized federated graph learning via dynamic structure awareness and fisher information elastic alignment. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 19586–19597, 2020. URL https://arxiv.org/abs/2006.04088.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification, 2019. URL https://arxiv.org/abs/1909.06335.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 22118–22133, 2020.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

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

Gihun Lee, Minchan Jeong, Sangmook Kim, Jaehoon Oh, and Se-Young Yun. FedSOL: Stabilized orthogonal learning with proximal restrictions in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12512–12522, 2024. doi: 10.1109/CVPR52733.2024.01189.

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

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 2, pp. 429–450, 2020b. URL https://proceedings.mlsys.org/paper/2020/hash/38af86134b65d0f10fe33d30dd76442e-Abstract.html.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations (ICLR)*, 2020c. URL https://arxiv.org/abs/1907.02189.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. FedGTA: Topology-aware averaging for federated graph learning. *Proceedings of the VLDB Endowment*, 17(1):41–50, 2024. doi: 10.14778/3617838.3617842.

Junkang Liu, Fanhua Shang, Hongying Liu, Yuxuan Tian, Yuanyuan Liu, Jin Liu, Kewen Zhu, and Zhouchen Lin. FedAdamW: A communication-efficient optimizer with convergence and generalization guarantees for federated large models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23748–23756, 2026. doi: 10.1609/aaai.v40i28.39549.

Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):4279–4295, 2025. doi: 10.1109/TNNLS.2024.3360429.

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

Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. FedNI: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Transactions on Medical Imaging*, 42(7):2032–2043, 2023. doi: 10.1109/TMI.2022.3188728.

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)*, 2021a.

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)*, 2021b. URL https://arxiv.org/abs/2003.00295.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):3710–3722, 2021. doi: 10.1109/TNNLS.2020.3015958.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2018.

Changlong Shi, He Zhao, Bingjie Zhang, Mingyuan Zhou, Dandan Guo, and Yi Chang. FedAWA: Adaptive optimization of aggregation weights in federated learning using client vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30651–30660, 2025. doi: 10.1109/CVPR52734.2025.02854.

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1805.09767.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed SGD with slow momentum. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1910.00643.

Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-IID graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 18839–18852, 2021. URL https://arxiv.org/abs/2106.13423.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48. PMLR, 2016. URL https://proceedings.mlr.press/v48/yang16.html.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 6671–6682, 2021. URL https://arxiv.org/abs/2106.13430.

Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 4560–4570, 2022. doi: 10.1145/3534678.3539121.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 649–657, 2015.

Zhanting Zhou, KaHou Tam, Yiding Feng, Ziqiang Zheng, Zeyu Ma, and Yang Yang. FedIA: Towards domain-robust aggregation in federated graph learning, 2025. URL https://arxiv.org/abs/2509.18171.

Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. FedTAD: Topology-aware data-free knowledge distillation for subgraph federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5716–5724, 2024. doi: 10.24963/ijcai.2024/632.

A Setup and Notation

Client $k \in \{1, \dots, K\}$ holds a subgraph $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k, \mathbf{X}_k, \mathbf{Y}_k)$ with $n_k = |\mathcal{V}_k|$ nodes, row-normalized adjacency $\mathbf{A}_k$, and a training node set T_k. In a decoupled GNN the propagation is precomputed as $\mathbf{z}_k^{(h)} = \mathbf{A}_k^h \mathbf{X}_k$ for $h = 0, \dots, K$, and the concatenation $[\mathbf{z}_k^{(0)}; \dots; \mathbf{z}_k^{(K)}]$ is passed through a readout. For the analysis we use the linear readout with squared loss $\mathcal{L}_k(\mathbf{W}) = \frac{1}{2|T_k|} \sum_{t \in T_k} \|\mathbf{W}\mathbf{z}_t - \mathbf{y}_t\|^2$ as a surrogate for the cross-entropy objective of Equation equation 1; the nonlinear head follows by the chain rule at the same granularity. Writing $\boldsymbol{\delta}_t = \mathbf{W}\mathbf{z}_t - \mathbf{y}_t$ for the residual,

$$\frac{\partial \mathcal{L}_k}{\partial \mathbf{W}_{c,h}} = \frac{1}{|T_k|} \sum_{t \in T_k} \delta_{t,c}\, \mathbf{z}_t^{(h)}. \tag{16}$$

