FEDSUCO: FEDERATED GRAPH CLUSTERING VIA RIEMANNIAN GEOMETRY-GUIDED SUPERNODE COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph clustering requires learning across fragmented graph structures while keeping individual-node information local. We propose FedSuCo, which uses supernodes guided by Riemannian geometry as the unit of collaboration. Specifically, in FedSuCo, clients learn Riemannian node representations and construct supernodes through geometric renormalization, preserving coarse structural organization while retaining clustering-relevant angular similarity. Supernode representations and securely aggregated connectivity statistics jointly form a weighted global quotient graph, on which the server learns structural guidance and transfers it back to local nodes. Final federated spherical clustering aggregates only cluster-level sufficient statistics, while node representations and assignments remain local. Experiments on five real-world graph datasets against thirteen federated baselines show consistent improvements across clustering metrics.

1 INTRODUCTION

Attributed graphs are often distributed across multiple organizations, each of which holds a private subgraph whose node attributes and local relations cannot be centralized (Liu et al., 2025b; Fu et al., 2022). Federated learning enables collaborative model training while keeping data local (McMahan et al., 2017). Unlike conventional federated learning, however, partitioning graph data across clients does not merely decentralize samples. It also fragments the graph topology that defines how nodes interact. Consequently, cross-client connections cannot be directly incorporated into local message passing, leaving each client with only a partial view of the underlying graph (Liu et al., 2025a; ZHANG et al., 2021). As a result, server-side aggregation of locally computed GNN updates cannot explicitly recover the fragmented cross-client structural context. This limitation is particularly critical for unsupervised node clustering, where no label supervision is available to provide a shared semantic signal across clients, making it more difficult to learn consistent node representations and cluster assignments from fragmented local graphs (Fu et al., 2025).

Existing unsupervised federated graph methods enable collaboration through different mechanisms. GCFGAE coordinates forward and backward computations across overlapping vertices (Guo et al., 2023). FedPAM aligns client representation spaces using learnable anchors and adaptively integrates model parameters (Fu et al., 2025). FedNCN takes a further step by inferring representative samples from local clustering signals, reconstructing cross-subgraph links on the server, and learning consensus prototypes on the resulting graph (Liu et al., 2025a). These efforts explore cross-client collaboration through different forms of shared information (e.g., overlapping vertices, representative samples), yet their structural collaboration largely remains tied to individual nodes or node-level proxies. This exposes an inherent dilemma between exploiting cross-client structural information and keeping individual-node information local, and motivates a broader question: *can global structural knowledge be learned through higher-level structural units while keeping individual-node information local?*

To enable collaboration through such higher-level structural units, their construction should preserve meaningful graph organization rather than simply aggregate arbitrary nodes. Network geometry (Boguñá et al., 2021) provides a basis for constructing such units. By representing graph structure in a geometric space, it provides a principled way to characterize and organize structural

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.

relationships among nodes. Recent advances in graph foundation models have further highlighted Riemannian geometry as a principled framework for capturing structural patterns across diverse graphs (Sun et al., 2024; Yu & Sun, 2026; Chami et al., 2019). Riemannian manifolds provide flexible geometric spaces in which graph structure can be characterized through distances, directions, and curvature, making them well suited for organizing structurally related nodes. More importantly, geometric renormalization (García-Pérez et al., 2018; Jankowski et al., 2023) provides a natural mechanism for turning geometric organization into multiscale structure by grouping nearby nodes into higher-level units while preserving network organization across scales.

Motivated by these observations, we use geometrically constructed supernodes as the units of federated structural collaboration, and propose **FedSuCo**, a federated graph clustering framework that couples geometry-guided local condensation with global structural collaboration over supernodes. In FedSuCo, clients learn node representations in a common Riemannian space while keeping individual-node representations and node-to-supernode assignments local. Following the renormalization view (García-Pérez et al., 2018; Jankowski et al., 2023), the underlying geometry condenses geometrically nearby nodes into supernodes while preserving coarse structural organization. Cross-client edges, which are unavailable for direct local propagation, are recorded by their endpoint owners and are securely aggregated only into supernode-level connectivity statistics. Together with supernode representations, these statistics form a weighted quotient graph for global structural learning. The learned structural information is then transferred to local node representations through locally retained node-to-supernode assignments. For final clustering, clients share only aggregated cluster statistics for global updates, while node-level assignments remain local. In this process, Riemannian geometry provides a common structural basis across FedSuCo: it shapes local representations Yu & Sun (2026); Chami et al. (2019), guides node condensation from the view of geometric renormalization (Jankowski et al., 2023), and retains the angular component that captures clustering-relevant node similarity for final clustering (Wang et al., 2019; Yang et al., 2023).

Our contributions are as follows:

- We introduce a geometry-guided federated supernode collaboration framework that elevates supernodes to the basic units of cross-client structural learning, enabling global structural collaboration without exposing individual-node information.
- We develop a Riemannian geometry-guided condensation mechanism that combines geometric renormalization with angular information from learned representations, preserving coarse structural organization while retaining clustering-relevant node similarity in the resulting supernodes.
- We develop a two-server secure computation prototype that realizes the cross-client aggregation and federated clustering of FedSuCo, and conduct experiments on five real-world datasets against thirteen baselines, showing improved clustering performance over these baselines.

2 Background and Related work

Deep Graph Clustering. Deep graph clustering learns node partitions without labels through representation learning and structure optimization. DCRN mitigates representation collapse by reducing sample- and feature-level correlations (Liu et al., 2022b). SUBLIME optimizes graph topology through contrastive learning and structure bootstrapping (Liu et al., 2022a), while DeSE jointly optimizes graph structure, soft cluster assignments, and differentiable structural entropy (Zhang et al., 2025). These methods primarily address centralized graphs and do not directly handle cross-client structural collaboration under graph partitioning.

Federated Graph Learning. Generic federated methods coordinate training through parameter or gradient aggregation (McMahan et al., 2017; Li et al., 2020; Liao et al., 2024). GCFGAE combines federated and split learning to coordinate layer-wise computations over overlapping vertices, aiming for consistency with a centralized graph autoencoder (Guo et al., 2023). FedPAM aligns representations through learnable anchors and fused Gromov–Wasserstein optimal transport, and integrates model parameters in a low-rank tensor space (Fu et al., 2025). FedNCN infers representative samples from local clustering signals, mends cross-subgraph links, and learns consensus prototypes (Liu et al., 2025a). FedSuCo performs structural collaboration through geometrically condensed supernodes and aggregated edge counts on a weighted quotient graph. Clients apply supernode-

level feedback through private memberships, retaining node representations and node-to-supernode assignments locally.

Riemannian Geometry and Renormalization HGCN and constant-curvature GCNs extend graph convolution to non-Euclidean spaces, introducing geometric inductive biases into graph representation learning (Chami et al., 2019; Bachmann et al., 2020). Topology-based geometric embedding infers hyperspherical similarity coordinates and popularity-related hidden degrees (Jankowski et al., 2023). LSEnet combines Lorentz geometry and differentiable structural information to learn hierarchical partitions (Sun et al., 2024). Geometric renormalization constructs multiscale networks by aggregating geometrically proximate nodes, providing a basis for coarsening that retains coarse network organization (García-Pérez et al., 2018). These studies establish foundations for geometric representation and multiscale modeling. FedSuCo applies geometry-guided local condensation to federated quotient-graph collaboration and node-level feedback.

3 Problem Setup and Threat Model

3.1 Federated Graph Clustering

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $N = |\mathcal{V}|$ and $\mathbf{X} \in \mathbb{R}^{N \times d_x}$. Its nodes are partitioned into disjoint client sets $\mathcal{V} = \bigcup_{c=1}^{C} \mathcal{V}_c$, with $n_c = |\mathcal{V}_c|$ and $N = \sum_c n_c$. We denote within-client edges and edges between clients c and c' by

$$\mathcal{E}_c^{\text{in}} = \{\{u, v\} \in \mathcal{E} : u, v \in \mathcal{V}_c\}, \qquad \mathcal{E}_{cc'} = \{\{u, v\} \in \mathcal{E} : u \in \mathcal{V}_c,\ v \in \mathcal{V}_{c'}\}, \quad c \neq c'. \tag{1}$$

Client c holds $\mathcal{D}_c = (\mathbf{X}_c, \mathcal{E}_c^{\text{in}}, \mathcal{E}_c^{\times})$, where $\mathcal{E}_c^{\times} = \bigcup_{c' \neq c} \mathcal{E}_{cc'}$. Cross-client edges represent inter-organizational relationships, such as interbank transfers, interhospital referrals, or intercompany transactions. Given K clusters, the goal is to obtain cluster assignments with consistent identities across clients; each client retains its node predictions $\hat{\mathbf{y}}_c$ locally. Ground-truth labels $y_{c,i}^{\text{gt}}$ are used only for evaluation. Appendix A summarizes the notation.

3.2 System and Threat Model

Clients collaborate through two servers: S_0 performs global model updates and supergraph learning, while S_1 assists with secure computation. For existing cross-client edges, we assume that their endpoint owners have corresponding edge records.

We adopt a semi-honest model in which clients and servers follow the protocol. An adversary may passively corrupt at most one server: $\mathcal{A} \subseteq \{S_0, S_1\}$ and $|\mathcal{A}| \leq 1$. The corrupted server can inspect its received messages, local state, public metadata, and authorized outputs.