Local training uses Adam Kingma & Ba (2015) with persistent state $(\mathbf{m}_k, \mathbf{v}_k)$, where $\mathbf{m}_k$ (first moment, β_1=0.9) is a fast variable and $\mathbf{v}_k$ (second moment, β_2=0.999) is a slow one. The second moment is the empirical analog of $v_{k,(c,h)} = \mathbb{E}\big[(\partial \mathcal{L}_k / \partial \mathbf{W}_{c,h})^2\big]$ with bias correction dropped.

The server keeps one second moment per client. Let S_t be the participating set in round t, let $\mathbf{p}_k$ be the hop energy profile of client k, and let the topology-similarity weights be

$$w_{kj} \propto \exp\Big(-\frac{\|\mathbf{p}_k - \mathbf{p}_j\|^2}{2h^2}\Big), \qquad h \propto \sigma_w, \tag{17}$$

where σ_w is the observed dispersion of the client profiles in the current round. The server-side second moment is Equation equation 6,

$$\bar{\mathbf{v}}_k^{(t)} = \lambda\, \bar{\mathbf{v}}_k^{(t-1)} + (1-\lambda)\, \frac{\sum_{j \in S_t} w_{kj}\, \mathbf{v}_j^{(t)}}{\sum_{j \in S_t} w_{kj}}, \qquad \lambda = 0.9, \tag{18}$$

and is broadcast with the model. The client mixes it into its own second moment, Equation equation 9, leaving $\mathbf{m}_k$ untouched. Uploads and the broadcast carry the second moment in a hop-block form (Section B.3), about $(K{+}1) + L^-$ scalars against d parameters.

B Proofs

B.1 Factorization of the second moment

Assumption 1 (Residual–input separability)**.** *$\delta_t \perp \mathbf{z}_t$ to first order: writing $\delta_t = \bar{\delta}_t + (\partial\delta/\partial z)\Delta z_t$, the cross moment $\mathbb{E}[\delta_{t,c}\, \Delta z_t^{(h)}]$ vanishes at leading order.*

Assumption 2 (Task-factor balance)**.** *$\mathbb{E}[\delta_c^2] = \tau^2$ for every output coordinate c.*

Proof. Under Assumption 1 the cross terms between distinct samples vanish, so

$$\mathbb{E}\Big[\Big(\sum_t \delta_{t,c} \mathbf{z}_t^{(h)}\Big)^2\Big] = \sum_t \mathbb{E}[\delta_{t,c}^2]\, \mathbb{E}[(\mathbf{z}_t^{(h)})^2], \tag{19}$$

and Assumption 2 makes the factor τ^2 independent of c. Hence $v_{k,(c,h)} \propto \mathbb{E}[\|\mathbf{z}_k^{(h)}\|^2]$ up to the per-coordinate normalization, and the hop energy share of the second moment is Equation equation 4,

$$\frac{\sum_c v_{k,(c,h)}}{\sum_{c,h'} v_{k,(c,h')}} = \frac{\|\mathbf{A}_k^h \mathbf{X}_k\|_F^2}{\sum_{h'} \|\mathbf{A}_k^{h'} \mathbf{X}_k\|_F^2} = e_k(h), \tag{20}$$

which contains no n_k: the profile depends only on the multi-hop energy of the client's own subgraph. Dropping Assumption 2 perturbs the ratio by at most $(\kappa_\tau - 1)/(\kappa_\tau + 1)$, with $\kappa_\tau = \max_c \mathbb{E}[\delta_c^2] / \min_c \mathbb{E}[\delta_c^2]$; dropping Assumption 1 adds second-order cross moments bounded by $\|\partial\delta/\partial z\|^2 \mathbb{E}[\Delta z^2]$.

B.2 Distortion bound and the failure condition

Lemma 1 (Relative Lipschitz bound of $s \mapsto s^{-1/2}$)**.** *For $a, b > 0$,*

$$\frac{\left|a^{-1/2} - b^{-1/2}\right|}{b^{-1/2}} \leq \frac{|a-b|}{2\min(a,b)}. \tag{21}$$

Proof. Dividing by $b^{-1/2}$ gives $|\sqrt{b/a}-1|$. If $b \geq a$, write $u = \sqrt{b/a} \geq 1$; then $u-1 \leq (u^2-1)/2$ is the bound. If $b < a$, write $u = \sqrt{b/a} < 1$; then $1 - u \leq (1-u^2)/(2u^2)$ because $2u^2 \leq 1 + u$, i.e. $(2u+1)(u-1) \leq 0$. Both cases give the claim.

Proposition 1 (Update-space distortion)**.** *Comparing the Adam step under a mixed second moment v_s with the step under the client's own second moment v^*, the relative distortion obeys Equation equation 5,*

$$D = \frac{\mathbb{E}|u(v_s) - u(v^*)|}{\mathbb{E}|u(v^*)|} \leq \frac{1}{2}\,\mathbb{E}\left[\frac{|v_s - v^*|}{v^*}\right], \tag{22}$$

whenever $v_s \geq v^*/2$ coordinatewise; if $v_s < v^*/2$ on a set of coordinates of positive measure, the bound fails and D is unbounded.

Proof. The Adam step at a coordinate is $u(v) = \eta g/(\sqrt{v} + \epsilon)$. Omitting ϵ (or absorbing it, which only weakens the ratio),

$$\frac{|u(v_s) - u(v^*)|}{|u(v^*)|} = \frac{\left|v_s^{-1/2} - (v^*)^{-1/2}\right|}{(v^*)^{-1/2}} \leq \frac{|v_s - v^*|}{2\min(v_s, v^*)}, \tag{23}$$

by Lemma 1. When $v_s \geq v^*/2$ we have $\min(v_s, v^*) \geq v^*/2$, so the per-coordinate distortion is at most $\frac{1}{2}|v_s - v^*|/v^*$. The bound is pointwise, so it passes to the ratio of expectations: $A \leq cB$ coordinatewise with $B > 0$ implies $\mathbb{E}[A] \leq c\,\mathbb{E}[B]$. Where $v_s < v^*/2$ on a positive-measure set, $v_s^{-1/2}$ can exceed $(v^*)^{-1/2}$ by an arbitrarily large factor, and D is unbounded. This is the regime in which replacing a client's second moment by a structureless mean is destructive; the mixed coefficient of Section B.5 prices it per coordinate.

B.3 Compression bound

Proposition 2 (Hop-block sufficiency)**.** *Transmit one scalar per hop block, the root mean square of v^2 within the block, and one scalar per remaining tensor. Under Assumptions 1 and 2 with isotropic within-block feature second moments, v is exactly block-constant and the reconstruction error vanishes. In general, writing $v_{k,(c,h)} = c_h(1 + \varepsilon_{k,(c,h)})$ with $\mathbb{E}[\varepsilon] = 0$, the update-space distortion is Equation equation 8,*

$$D(\hat{\mathbf{v}}) \leq \frac{1}{2} \cdot \frac{\kappa_Z - 1}{\kappa_Z + 1}, \tag{24}$$

where κ_Z is the within-block anisotropy of the feature second moment. The bound carries no ambient dimension d.