3.3 Privacy Objectives and Disclosure Boundary

Our privacy objective is to protect client inputs and intermediate values while releasing the results needed for global collaboration. Let $\mathcal{R} = (\mathcal{R}_{\text{coarse}}, \mathcal{R}_{\text{train}}, \mathcal{R}_{\text{cluster}}, \mathcal{R}_{\text{eval}})$ denote the authorized releases at the four stages:

Authorized releases are visible only to their designated protocol recipients. Public metadata $\mathcal{M}_{\text{pub}}$ includes participant identities, client node and supernode counts, message dimensions, cross-edge batch sizes, iteration counts, and public configurations. Privacy is conditional on these releases: minimum supernode size controls disclosure granularity but does not, by itself, provide differential privacy or prevent inference from disclosed results.

4 Method

4.1 Framework Overview

Parameter aggregation alone does not explicitly establish structural relationships between nodes on different clients. Directly sharing node representations, adjacency, or node-to-supernode memberships would exceed the disclosure boundary in Section 3.3. FedSuCo therefore uses geometrically

Table 1: Privacy and disclosure boundary of FedSuCo.

Stage	Client-private data	Authorized releases
Coarsening	$\mathcal{D}_c$, node embeddings and directions, and assignment matrix $\mathbf{P}_c$	$\mathcal{R}_{\text{coarse}}$: supernode directions, sizes, aggregated hidden degrees, and coarse edge counts
Training	Individual client gradient $\mathbf{g}_c$	$\mathcal{R}_{\text{train}}$: aggregate gradients and diagnostics, shared parameters, and supernode-level feedback
Clustering	Node responsibilities r_{cij}, client-specific sufficient statistics $(N_{cj}, \mathbf{S}_{cj})$, and node predictions	$\mathcal{R}_{\text{cluster}}$: global sufficient statistics $N_j = \sum_c N_{cj}$ and $\mathbf{S}_j = \sum_c \mathbf{S}_{cj}$, mixture parameters, and aggregate initialization scores
Evaluation	Node ground-truth labels and predictions	$\mathcal{R}_{\text{eval}}$: global contingency table and derived metrics

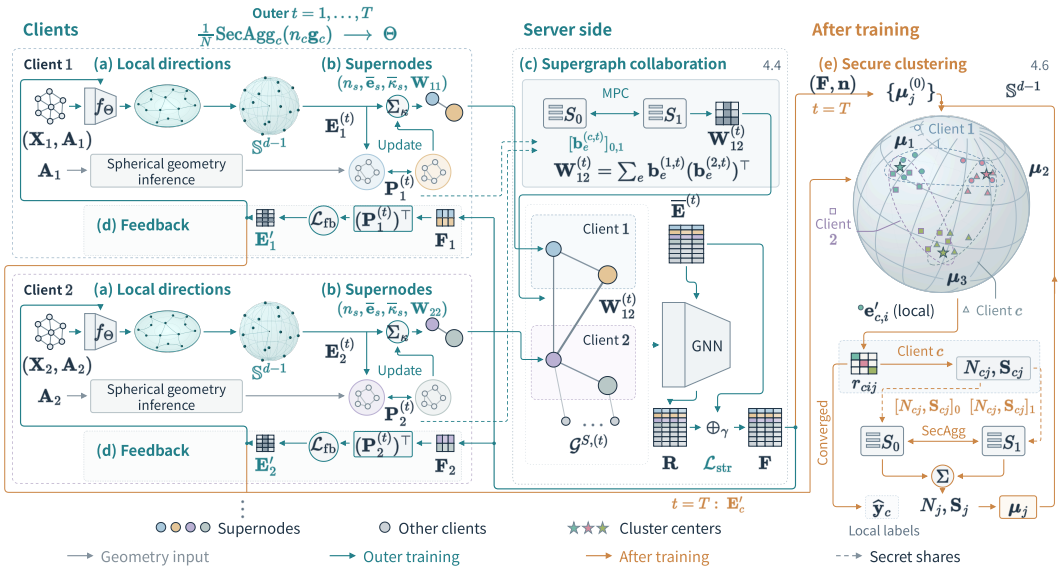

Figure 1: Overview of FedSuCo. Clients construct geometry-guided supernodes for global structural collaboration, receive supernode-level feedback, and securely aggregate sufficient statistics for federated spherical clustering.

constructed supernodes as the unit of structural collaboration, allowing the server to model cross-client connectivity through coarse representations and edge counts.

FedSuCo proceeds in three stages. Clients first learn hyperbolic representations and extract spherical directions. They then construct supernodes and release their sizes, directions, hidden degrees, and local coarse edge statistics; cross-client superedge counts are obtained through secure computation. The server learns structural corrections on the global weighted supergraph and returns refined supernode directions for local node-level feedback. These learning, coarsening, and feedback stages repeat over outer rounds $t = 1, \ldots, T$. After the final round, clients perform global spherical clustering by sending secret shares of sufficient statistics to the two servers, while retaining node-level inference locally.

4.2 Local Riemannian Direction Learning

The local encoder learns hyperbolic embeddings, from which we extract spherical directions for subsequent coarsening and clustering. Let $\mathbf{A}_c$ be the local adjacency induced by the within-client edges $\mathcal{E}_c^{\text{in}}$. Client c computes $\mathbf{Z}_c = f_\Theta(\mathbf{X}_c, \mathbf{A}_c)$ and extracts unit directions $\mathbf{e}_{c,i} = \text{norm}(\mathbf{z}_{c,i})$, where $\text{norm}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|_2$ for nonzero $\mathbf{v}$. These directions lie on $\mathbb{S}^{d-1}$ and form the rows of

$\mathbf{E}_c \in \mathbb{R}^{n_c \times d}$. All clients share the encoder parameters, while message passing uses only local edges.

Training combines feature and link reconstruction through $\mathcal{L}_c^{\text{loc}} = \lambda_{\text{feat}}\mathcal{L}_{\text{feat}}^c + \lambda_{\text{link}}\mathcal{L}_{\text{link}}^c$. To make link reconstruction depend on angular relationships, the link decoder places the unit directions on a fixed hyperbolic shell with a common radius. Structural supervision therefore acts directly on the directions used downstream, without relying on differences in embedding radii. The shared model is optimized through secure gradient aggregation weighted by client node counts. Encoder, decoder, and optimization details are provided in Appendix B.

4.3 Geometry-Guided Supernode Construction

Structure-aware grouping. Coarsening solely by learned directional similarity does not explicitly account for node popularity or structural geometry. We therefore adopt a spherical geometry-guided renormalization strategy. Each client infers spherical positions $\mathbf{y}_{c,i} \in \mathbb{S}^{d-1}$, hidden degrees $\kappa_{c,i}$, and a geometric parameter β_c from its local topology (Jankowski et al., 2023). Guided by geometric renormalization (García-Pérez et al., 2018), we form supernodes around high-hidden-degree seeds according to geodesic proximity. Remaining nodes are assigned to existing groups, ensuring that every node is covered and every supernode contains at least b nodes. The grouping procedure is detailed in Appendix C.

Grouping and representation aggregation. The inferred geometry determines group membership, whereas the released directions are constructed from the learned node representations. For supernode $\mathcal{V}_{c,s}$, we use hidden-degree-weighted aggregation:

$$\overline{\mathbf{e}}_{c,s} = \frac{\sum_{i \in \mathcal{V}_{c,s}} \kappa_{c,i}\mathbf{e}_{c,i}}{\left\| \sum_{i \in \mathcal{V}_{c,s}} \kappa_{c,i}\mathbf{e}_{c,i} \right\|_2}. \tag{2}$$

The group has size $n_{c,s} = |\mathcal{V}_{c,s}|$ and hidden degree $\overline{\kappa}_{c,s} = (\sum_{i \in \mathcal{V}_{c,s}} \kappa_{c,i}^{\beta_c})^{1/\beta_c}$. These statistics and within-group and between-group edge counts form the coarse payload specified in Section 3.3. The client retains its membership matrix $\mathbf{P}_c \in \{0,1\}^{m_c \times n_c}$ for subsequent local feedback, where m_c is its number of supernodes and $[\mathbf{P}_c]_{s,i} = \mathbf{1}[i \in \mathcal{V}_{c,s}]$.

4.4 Global Weighted Supernode Collaboration

Recovering cross-client structure. Local coarsening preserves only within-client connectivity. For clients $c < c'$, the number of edges between their supernodes s and t is

$$[\mathbf{W}_{cc'}]_{s,t} = \sum_{\{u,v\} \in \mathcal{E}_{cc'}} \mathbf{1}[u \in \mathcal{V}_{c,s}]\mathbf{1}[v \in \mathcal{V}_{c',t}]. \tag{3}$$

Under the system model in Section 3.2, the servers securely compute these aggregate counts using the protocol in Appendix E. Combining them with local coarse statistics yields the global weighted supernode graph $\mathcal{G}^S = (\mathcal{V}^S, \mathbf{W})$, with $M = |\mathcal{V}^S| = \sum_c m_c$ supernodes. Superedge weights retain the number of underlying edges, preserving cross-client connectivity strength at the coarse level.

Size- and popularity-aware structural learning. A weighted graph encoder produces structural proposals $\mathbf{r}_s$, which are combined with the initial directions through gated residual updates, $\mathbf{f}_s = \text{norm}(\gamma\overline{\mathbf{e}}_s + (1-\gamma)\mathbf{r}_s)$. The gate $\gamma \in (0,1)$ controls how strongly global propagation modifies the initial representations. Since superedge counts depend on group size, popularity, and directional relationships, we adopt a binomial model:

$$W_{st} \sim \text{Binomial}(T_{st}, p_{st}), \quad \text{logit}(p_{st}) = \xi\mathbf{f}_s^\top\mathbf{f}_t + \log\overline{\kappa}_s + \log\overline{\kappa}_t + b_{\text{str}}. \tag{4}$$

Here, T_{st} is the number of possible underlying node pairs, with $T_{st} = n_s n_t$ for distinct supernodes; $\xi > 0$ and b_{str} are learnable. This formulation distinguishes connection opportunities and node popularity from directional similarity. The server minimizes a normalized binomial negative log-likelihood with directional alignment and structural regularization to obtain the refined directions $\mathbf{F} \in \mathbb{R}^{M \times d}$. The encoder and complete objective are given in Appendix D.

4.5 Coarse-to-Fine Feedback and Training

The server returns the refined directions $\mathbf{F}_c \in \mathbb{R}^{m_c \times d}$ associated with client c. The client uses its membership matrix to lift these coarse targets to individual nodes and constructs a directional feedback loss:

$$\widetilde{\mathbf{F}}_c = \mathbf{P}_c^\top \mathbf{F}_c, \qquad \mathcal{L}_{\mathrm{fb}}^c = \frac{1}{n_c} \sum_i \left(1 - \mathbf{e}_{c,i}^\top \widetilde{\mathbf{f}}_{c,i}\right). \tag{5}$$

The client optimizes $\mathcal{L}_c^{\mathrm{refine}} = \mathcal{L}_c^{\mathrm{loc}} + \lambda_{\mathrm{fb}} \mathcal{L}_{\mathrm{fb}}^c$, transferring cross-client structural information to node representations. Feedback acts as soft supervision alongside local reconstruction and does not require nodes within a supernode to share identical representations. The lift and node-level optimization are evaluated locally, producing refined directions $\mathbf{E}'_c$ for clustering.

4.6 Privacy-Preserving Federated Clustering

Coarse initialization and node-level inference. The server initializes K cluster directions using the global supernode directions $\mathbf{f}_s$, weighted by their sizes n_s. Clustering is then performed over the original node directions retained by clients, so supernodes provide global initialization without determining their members' final labels. FedSuCo supports spherical k-means and von Mises–Fisher (vMF) mixtures (Banerjee et al., 2005) through a unified responsibility variable r_{cij}: spherical k-means uses the hard assignment $r_{cij} = \mathbf{1}[j = \arg\max_h \boldsymbol{\mu}_h{}^\top \mathbf{e}'_{c,i}]$, whereas vMF uses the corresponding soft posterior.

Secure updates through sufficient statistics. Both clustering variants require only the effective count and directional sum of each component:

$$(N_j, \mathbf{S}_j) = \mathrm{SecAgg}_c \left(\sum_i r_{cij},\ \sum_i r_{cij} \mathbf{e}'_{c,i} \right). \tag{6}$$

The operator SecAgg reconstructs only the sum of client contributions under the disclosure policy in Section 3.3. Both variants update the cluster direction as $\boldsymbol{\mu}_j = \mathbf{S}_j / \|\mathbf{S}_j\|_2$; vMF additionally updates its mixture weights and concentrations. After convergence, clients derive their node assignments locally. The aggregation protocol is given in Appendix E, and the complete clustering updates are provided in Appendix F.

5 Experiments and Analysis

5.1 Experimental Setup

Datasets and federated partitioning. We evaluate FedSuCo on five graph datasets: Cora and CiteSeer (Sen et al., 2008), PubMed (Namata et al., 2012), and Amazon Photo and Amazon Computers (Shchur et al., 2019). For Cora, we retain its largest connected component. Datasets are loaded with PyTorch Geometric (Fey & Lenssen, 2019), using the Planetoid benchmark versions (Yang et al., 2016) for the citation networks. We partition each graph using METIS (Karypis & Kumar, 1998), with 10 clients by default. All methods use the same fixed METIS owner assignments for each dataset and client count, and node labels are excluded from representation learning and clustering optimization.

Baselines and evaluation metrics. We compare FedSuCo against thirteen federated baselines. Twelve baselines are constructed by combining four federated optimization frameworks (FedAvg, FedProx, FedSGD, and FedU^2) with three unsupervised graph clustering models (DCRN, DeSE, and SUBLIME). We additionally include GCFGAE (Guo et al., 2023) as a graph-specific federated baseline. We evaluate clustering performance using clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and macro-averaged F1 (Macro-F1), with higher values indicating better performance. FedSuCo uses spherical k-means (FedSuCo-SKM) (Hornik et al., 2012) and a federated von Mises–Fisher mixture model (FedSuCo-vMF) (Banerjee et al., 2005) for final clustering.

Implementation and evaluation protocol. For the main experiments, we fix the client partitions and conduct five independent training runs with different random seeds, reporting the mean and

Table 2: Main clustering results (%) over five random seeds. F1 denotes Macro-F1. Bold values indicate the best result, while underlined values indicate the strongest competing result for each metric.

	Cora				CiteSeer				PubMed			
Method	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
FedAvg + DCRN	57.38±6.20	42.76±4.37	33.58±6.36	52.46±5.41	46.13±5.17	20.32±3.08	19.02±3.32	42.68±6.04	49.76±4.53	11.13±5.09	8.28±6.64	44.03±8.41
FedAvg + DeSE	36.92±3.25	17.41±4.07	11.41±2.83	30.46±5.39	43.02±3.60	19.75±2.52	16.68±2.94	39.53±3.82	47.80±2.52	15.55±2.21	9.47±1.74	44.70±6.38
FedAvg + SUBLIME	44.03±2.55	27.54±1.67	18.43±1.22	40.20±2.68	43.79±3.39	24.03±1.58	19.57±1.77	41.59±2.90	59.97±0.07	21.27±0.18	18.95±0.15	60.38±0.10
FedProx + DCRN	52.09±5.19	37.07±3.80	27.83±4.64	47.99±8.25	41.12±4.57	15.65±2.56	14.32±3.62	37.76±3.34	56.95±6.71	16.08±7.50	15.87±8.26	52.63±11.25
FedProx + DeSE	37.09±3.53	17.87±2.01	11.94±1.64	30.03±4.10	43.08±2.11	19.18±1.77	16.24±2.17	33.32±3.23	45.81±2.18	13.29±1.27	8.64±2.69	41.86±6.93
FedProx + SUBLIME	40.37±1.68	22.92±1.16	13.89±1.15	38.67±2.11	42.33±1.73	22.60±1.10	18.44±1.04	40.34±1.46	55.78±2.02	20.88±0.91	15.86±1.97	55.01±2.69
FedSGD + DCRN	55.76±6.43	40.04±3.99	30.89±5.86	50.99±8.59	43.29±6.50	17.81±3.14	16.65±3.50	39.43±7.16	43.32±6.32	4.53±6.19	4.23±6.61	35.29±9.22
FedSGD + DeSE	32.12±3.64	13.31±2.89	7.73±3.30	22.85±3.40	30.49±1.93	8.77±2.90	6.20±1.93	23.45±4.25	41.49±1.89	0.77±1.05	0.77±1.78	24.75±6.30
FedSGD + SUBLIME	41.51±2.25	24.18±1.91	13.19±1.75	40.19±3.60	40.31±2.70	19.68±1.34	13.49±1.03	38.56±2.31	39.21±0.15	0.12±0.01	0.11±0.02	34.09±0.10
FedU^2 + DCRN	27.63±3.29	10.15±2.04	4.29±1.73	25.68±4.15	36.45±4.38	17.22±5.09	12.69±4.56	35.36±4.79	41.73±2.82	4.45±2.39	3.64±1.84	41.04±2.90
FedU^2 + DeSE	36.85±3.02	17.63±4.13	11.50±2.73	31.13±4.61	43.27±2.67	20.13±2.26	16.91±2.51	39.83±3.33	49.77±2.61	15.84±2.08	10.68±1.23	48.31±5.61
FedU^2 + SUBLIME	49.34±3.00	32.18±2.36	22.92±3.02	45.78±4.97	48.33±3.85	28.06±1.65	23.42±2.71	44.78±3.18	60.24±0.05	21.50±0.08	19.08±0.17	60.68±0.04
GCFGAE†	60.10±8.94	49.20±4.53	36.84±8.60	59.16±10.68	34.91±4.09	18.04±4.23	6.84±2.01	31.61±4.51	60.88±2.96	25.40±2.58	21.13±3.62	60.57±2.54
FedSuCo-SKM	73.80±4.19	**56.66±2.20**	**52.82±4.99**	70.86±5.79	58.49±2.81	**32.69±1.96**	**31.14±3.47**	54.55±2.50	**66.63±1.11**	**25.96±1.29**	**26.44±1.70**	**66.84±0.97**
FedSuCo-vMF	**74.13±1.89**	56.20±1.64	52.49±2.98	**71.11±3.80**	**58.77±2.59**	32.45±1.64	30.81±2.18	**55.48±2.23**	64.69±1.34	23.68±1.27	23.40±1.90	64.84±1.24