Proof. Under Assumptions 1 and 2 with isotropic within-block feature second moments, $v_{k,(c,h)} \propto \mathbb{E}[\|\mathbf{z}^{(h)}\|^2]$ is independent of c, so v is block-constant and the reconstruction error is zero. In general, for a nonnegative variable with condition number κ_Z, the mean absolute relative deviation satisfies $\mathbb{E}[\varepsilon] \leq (\kappa_Z - 1)/(\kappa_Z + 1)$; substituting into Proposition 1 gives the bound. Because it sees only the within-block spread, it is independent of the number $C \cdot d$ of coordinates in a block, so the channel stays sufficient as the model grows. Block averaging also reduces the per-coordinate sampling noise of order $\sigma_n/|T_k|^{1/2}$ in proportion to the within-block average.

B.4 Estimation-level shrinkage and the safety boundary

Fix client k and a coordinate; write v_k for the true second moment there, $\bar{\mathbf{v}}_k$ for the server-side second moment sent to client k, and $\Delta_k = \bar{\mathbf{v}}_k - v_k$ for the mismatch of Equation equation 2.

Proposition 3 (Estimation-level optimum). *Treat the bias-corrected local moving average $\hat{v}_k$ and the server-side second moment $\hat{\bar{v}}_k$ as estimators with $\mathbb{E}[\hat{v}_k] = v_k$, $\mathbb{E}[\hat{\bar{v}}_k] = \bar{\mathbf{v}}_k$, $\mathrm{Var}(\hat{v}_k) = \mathrm{Var}_{l,k}$, $\mathrm{Var}(\hat{\bar{v}}_k) = \mathrm{Var}_a$ of Equation equation 7, and $\mathrm{Cov}(\hat{v}_k, \hat{\bar{v}}_k) = 0$ to first order. The mixture $v_{\mathrm{new}}(\beta) = \beta\hat{v}_k + (1-\beta)\hat{\bar{v}}_k$ has Equation equation 10 and is minimized at Equation equation 11,*

$$\mathrm{MSE}_k(\beta) = \beta^2 \mathrm{Var}_{l,k} + (1-\beta)^2\big(\mathrm{Var}_a + \Delta_k^2\big), \tag{25}$$

$$\beta_k^* = \frac{\mathrm{Var}_a + \Delta_k^2}{\mathrm{Var}_{l,k} + \mathrm{Var}_a + \Delta_k^2} \in [0,1]. \tag{26}$$

Proof. Write $v_{\mathrm{new}} - v_k = \beta(\hat{v}_k - v_k) + (1-\beta)(\hat{\bar{v}}_k - \bar{\mathbf{v}}_k) + (1-\beta)\Delta_k$. The first two terms are zero-mean and uncorrelated, the third is deterministic, giving the MSE. Differentiating, $\partial_\beta \mathrm{MSE}_k = 2\beta\mathrm{Var}_{l,k} - 2(1-\beta)(\mathrm{Var}_a + \Delta_k^2)$, which vanishes at β_k^*; the second derivative is positive and $\beta_k^* \in [0,1]$. At $\beta = 0$ the objective is $\mathrm{Var}_a + \Delta_k^2$ and at $\beta = 1$ it is $\mathrm{Var}_{l,k}$, so the mixture is never worse than either pure policy.

Remark 1 (Regimes). *If $\Delta_k = 0$ and $|S_t| r_{\mathrm{eff}} \gg 1$ then $\beta_k^* \to 0$: overwriting the local second moment with the server-side mean is optimal, which is the reset of FedAdamW Liu et al. (2026). If $\Delta_k^2 \gg \mathrm{Var}_{l,k}, \mathrm{Var}_a$ then $\beta_k^* \to 1$: pure local protection. A fixed coefficient $\bar{\beta}$ is optimal only on the surface $\mathrm{Var}_a + \Delta_k^2 = \frac{\bar{\beta}}{1-\bar{\beta}}\mathrm{Var}_{l,k}$, so no constant is uniformly correct.*