	Amazon Photo				Amazon Computers			
Method	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
FedAvg + DCRN	63.71±2.68	54.02±3.28	42.67±3.29	59.87±2.67	41.47±5.22	34.73±5.74	20.29±5.87	26.97±7.81
FedAvg + DeSE	48.13±6.30	33.19±4.64	27.01±7.77	32.55±5.12	38.53±3.58	26.90±2.98	14.59±3.90	24.33±3.22
FedAvg + SUBLIME	51.23±4.36	44.94±2.36	30.06±5.16	43.50±3.01	39.19±1.23	37.00±0.64	18.25±0.48	30.75±0.64
FedProx + DCRN	64.49±2.54	54.93±2.57	42.43±2.32	61.43±1.85	39.67±3.28	35.72±2.68	18.85±3.36	26.55±5.35
FedProx + DeSE	50.30±6.72	35.49±6.73	30.37±9.10	33.57±6.06	42.33±4.17	31.62±2.63	18.71±5.20	30.74±3.47
FedProx + SUBLIME	54.12±0.54	47.73±0.11	31.20±0.29	46.16±0.26	42.31±3.08	41.66±2.25	21.94±1.93	34.30±3.06
FedSGD + DCRN	64.72±1.97	54.48±2.89	42.50±3.29	60.54±2.70	42.47±6.09	35.83±6.06	20.32±5.88	28.39±9.65
FedSGD + DeSE	42.30±6.97	29.46±8.19	18.67±9.97	26.24±7.20	35.92±6.15	26.00±2.54	12.02±4.07	21.16±3.30
FedSGD + SUBLIME	44.95±5.02	30.37±4.10	22.27±5.83	33.45±4.14	35.00±1.98	23.29±4.02	12.61±2.36	24.79±1.42
FedU^2 + DCRN	33.55±3.88	21.57±7.42	11.10±5.72	25.17±3.83	30.26±2.17	16.68±2.82	7.57±2.14	27.09±3.57
FedU^2 + DeSE	38.13±5.47	23.50±8.21	14.04±7.88	16.47±3.42	39.13±4.66	17.78±2.96	10.38±6.37	15.50±4.77
FedU^2 + SUBLIME	42.31±2.82	38.22±1.82	23.21±1.67	33.12±2.05	34.50±2.92	31.76±3.43	13.78±2.98	26.47±2.87
GCFGAE†	19.25±0.09	2.51±0.30	0.28±0.03	13.45±0.41	19.31±1.19	1.77±0.73	0.41±0.17	10.65±0.34
FedSuCo-SKM	**65.42±6.80**	**59.70±7.35**	**45.08±9.05**	**63.91±7.65**	**43.35±5.28**	**42.83±2.98**	**25.96±2.94**	**36.18±7.77**
FedSuCo-vMF	64.61±8.70	58.71±6.80	44.67±10.57	61.69±9.65	40.19±4.32	41.32±2.57	22.04±1.57	35.13±5.35

sample standard deviation. Results are evaluated at prespecified final training budgets. Both FedSuCo clustering variants use the same representation checkpoint for each run. The default supernode grouping parameter is set to 5. Complete dataset statistics, partition construction, baseline implementations and additional experimental results are provided in Appendix G.

5.2 Main Results

Table 2 compares FedSuCo with thirteen federated baselines. Under the reported protocol, FedSuCo achieves the highest mean on all five datasets and all four evaluation metrics. The spherical k-means variant (FedSuCo-SKM) obtains the best result in 16 of the 20 dataset–metric comparisons, while the von Mises–Fisher mixture variant (FedSuCo-vMF) performs better on ACC and Macro-F1 for Cora and CiteSeer. The gains are particularly pronounced on the citation networks, indicating that supernode-level structural collaboration effectively recovers information lost by isolated client subgraphs. On the Amazon datasets, the improvements are smaller but remain consistent across most metrics.

5.3 Scalability

We evaluate whether FedSuCo remains effective as the federation becomes increasingly fragmented by varying the number of clients $C \in \{2, 5, 10, 20\}$. The two competing methods are selected once under the 10-client protocol and are not re-selected or retuned for other client counts. Figure 2 reports ACC as the mean and sample standard deviation over five training seeds. Complete results for NMI, ARI, and Macro-F1 are deferred to Appendix G.3.

FedSuCo maintains a clear advantage as the number of clients increases. At $C = 20$, FedSuCo-vMF exceeds the stronger of the two fixed competitors by 12.21–17.72 ACC percentage points across all five datasets. The response is dataset dependent: Cora reaches its highest accuracy at an intermediate client count rather than declining monotonically with C. This may reflect an implicit regularization effect, as moderate partitioning limits reliance on local neighborhoods. In contrast, CiteSeer degrades moderately as C increases, while Amazon Photo exhibits greater variance under highly fragmented partitions.

5.4 SUPERNODE CAPACITY

We vary the supernode capacity $b \in \{3, 5, 8, 20\}$ with ten clients. As shown in Figure 3, no single capacity dominates across all datasets. Cora remains stable for $b = 3$–8, with spherical k-means favoring $b = 3$ and vMF favoring $b = 5$. CiteSeer performs best around $b = 8$, whereas PubMed favors $b = 5$ under spherical k-means and yields comparable vMF results at $b = 3$ and $b = 5$. Amazon Photo benefits most from $b = 3$, while Amazon Computers generally favors the range $b = 3$–5. Increasing the capacity to $b = 20$ typically reduces ACC, indicating that overly aggressive coarsening can remove useful node-level distinctions. Overall, FedSuCo is stable within a moderate capacity range, and $b = 5$ provides a practical default across datasets.

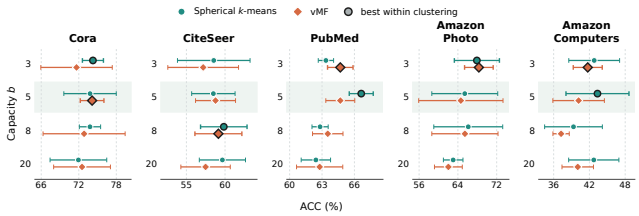

Figure 3: Sensitivity to the supernode capacity under federated spherical k-means and vMF clustering.

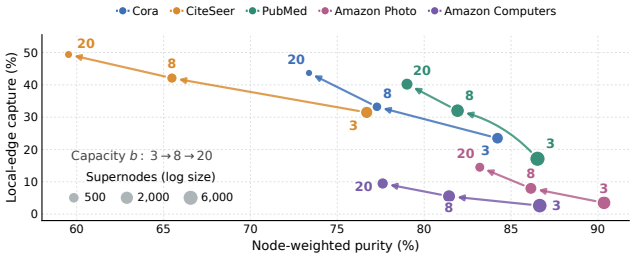

Figure 4: Structural changes induced by the supernode capacity.

Figure 4 explains the structural effect behind this sensitivity. As b increases, each supernode contains more nodes, producing a smaller server-side supergraph. Larger supernodes also absorb a greater fraction of local edges, thereby increasing local-edge capture. However, they are more likely to combine nodes from different semantic classes, which consistently reduces node-weighted purity. Capacity therefore controls a trade-off between semantic resolution, structural coverage, and supergraph size. Fine-grained supernodes preserve more homogeneous node groups but require a larger global supergraph, whereas coarse supernodes provide stronger compression and capture more local connectivity at the cost of mixing dissimilar nodes. The resulting clustering performance is consequently non-monotonic and dataset dependent, supporting the use of a moderate capacity rather than aggressive coarsening.

5.5 ABLATION AND MECHANISM ANALYSIS

Effect of server feedback. Table 3 reveals a dataset-dependent effect of coarse-to-fine feedback. Removing it reduces Cora ACC by approximately 28–35 percentage points across the two client settings, but improves CiteSeer ACC by 5.58 points; Amazon Photo exhibits smaller average gains

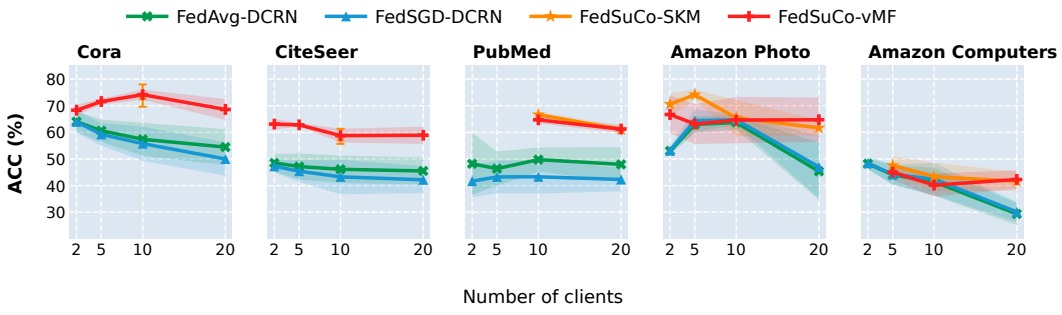

Figure 2: Clustering accuracy with varying numbers of clients.

Table 3: Effects of server feedback and hard-membership refinement.

Setting	Full	No-feedback	Fixed-membership
Cora, 10 clients	74.13 ± 1.89	39.44 ± 4.12	74.76 ± 1.58
Cora, 20 clients	68.59 ± 3.93	40.27 ± 4.85	63.11 ± 12.12
CiteSeer, 10 clients	58.77 ± 2.59	64.35 ± 2.34	58.54 ± 2.49
Amazon Photo, 10 clients	64.61 ± 8.70	61.25 ± 9.13	64.25 ± 12.34

Table 4: Analytical application-level communication with 10 clients.

Dataset	Coarse upload (KiB/outer)	Cross-edge payload (MiB/outer)	Total lower bound (GiB/run)
Cora	65.59	1.75	10.12
CiteSeer	74.78	1.08	23.61
PubMed	380.89	132.19	5.89
Amazon Photo	434.06	113.83	5.79
Amazon Computers	1,027.83	593.34	13.07

from feedback. Thus, transferring server-side structural information can substantially improve node representations, but may also induce negative transfer under the current configuration, motivating adaptive feedback strength.

Effect of adaptive membership refinement. Fixing hard memberships has little effect on mean ACC in the 10-client experiments, but reduces performance and increases variability on Cora-20. A complementary single-seed study with a 5% movement budget increases node-weighted purity and local-edge capture by approximately 3.8 and 19.6 percentage points, respectively. These observations suggest that adaptive reassignment can improve semantic coherence and structural retention, although downstream clustering metrics do not improve uniformly.

5.6 Communication and Privacy Analysis

Communication cost. Table 4 reports analytical communication costs. Coarse uploads remain compact, whereas the current dense one-hot protocol for secure cross-edge construction incurs substantial costs on larger graphs: its per-outer payload increases from 1.75 MiB on Cora to 593.34 MiB on Amazon Computers. This identifies private cross-edge aggregation as a scaling bottleneck of the current prototype and motivates a more efficient sparse implementation. The reported totals are application-level lower bounds, excluding inter-server MPC traffic and transport overhead.

Numerical correctness and privacy. Small-scale two-party tests using MP-SPDZ (Keller, 2020) yield zero error for secure summation and FedSGD parameter updates. Separate additive-sharing checks for Cora vMF and Amazon Photo spherical k-means produce zero label disagreement with their plaintext counterparts under matched initializations, with maximum center errors below 10^{-13}. These results verify numerical consistency in the tested settings. Input confidentiality is conditional on the non-colluding, semi-honest threat model and authorized releases in Section 3.3; disclosed coarse counts may still support inference. Main performance experiments use trusted simulation for cross-edge construction, while two-party execution and clustering equivalence are validated separately. Detailed accounting, validation results, and output-disclosure analysis are provided in the appendix.

6 Conclusion

We presented FedSuCo, a federated graph clustering framework that uses geometry-guided supernodes for global structural collaboration. It combines local hyperbolic representation learning, weighted supergraph modeling, and coarse-to-fine feedback while retaining node representations and memberships at clients. Two-server aggregation of sufficient statistics supports global spherical clustering under an explicit disclosure boundary. Experiments on five graph datasets show improved clustering performance over federated baselines, while ablations reveal dataset-dependent effects of feedback and membership refinement. Separate protocol validations confirm numerical consistency in the tested settings.

Ethics Statement

This study uses publicly available benchmark datasets and does not involve human subjects or personally identifiable information. FedSuCo keeps individual-node representations and assignments local and exposes only aggregated statistics under the stated threat model; we identify no additional ethical concerns beyond the privacy limitations discussed in the paper.

Reproducibility Statement

We provide all essential details, including datasets, pseudocode, hyperparameters, and environment settings, to facilitate the reproducibility of our experiments. We will also open-source the complete codebase.

AI Use Statement

We used ChatGPT and Codex to assist with methodological and experimental discussions, code development and debugging, interpretation of experimental results, manuscript drafting and translation, and figure preparation. The authors reviewed the AI-assisted content, checked the implementation against the intended method, and verified the reported results. The authors take full responsibility for the final manuscript, code, and conclusions.

References

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Gregor Bachmann, Gary Becigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 486–496. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bachmann20a.html.

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(46): 1345–1382, 2005. URL http://jmlr.org/papers/v6/banerjee05a.html.

Donald Beaver. Efficient multiparty protocols using circuit randomization. In Joan Feigenbaum (ed.), *Advances in Cryptology — CRYPTO '91*, pp. 420–432, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. ISBN 978-3-540-46766-3.

Marián Boguñá, Ivan Bonamassa, Manlio De Domenico, Shlomo Havlin, Dmitri Krioukov, and M. Ángeles Serrano. Network geometry. *Nature Reviews Physics*, 3(2):114–135, 2021. ISSN 2522-5820. doi: 10.1038/s42254-020-00264-4. URL https://doi.org/10.1038/s42254-020-00264-4.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0415740eaa4d9decbc8da001d3fd805f-Paper.pdf.

Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric, 2019. URL https://arxiv.org/abs/1903.02428.

Lele Fu, Tianchi Liao, Sheng Huang, Bowen Deng, Chuanfu Zhang, Shirui Pan, and Chuan Chen. Unsupervised federated graph learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*,

volume 38, Main Conference, pp. 97431–97457. Curran Associates, Inc., 2025. doi: 10.52202/085713-3257. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8c4c80d3eee8a56da1302bc6dcd2d953-Paper-Conference.pdf.

Xingbo Fu, Binchi Zhang, Yushun Dong, Chen Chen, and Jundong Li. Federated graph machine learning: A survey of concepts, techniques, and applications. *SIGKDD Explor. Newsl.*, 24(2): 32–47, December 2022. ISSN 1931-0145. doi: 10.1145/3575637.3575644. URL https://doi.org/10.1145/3575637.3575644.

Guillermo García-Pérez, Marián Boguñá, and M. Ángeles Serrano. Multiscale unfolding of real networks by geometric renormalization. *Nature Physics*, 14(6):583–589, 2018. ISSN 1745-2481. doi: 10.1038/s41567-018-0072-5. URL https://doi.org/10.1038/s41567-018-0072-5.

Kun Guo, Yutong Fang, Qingqing Huang, Yuting Liang, Ziyao Zhang, Wenyu He, Liu Yang, Kai Chen, Ximeng Liu, and Wenzhong Guo. Globally consistent federated graph autoencoder for non-iid graphs. In Edith Elkind (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 3768–3776. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/419. URL https://doi.org/10.24963/ijcai.2023/419. Main Track.

Kurt Hornik, Ingo Feinerer, Martin Kober, and Christian Buchta. Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012. doi: 10.18637/jss.v050.i10. URL https://www.jstatsoft.org/index.php/jss/article/view/v050i10.

Robert Jankowski, Antoine Allard, Marián Boguñá, and M. Ángeles Serrano. The d-mercator method for the multidimensional hyperbolic embedding of real networks. *Nature Communications*, 14(1):7585, 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-43337-5. URL https://doi.org/10.1038/s41467-023-43337-5.

George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998. doi: 10.1137/S1064827595287997. URL https://doi.org/10.1137/S1064827595287997.

Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, pp. 1575–1590, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899. doi: 10.1145/3372297.3417872. URL https://doi.org/10.1145/3372297.3417872.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze (eds.), *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Fengyuan Yu, Huabin Zhu, Binhui Yao, Tao Wang, Xiaolin Zheng, and Yanchao Tan. Rethinking the representation in federated unsupervised learning with non-iid data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22841–22850, June 2024.

Jingxin Liu, Renda Han, Wenxuan Tu, Haotian Wang, Junlong Wu, and Jieren Cheng. Federated node-level clustering network with cross-subgraph link mending. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38540–38556. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/liu25v.html.

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, 2025b. doi: 10.1109/TNNLS.2024.3360429.

Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 1392–1403, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512186. URL https://doi.org/10.1145/3485447.3512186.

Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7603–7611, 2022b.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *Workshop on Mining and Learning with Graphs*, 2012.

Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. ISSN 1860-0980. doi: 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3), 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2019. URL https://arxiv.org/abs/1811.05868.

Li Sun, Zhenhao Huang, Hao Peng, Yujie Wang, Chunyang Liu, and Philip S. Yu. LSEnet: Lorentz structural entropy neural network for deep graph clustering. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47078–47104. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sun24g.html.

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: a deep attentional embedding approach. In *IJCAI*, 2019.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10834–10842, 2023.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/yanga16.html.

Philip S. Yu and Li Sun. Riemannian geometry speaks louder than words: From graph foundation model to next-generation graph intelligence, 2026. URL https://arxiv.org/abs/2603.21601.

Jingyun Zhang, Hao Peng, Li Sun, Guanlin Wu, Chunyang Liu, and Zhengtao Yu. Unsupervised graph clustering with deep structural entropy. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 3752–3763, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737173. URL https://doi.org/10.1145/3711896.3737173.

Ke ZHANG, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6671–6682. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/34adeb8e3242824038aa65460a47c29e-Paper.pdf.

A NOTATION

Bold lowercase and uppercase letters denote vectors and matrices, with vector collections stacked by rows. Supernode indices s, t are local when a client index is present and global otherwise. Superscripts (t), (τ), and (ℓ) denote outer rounds, optimizer steps, and clustering iterations, respectively, and are omitted in single-round formulas. Node-pair and layer indices are local to their equations.

Table 5: Notation used in FedSuCo.