B.5 Update-level risk and the deployed coefficient

The seeded value is also the denominator of the Adam step, so the gain appears in update space. Fix a client and a coordinate; let $g = s + \xi$ with $\mathbb{E}[\xi] = 0$ and $\mathbb{E}[\xi^2] = \sigma^2$. With normalizer $u = v(\beta)^{-1/2}$ the step is $ug = us + u\xi$; measuring it against the step the client would take under its own second moment, $u_k = v_k^{-1/2}$, Equation equation 12,

$$R_k(\beta) = \underbrace{s^2(u - u_k)^2}_{\text{distortion}} + \underbrace{\sigma^2 u^2}_{\text{noise floor}}, \tag{27}$$

$$u = v(\beta)^{-1/2}, \tag{28}$$

$$v(\beta) = \beta v_k + (1-\beta)\bar{\mathbf{v}}_k. \tag{29}$$

This is an identity: the cross term vanishes because $\mathbb{E}[\xi] = 0$. With m the first moment and v_k the second moment of the local Adam state, s is estimated by m and σ^2 by $\hat{\sigma}^2 = \max(v_k - m^2, 0)$, since $\mathbb{E}[g^2] - \mathbb{E}[g]^2 = v_k - m^2$. The deployed coefficient is $\beta_k^{\star\star} = \arg\min_{\beta\in[0,1]} R_k(\beta)$. All four inputs are pre-existing, so the rule introduces no hyperparameter.

Proposition 4 (Closed form). *Write $s = m^2$, $\sigma^2 = \hat{\sigma}^2$, $u_k = v_k^{-1/2}$, $u_0 = \bar{\mathbf{v}}_k^{-1/2}$, and let $u^\dagger = \mathrm{clip}(u^\star, \min(u_0, u_k), \max(u_0, u_k))$ with $u^\star = s\,u_k/(s+\sigma^2)$. Then*

$$v_k(\beta^{\star\star}) = u^{\dagger\,-2} = \mathrm{clip}\left(v_k\Big(1 + \frac{\sigma^2}{s}\Big)^2,\, v_{\min},\, v_{\max}\right), \tag{30}$$

with $v_{\min} = \min(v_k, \bar{\mathbf{v}}_k)$, $v_{\max} = \max(v_k, \bar{\mathbf{v}}_k)$, and on an interior solution Equation equation 14,

$$\beta_k^{\star\star} = \frac{u^{\dagger\,-2} - \bar{\mathbf{v}}_k}{v_k - \bar{\mathbf{v}}_k} = \frac{v_k - v_k(1+\sigma^2/s)^2}{v_k - \bar{\mathbf{v}}_k}, \tag{31}$$

pinned at 0 or 1 otherwise. The rule depends on the optimizer state only through $\rho = s/\sigma^2$, so it is invariant to the gradient scale.

Proof. With $u = v(\beta)^{-1/2}$ the risk is $R = (mu - mu_k)^2 + \sigma^2 u^2 = (s+\sigma^2)u^2 - 2su_k u + su_k^2$, a strictly convex quadratic in u minimized at $u^\star = s\,u_k/(s+\sigma^2)$. As β ranges over $[0,1]$, $v(\beta)$ moves monotonically between $\bar{\mathbf{v}}_k$ and v_k, so u ranges monotonically over the interval between u_0 and u_k; minimizing a convex function of u over that interval gives the clip, hence $u^\dagger$ and equation 30. Solving $v(\beta) = u^{\dagger\,-2}$ for an interior solution gives equation 31. If $m = 0$ then $s = 0$ and $R = \sigma^2 u^2$ is minimized by the smallest admissible u, i.e. $v = v_{\max}$; if $\sigma^2 = 0$ then $u^\star = u_k$ and the mixture is a no-op; the clip covers both cases.