Notation	Meaning	Notation	Meaning
Graph, clients, and dimensions			
$\mathcal{G}, \mathcal{V}, \mathcal{E}$	Graph, node set, and edge set	$C;\ c, c'$	Client count; client indices
$N,\ n_c$	Total and client node counts	$\mathcal{V}_c, \mathcal{D}_c$	Client node set and private data
d_x	Node feature dimension	d	Hyperbolic and spherical coordinate dimension
$\mathbf{X}, \mathbf{X}_c$	Global and local feature matrices	$\mathbf{A}_c,\ a_{c,ij}$	Local adjacency and its entries
$\mathcal{E}_c^{\mathrm{in}}, \mathcal{E}_c^{\times}$	Internal and incident external edges	$\mathcal{E}_{cc'}$	Edges between clients c and c'
K	Number of global clusters	$\hat{y}_{c,i},\ y_{c,i}^{\mathrm{gt}}$	Predicted cluster and evaluation label
$t,\ T$	Outer round and total outer rounds	$\tau,\ \ell$	Optimizer step and clustering iteration
Local geometry and supernode construction			
$\mathcal{H}_\chi,\ \chi$	Hyperbolic ball and curvature	$\mathbb{S}^{d-1}$	Unit sphere in $\mathbb{R}^d$
$\mathbf{Z}_c, \mathbf{z}_{c,i}$	Hyperbolic node embeddings	$\mathbf{E}_c, \mathbf{e}_{c,i}$	Learned unit node directions
$\mathbf{Y}_c, \mathbf{y}_{c,i}$	Geometric coordinates in dimension d	$\kappa_{c,i}, \beta_c$	Hidden degree and local geometric parameter
b	Base group size, not an upper bound	$m_c,\ M$	Local and total supernode counts
$\mathcal{V}_{c,s},\ n_{c,s}$	Supernode members and size	$\mathbf{P}_c, \mathbf{P}_c^{(t)}$	Membership matrix, $m_c \times n_c$
$\overline{\mathbf{E}}, \overline{\mathbf{e}}_{c,s}$	Initial supernode directions	$\overline{\kappa}_{c,s}$	Aggregated hidden degree
$\mathbf{n}$	Global supernode size vector	$\boldsymbol{\kappa}_c$	Local hidden-degree vector
r_c	Number of remainder nodes	$\mathbf{m}_{c,s}$	Weighted geometric group center
$d_{\mathrm{geo}},\ J_{c,is}$	Geodesic distance and assignment cost	$\mathbf{Q}_c^\star$	Orthogonal coordinate alignment
$u_{c,s},\ w_{c,st}$	Internal and between-group edge counts	$\mathbf{W}_{cc'}$	Cross-client coarse edge counts
Local learning and node feedback			
$f_\Theta,\ \Theta$	Shared encoder and shared model parameters	$D_\phi,\ \phi$	Feature decoder and its parameters
$\overline{\mathbf{A}}_c, \widetilde{\mathbf{D}}_c$	Normalized adjacency; degree diagonal	$\mathbf{W}_p, \mathbf{B}_c$	Feature weights and projection
$\mathcal{K}_{\chi,\ell}, \mathbf{H}_c^{(\ell)}$	Local layer and hidden state (layer ℓ)	α	Feature branch mixing coefficient
$\Pi_o, \mathbf{U}_c$	Tangent projection and coordinates	$\mathrm{Exp}_o^{\mathcal{H}_\chi}$	Exponential map at the origin
$\rho,\ d_{\mathcal{H}_\chi}$	Shell radius; hyperbolic distance	$b_{\mathrm{loc}}, \tau_{\mathrm{loc}}$	Local link bias and temperature
$\widehat{a}_{c,ij}$	Reconstructed link probability	Ω_c, ω_c	Node-pair set; positive-class weight

Continued on the next page

Table 5 (continued)

Notation	Meaning	Notation	Meaning
$\mathcal{L}_{\text{feat}}^c$, $\mathcal{L}_{\text{link}}^c$	Feature and link losses	$\mathcal{L}_c^{\text{loc}}$, $\mathcal{L}_c^{\text{refine}}$	Local and refinement objectives
$\mathbf{F}_c$	Returned directions, $m_c \times d$	$\widetilde{\mathbf{F}}_c$	Lifted node targets, $n_c \times d$
$\mathcal{L}_{\text{fb}}^c$	Directional feedback loss	$\mathbf{E}'_c$, $\mathbf{e}'_{c,i}$	Refined node directions
$\mathbf{g}_c$, $\overline{\mathbf{g}}$	Client and global gradients	$\widetilde{\mathbf{g}}$, B_g	Clipped gradient; norm threshold
λ_{feat}, λ_{link}, λ_{fb}	Local objective weights	$\text{norm}(\mathbf{v})$	Unit normalization for nonzero $\mathbf{v}$
Global weighted supergraph			
$\mathcal{G}^S$, $\mathcal{V}^S$	Global supergraph and supernode set	$\mathbf{W}$, W_{st}	Coarse edge-count matrix
$\mathbf{X}^S$, $\mathbf{x}_s^S$	Server input features, $M \times (d+3)$	$\widehat{\mathbf{W}}$, $\mathbf{D}_S$	Propagation matrix; degree diagonal
$\mathbf{H}_S^{(\ell)}$	Server hidden state at layer ℓ	$\mathbf{U}_0$, $\mathbf{U}_1$, $\mathbf{U}_2$	Server encoder weight matrices
$\mathbf{b}_0$, $\mathbf{b}_2$	Server encoder bias vectors	$\mathbf{R}$, $\mathbf{r}_s$	Structural proposals
$\mathbf{F}$, $\mathbf{f}_s$	Refined supernode directions, $M \times d$	γ, a_γ	Residual gate and its logit
T_{st}, p_{st}	Pair count; connection probability	ℓ_{st}	Connection logit, $\text{logit}(p_{st})$
ξ, b_{str}	Similarity scale; structural bias	d_s^{obs}, d_s^{pred}	Observed and predicted row sums
$\mathcal{N}$, q_{st}	Nonedges; clipped similarity	h, δ	Centering operator; margin
$\mathcal{L}_{\text{str}}$, $\mathcal{L}_{\text{align}}$	Reconstruction and alignment losses	$\mathcal{L}_{\text{push}}$, $\mathcal{L}_{\text{shape}}$	Repulsion and profile terms
$\mathcal{L}_S$	Complete server objective	λ_{str}, λ_{align}, λ_{push}, λ_{shape}	Server objective weights
Federated spherical clustering and evaluation			
$\boldsymbol{\mu}_j$	Unit direction of cluster j	π_j, ν_j	vMF mixture weight and concentration
Φ	Collection of mixture parameters	$C_d(\nu)$	vMF normalizer in dimension d
η_{cij}, r_{cij}	vMF log score; responsibility	N_{cj}, N_j	Local and global effective counts
$\mathbf{S}_{cj}$, $\mathbf{S}_j$	Local and global directional sums	$\overline{R}_j$	Mean resultant length, $\|\mathbf{S}_j\|_2/N_j$
D_s	Distance to nearest seed	$\nu_{\min}$, $\nu_{\max}$	Concentration clipping bounds
$\mathbf{C}_c^{\text{eval}}$, $\mathbf{C}^{\text{eval}}$	Local and global contingency tables	ε, σ	Stability constant; sigmoid function
Privacy and secure computation			
S_0, S_1	Coordinator and assisting server	$q \in \{0,1\}$	Server and additive-share index
$\mathcal{A}$	Set of passively corrupted servers	$\mathcal{M}_{\text{pub}}$	Public protocol metadata
$\mathcal{R}$	Collection of authorized releases	$\mathcal{R}_{\text{coarse}}$, $\mathcal{R}_{\text{train}}$	Coarsening and training releases
$\mathcal{R}_{\text{cluster}}$, $\mathcal{R}_{\text{eval}}$	Clustering and evaluation releases	SecAgg	Secure aggregation opening only the sum
$\mathbf{v}_c$, $\mathbf{v}_c^{\text{enc}}$	Client vector and ring encoding	$\boldsymbol{\zeta}_c$	Fresh uniform random mask
$[\mathbf{v}_c]_q$, $\mathbf{v}_q^{\Sigma}$	Input share; server share sum	$\mathbf{b}_e^{(c)}$	One-hot membership of edge e's endpoint
$\mathbb{Z}_{2^L}$, L	Arithmetic ring and its bit length	s_{fp}	Fixed-point encoding scale

B Local Encoder and Federated Optimization

B.1 Riemannian Encoder

Let $\mathcal{H}_\chi$ denote the hyperbolic ball in d-dimensional coordinates with curvature $\chi < 0$. Client c constructs $\overline{\mathbf{A}}_c = \widetilde{\mathbf{D}}_c^{-1/2}(\mathbf{A}_c + \mathbf{I})\widetilde{\mathbf{D}}_c^{-1/2}$, where $\widetilde{\mathbf{D}}_c$ is the degree matrix after adding self-loops. Complete node features are projected into a lower-dimensional space and processed by two curvature-aware graph convolutional layers:

$$\mathbf{B}_c = \mathbf{X}_c\mathbf{W}_p, \quad \mathbf{H}_c^{(1)} = \mathcal{K}_{\chi,1}(\mathbf{B}_c, \overline{\mathbf{A}}_c), \quad \mathbf{H}_c^{(2)} = \mathcal{K}_{\chi,2}(\mathbf{H}_c^{(1)}, \overline{\mathbf{A}}_c). \tag{7}$$

The residual branches $\mathbf{B}_c$ and $\mathbf{H}_c^{(2)}$ both lie in $\mathbb{R}^{n_c \times d}$, with $\mathbf{W}_p \in \mathbb{R}^{d_x \times d}$. Here, $\mathcal{K}_{\chi,\ell}$ denotes a κ-GCN layer (Bachmann et al., 2020) with curvature-aware linear transformation and neighborhood

aggregation. The first layer uses LeakyReLU and dropout, while the second applies no output activation. The branches are fused, projected into tangent coordinates at the origin, and mapped to the hyperbolic space:

$$\mathbf{U}_c = \Pi_o\Big((1-\alpha)\mathbf{H}_c^{(2)} + \alpha\mathbf{B}_c\Big), \qquad \mathbf{Z}_c = \mathrm{Exp}_o^{\mathcal{H}_\chi}(\mathbf{U}_c). \tag{8}$$

The coefficient α controls the feature-branch contribution, and Π_o denotes the encoder's tangent-coordinate projection, producing $\mathbf{U}_c \in \mathbb{R}^{n_c \times d}$.

B.2 Local Reconstruction Objectives

The feature decoder D_ϕ consists of two affine layers with an intermediate ReLU activation. Its objective is $\mathcal{L}_{\text{feat}}^c = \|D_\phi(\mathbf{Z}_c) - \mathbf{X}_c\|_F^2/(n_c d_x)$. For link reconstruction, nodes are placed on a fixed shell of radius ρ inside the hyperbolic ball and decoded as

$$\widehat{a}_{c,ij} = \sigma\left(\frac{b_{\text{loc}} - d^2_{\mathcal{H}_\chi}(\rho\mathbf{e}_{c,i}, \rho\mathbf{e}_{c,j})}{\tau_{\text{loc}}}\right), \tag{9}$$

Here, $a_{c,ij} = [\mathbf{A}_c]_{i,j}$, $\widehat{a}_{c,ij}$ is its reconstructed probability, and b_{loc} and $\tau_{\text{loc}} > 0$ are learnable. Fixing the radius makes pairwise distances depend on directional relationships. For the local node-pair set Ω_c, the link objective is

$$\mathcal{L}_{\text{link}}^c = -\frac{1}{|\Omega_c|}\sum_{(i,j)\in\Omega_c}\left[\omega_c a_{c,ij}\log\widehat{a}_{c,ij} + (1-a_{c,ij})\log(1-\widehat{a}_{c,ij})\right]. \tag{10}$$

Dense reconstruction uses all local adjacency entries and a clipped positive-class weight ω_c determined by positive and negative counts. Sampled reconstruction uses selected positive and negative pairs with $\omega_c = 1$.

B.3 Federated Gradient Optimization

The shared parameters Θ include the local encoder and decoder parameters. At optimizer step τ, each client computes $\mathbf{g}_c^{(\tau)}$ from $\mathcal{L}_c^{\text{loc}}$ or $\mathcal{L}_c^{\text{refine}}$, depending on the training stage. The server obtains

$$\overline{\mathbf{g}}^{(\tau)} = \frac{1}{N}\,\mathrm{SecAgg}_c\left(n_c\mathbf{g}_c^{(\tau)}\right). \tag{11}$$

The aggregate gradient is clipped by its global norm before being passed to Adam (Kingma & Ba, 2017). With threshold B_g, the optimizer uses $\widetilde{\mathbf{g}}^{(\tau)} = \overline{\mathbf{g}}^{(\tau)}\min\{1, B_g/\|\overline{\mathbf{g}}^{(\tau)}\|_2\}$, with zero gradients left unchanged. During feedback, returned supernode directions are fixed targets and receive no gradients. The secure aggregation primitive is specified in Appendix E.

C Geometric Coarsening and Supernode Statistics

C.1 Complete Groups and Remainder Assignment

Assume $n_c \geq b$, and let $m_c = \lfloor n_c/b \rfloor$ and $r_c = n_c - m_c b$. At each step, the unassigned node with the largest hidden degree is selected as a seed and grouped with its nearest unassigned neighbors under $d_{\text{geo}}(\mathbf{y}_{c,i}, \mathbf{y}_{c,j}) = \arccos(\mathbf{y}_{c,i}^\top \mathbf{y}_{c,j})$. This produces m_c complete groups of size b.

Remaining nodes are incorporated into existing groups through capacity-balanced linear assignment. Each matching round computes weighted centers

$$\mathbf{m}_{c,s} = \mathrm{norm}\left(\sum_{i\in\mathcal{V}_{c,s}}\kappa_{c,i}\mathbf{y}_{c,i}\right) \tag{12}$$

and minimizes the assignment cost $J_{c,is} = 1 - \mathbf{y}_{c,i}^\top\mathbf{m}_{c,s}$, adding at most one node to each group. If the remainder exceeds the group count, centers are updated and matching continues. The disjoint groups cover every node and satisfy

$$\sum_s |\mathcal{V}_{c,s}| = n_c, \qquad \min_s |\mathcal{V}_{c,s}| \geq b, \qquad \max_s |\mathcal{V}_{c,s}| - \min_s |\mathcal{V}_{c,s}| \leq 1. \tag{13}$$

Thus, no undersized tail group is produced. The capacity parameter b specifies the base group size; absorbing remainder nodes can produce groups larger than b.

C.2 Coordinate Alignment and Rotation Invariance

Let $\mathbf{Y}_c \in \mathbb{R}^{n_c \times d}$ stack the geometric coordinates, using the same coordinate dimension as $\mathbf{E}_c$, and let $\boldsymbol{\kappa}_c = (\kappa_{c,i})_{i=1}^{n_c}$. These coordinates can be aligned with learned directions by solving the orthogonal Procrustes problem (Schönemann, 1966):

$$\mathbf{Q}_c^\star = \arg \min_{\mathbf{Q} \in \mathbb{R}^{d \times d},\, \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}} \|\mathbf{Y}_c \mathbf{Q} - \mathbf{E}_c\|_F^2.$$

Orthogonal transformations preserve all pairwise inner products and hence geodesic distances. The remainder assignment costs also remain unchanged when weighted group centers undergo the same rotation. Alignment therefore does not alter the distance-based grouping criterion.

C.3 Membership and Edge-Count Computation

During outer training, clients optimize local assignment scores using a geometric membership prior, directional coherence, local-edge retention, and entropy regularization. A capacity-constrained projection yields the binary membership matrix $\mathbf{P}_c^{(t)}$; budgeted refinement restricts changes to beneficial pairwise swaps. These updates preserve the initial group sizes:

$$\sum_s [\mathbf{P}_c^{(t)}]_{s,i} = 1, \qquad \sum_i [\mathbf{P}_c^{(t)}]_{s,i} = n_{c,s}. \tag{14}$$

Thus, $\mathbf{P}_c^{(t)}$ can change while m_c and $n_{c,s}$ remain fixed. Omitting the round index, the matrix satisfies $[\mathbf{P}_c]_{s,i} = 1$ if and only if $i \in \mathcal{V}_{c,s}$, with exactly one nonzero entry per column. For an undirected original subgraph without self-loops, the edge counts are

$$u_{c,s} = \frac{1}{2} \sum_{i,j \in \mathcal{V}_{c,s}} a_{c,ij}, \qquad w_{c,st} = \sum_{i \in \mathcal{V}_{c,s}} \sum_{j \in \mathcal{V}_{c,t}} a_{c,ij}, \quad s \neq t. \tag{15}$$

These define $[\mathbf{W}_{cc}]_{s,s} = u_{c,s}$ and $[\mathbf{W}_{cc}]_{s,t} = w_{c,st}$ for $s \neq t$. Together with Eq. equation 2, supernode sizes, and aggregated hidden degrees, these counts form the payload authorized in Section 3.3.

D Supergraph Encoder and Objective

D.1 Weighted Message Passing

Let $\mathbf{W} \in \mathbb{R}^{M \times M}$. Off-diagonal entries of $\mathbf{W}$ count edges between supernodes, and diagonal entries count within-supernode edges. With $d_s^{\mathrm{obs}} = \sum_t W_{st}$, the input is $\mathbf{x}_s^S = [\overline{\mathbf{e}}_s; \log n_s; \log \overline{\kappa}_s; \log(1 + d_s^{\mathrm{obs}})]$. Stacking these features gives $\mathbf{X}^S \in \mathbb{R}^{M \times (d+3)}$. After adding propagation self-loops, use $\widehat{\mathbf{W}} = \mathbf{D}_S^{-1/2}(\mathbf{W} + \mathbf{I})\mathbf{D}_S^{-1/2}$, where $\mathbf{D}_S$ contains row sums of $\mathbf{W} + \mathbf{I}$. The structural proposal is computed by

$$\begin{aligned} \mathbf{H}_S^{(1)} &= \mathrm{GELU}\Big(\widehat{\mathbf{W}}(\mathbf{X}^S \mathbf{U}_0 + \mathbf{1}\mathbf{b}_0^\top)\Big), \\ \mathbf{H}_S^{(2)} &= \mathrm{GELU}\Big(\widehat{\mathbf{W}}\mathbf{H}_S^{(1)}\mathbf{U}_1\Big), \qquad \mathbf{R} = \mathbf{H}_S^{(2)}\mathbf{U}_2 + \mathbf{1}\mathbf{b}_2^\top. \end{aligned} \tag{16}$$

Here, $\mathbf{R} \in \mathbb{R}^{M \times d}$ stacks the structural proposals $\mathbf{r}_s$. The initial directions and supernode sizes are stacked as $\overline{\mathbf{E}} = [\overline{\mathbf{e}}_s^\top]_{s=1}^M$ and $\mathbf{n} = (n_1, \ldots, n_M)^\top$. The residual gate is $\gamma = \sigma(a_\gamma)$, with learnable scalar a_γ.

D.2 Weighted Edge Reconstruction

The trial counts are $T_{st} = n_s n_t$ for distinct supernodes and $T_{ss} = n_s(n_s - 1)/2$ within a supernode. Let $\ell_{st} = \mathrm{logit}(p_{st})$ be defined by Eq. equation 4. Omitting parameter-independent binomial coefficients gives

$$\mathcal{L}_{\mathrm{str}} = \frac{\sum_{s \leq t} [T_{st}\, \mathrm{softplus}(\ell_{st}) - W_{st}\ell_{st}]}{\sum_{s \leq t} T_{st}}. \tag{17}$$

Only upper-triangular entries are included to avoid double counting.