Corollary 1 (The rule is protective). *$v_k(\beta^{\star\star}) \geq v_k$, so the realized step magnitude $|m|/\sqrt{v_k(\beta^{\star\star})}$ is never larger than the local one.*

(a) If $\bar{\mathbf{v}}_k = v_k$ the mixture is inert for every β.

(b) As $\rho \to \infty$ the target collapses to v_k and $\beta_k^{\star\star} = 1$: reliable coordinates are untouched.

(c) As $\rho \to 0$ the target grows without bound and the clip selects the endpoint on the server side, so the noise term is evaluated at a second moment no smaller than the local one.

(d) The server-side second moment enters only through the clip, so it can bound the window but never override a defensible local value.

Proof. $(1 + \sigma^2/s)^2 \geq 1$ and clipping to $[v_{\min}, v_{\max}]$ with $v_{\min} = \min(v_k, \bar{\mathbf{v}}_k)$ returns a value at least v_k; hence $v_k(\beta^{\star\star}) \geq v_k$. (a): $v_{\min} = v_{\max} = v_k$. (b): as $\rho \to \infty$ the target tends to v_k, whose preimage is $\beta_k^{\star\star} = 1$. (c): as $\rho \to 0$ the target tends to $+\infty$, so the clip returns $v_{\max}$, the endpoint on the server side. (d) holds because $v_k(\beta^{\star\star})$ is a function of v_k and ρ alone, clipped by $\bar{\mathbf{v}}_k$.

Corollary 2 (Return after absence). *Suppose a client is absent for T rounds. On return its first moment $\mathbf{m}_k$ is fast (β_1=0.9) and reflects the current gradients, while its second moment $\mathbf{v}_k$ is slow (β_2=0.999) and still carries the previous rounds, so $\hat{\sigma}^2 = v_k - m^2$ and $\rho = s/\sigma^2$ are formed from functionals of different distributions. The server-side second moment is updated every round by the participating clients and smoothed over $r_{\text{eff}} = (1+\lambda)/(1-\lambda) = 19$ rounds, so it tracks the current federation. The rule therefore re-anchors the stale second moment without a staleness detector, extra state, or additional communication. The sign of the displacement is regime-dependent and is not fixed by the theory.*

B.6 Elastic aggregation

Proposition 5 (Limit behaviour). *The aggregate*

$$\bar{\mathbf{v}}_k = \frac{\sum_j w_{kj}\, \mathbf{v}_j}{\sum_j w_{kj}}, \qquad w_{kj} \propto \exp\Big(- \frac{\|\mathbf{p}_k - \mathbf{p}_j\|^2}{2h^2} \Big), \tag{32}$$

is continuous in the profiles. If all profiles coincide, the weights are uniform and the aggregate equals the uniform mean of $\{\mathbf{v}_j\}_{j \in S_t}$. For well-separated profiles and small h, the aggregate concentrates on clients with profiles closest to $\mathbf{p}_k$.

Proof. The kernel is continuous and normalized, so the aggregate is continuous. If $\mathbf{p}_k = \mathbf{p}_j$ for all $j \in S_t$, every distance is zero, every weight is equal, and the normalized aggregate is the uniform mean. As $h \to 0$ with a positive gap between the profile of k and those of the remaining clients, the weights of the closest clients dominate and the aggregate tends to the mean over the closest profiles.

Remark 2 (Bandwidth calibration). *Setting $h \propto \sigma_w$, the observed dispersion of the client profiles, scales the kernel with the federation's structural spread: when the profiles are close the aggregate approaches the uniform mean, and when they are spread the aggregate concentrates on close profiles. Because $\mathbf{p}_k$ travels on the same channel as the model, the aggregation adds no descriptor and no extra byte. This is a calibration choice that fixes the interpolation between the uniform mean and a local average; it is not tuned per dataset.*

C LLM USE

Large language models (LLMs) were used to improve the readability of the manuscript, including language refinement and stylistic polishing. The research methodology, findings, and results are solely the authors' work. The authors take full responsibility for all parts of the manuscript.