D.3 Auxiliary Regularization and Complete Objective

Directional alignment. Node-count weighting constrains deviations from the initial directions:

$$\mathcal{L}_{\text{align}} = \frac{\sum_s n_s(1 - \mathbf{f}_s^\top \bar{\mathbf{e}}_s)}{\sum_s n_s}. \tag{18}$$

Nonedge repulsion. Let $\mathcal{N} = \{(s,t) : s < t,\ W_{st} = 0\}$ and $q_{st} = \text{clip}(\mathbf{f}_s^\top \mathbf{f}_t, -1+\varepsilon, 1-\varepsilon)$. The repulsion term is

$$\mathcal{L}_{\text{push}} = \frac{1}{|\mathcal{N}|} \sum_{(s,t)\in\mathcal{N}} \left[-\log \frac{1 - q_{st}}{2} \right] [q_{st} - \delta]_+, \tag{19}$$

where $[x]_+ = \max(x, 0)$ and δ is a margin. The term is zero if $\mathcal{N}$ is empty.

Structural profiles. Define $d_s^{\text{obs}} = \sum_t W_{st}$, $d_s^{\text{pred}} = \sum_t T_{st} p_{st}$, and the centering operator $h(\mathbf{v}) = \mathbf{v} - M^{-1}(\sum_s v_s)\mathbf{1}$. Up to terms independent of server parameters, the shape term is

$$\mathcal{L}_{\text{shape}} = \frac{1}{M} \left\| h(\log(1 + \mathbf{d}^{\text{pred}})) - h(\log(1 + \mathbf{d}^{\text{obs}})) \right\|_2^2. \tag{20}$$

Logarithms are applied elementwise. The complete objective is

$$\mathcal{L}_S = \lambda_{\text{str}}\mathcal{L}_{\text{str}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{push}}\mathcal{L}_{\text{push}} + \lambda_{\text{shape}}\mathcal{L}_{\text{shape}}. \tag{21}$$

The server uses Adam to optimize its encoder, gate, and connectivity decoder.

E Secure Computation Protocols

E.1 Additive Sharing and Secure Aggregation

All protocols use the system assumptions in Section 3.2 and the authorized outputs in Section 3.3. We use two-server additive secret sharing over $\mathbb{Z}_{2^L}$ (Cramer et al., 2015). For a client vector $\mathbf{v}_c$, encode $\mathbf{v}_c^{\text{enc}} = \text{round}(s_{\text{fp}}\mathbf{v}_c) \bmod 2^L$, where s_{fp} is the fixed-point scale and arithmetic takes place in $\mathbb{Z}_{2^L}$. Using a fresh, uniformly distributed cryptographic mask $\boldsymbol{\zeta}_c$, the client creates

$$[\mathbf{v}_c]_0 = \boldsymbol{\zeta}_c, \qquad [\mathbf{v}_c]_1 = \mathbf{v}_c^{\text{enc}} - \boldsymbol{\zeta}_c \pmod{2^L}. \tag{22}$$

For $q \in \{0, 1\}$, server S_q receives only share q and computes $\mathbf{v}_q^\Sigma = \sum_c [\mathbf{v}_c]_q$. After contributions from the declared participant set have been collected, the protocol reconstructs $\mathbf{v}_0^\Sigma + \mathbf{v}_1^\Sigma$ and decodes the aggregate. Masks are refreshed for every release. Integer edge counts and contingency tables use direct integer encoding.

For training, the shared vector contains node-count-weighted gradients and any prescribed aggregate diagnostics. For clustering, it contains component counts and directional sums. Nonlinear model and mixture updates are evaluated on the authorized aggregates. This primitive does not open individual client contributions.

E.2 Secure Cross-Client Superedge Counting

Using the cross-edge correspondence assumed in Section 3.2, each endpoint owner encodes edge e's local supernode membership as a one-hot vector, $\mathbf{b}_e^{(c)} \in \{0,1\}^{m_c}$ or $\mathbf{b}_e^{(c')} \in \{0,1\}^{m_{c'}}$. The owners send additive shares to the two servers. Secure multiplication and accumulation compute

$$[\mathbf{W}_{cc'}]_{s,t} = \sum_{e\in\mathcal{E}_{cc'}} b_{e,s}^{(c)} b_{e,t}^{(c')}. \tag{23}$$

Only the accumulated count matrix is opened; endpoint vectors and per-edge products remain secret-shared. For native two-party execution, we use the `semi2k` backend of MP-SPDZ (Keller, 2020). Secure multiplication uses additively shared Beaver triples (Beaver, 1992), with arithmetic over $\mathbb{Z}_{2^{64}}$.

E.3 Correctness, Precision, and Security Analysis

Correctness. Adding the two shares recovers the encoded input modulo 2^L; summing party-local shares therefore recovers the sum of all encoded client contributions. For cross-edge counting, each edge contributes one to exactly one supernode pair. Hence, the secure output equals the count in Eq. equation 3 and satisfies $\sum_{s,t}[\mathbf{W}_{cc'}]_{s,t} = |\mathcal{E}_{cc'}|$.

Numerical precision. The ring size and scale are chosen so that the decoded aggregates remain within the valid signed range. With nearest-integer rounding and no overflow, a sum of C client contributions has at most $C/(2s_{\mathrm{fp}})$ encoding error per coordinate. Integer counts are exact provided their range fits the ring. This is a per-aggregation bound, not an end-to-end error bound for iterative optimization.

Security argument. Either share of an encoded client contribution is uniformly distributed over the ring when viewed individually. Aggregate reconstruction reveals the prescribed sum. The cross-edge protocol additionally relies on the semi-honest security of its multiplication primitive. Subsequent server-side learning uses only authorized releases. These arguments are conditional on the assumptions and disclosure boundary in Section 3; they concern protocol transcripts and do not strengthen the privacy of the authorized outputs themselves.

F Federated Clustering and Evaluation

F.1 Supernode-Weighted Initialization

We initialize cluster centers with a size-weighted adaptation of k-means++ seeding (Arthur & Vassilvitskii, 2007) applied to the released supernode directions. The first center is sampled proportionally to n_s; subsequent centers are sampled proportionally to $n_s D_s^2$, where D_s is the minimum Euclidean distance from $\mathbf{f}_s$ to an existing center. On unit directions, squared Euclidean distance is proportional to cosine distance. This is equivalent to center seeding on supernode directions replicated by their counts. Mixture weights are then initialized uniformly and concentrations receive a common positive initial value. Subsequent responsibilities are evaluated on original node directions.

F.2 Local Responsibilities and Global Updates

At clustering iteration ℓ, let $\Phi^{(\ell)} = \{\pi_j^{(\ell)}, \boldsymbol{\mu}_j^{(\ell)}, \nu_j^{(\ell)}\}_{j=1}^K$. The vMF concentration ν_j is distinct from hidden degree κ and curvature χ. Each client computes

$$\eta_{cij}^{(\ell)} = \log \pi_j^{(\ell)} + \log C_d(\nu_j^{(\ell)}) + \nu_j^{(\ell)} {\boldsymbol{\mu}_j^{(\ell)}}^\top \mathbf{e}'_{c,i}, \qquad r_{cij}^{(\ell)} = \frac{\exp(\eta_{cij}^{(\ell)})}{\sum_{h=1}^K \exp(\eta_{cih}^{(\ell)})}, \tag{24}$$

where C_d is the vMF normalizing constant. The local statistics are $N_{cj}^{(\ell)} = \sum_i r_{cij}^{(\ell)}$ and $\mathbf{S}_{cj}^{(\ell)} = \sum_i r_{cij}^{(\ell)} \mathbf{e}'_{c,i}$. Each client's vector contains $K(d+1)$ scalars, with one share sent to each server. Equation equation 6 yields the global statistics.

With $\overline{R}_j^{(\ell)} = \|\mathbf{S}_j^{(\ell)}\|_2 / N_j^{(\ell)}$, the updates are

$$\pi_j^{(\ell+1)} = \frac{N_j^{(\ell)}}{N}, \qquad \boldsymbol{\mu}_j^{(\ell+1)} = \frac{\mathbf{S}_j^{(\ell)}}{\|\mathbf{S}_j^{(\ell)}\|_2}. \tag{25}$$

Concentrations use the stabilized approximation

$$\nu_j^{(\ell+1)} = \mathrm{clip}_{[\nu_{\min}, \nu_{\max}]} \left(\frac{d\overline{R}_j^{(\ell)} - (\overline{R}_j^{(\ell)})^3}{\max\{1 - (\overline{R}_j^{(\ell)})^2, \varepsilon\}} \right), \tag{26}$$

where $\varepsilon > 0$ protects the denominator. Iteration stops when center change falls below a tolerance or reaches the iteration limit. Clients recompute responsibilities using the final parameters before assigning labels. Multiple initializations can be compared using the securely aggregated directional inertia $\sum_{c,i}(1 - {\mathbf{e}'_{c,i}}^\top \boldsymbol{\mu}_{\hat{y}_{c,i}})$, an authorized selection score in Section 3.3.

Table 6: Dataset statistics. Edges denote unique undirected edges without self-loops.

Dataset	Nodes	Edges	Features	Classes
Cora	2,485	5,069	1,433	7
CiteSeer	3,327	4,552	3,703	6
PubMed	19,717	44,324	500	3
Amazon Photo	7,650	119,081	745	8
Amazon Computers	13,752	245,861	767	10

F.3 Secure Evaluation

Each client constructs a local contingency table and uses integer secure aggregation:

$$[\mathbf{C}_c^{\text{eval}}]_{j,h} = \sum_i \mathbf{1}\{\hat{y}_{c,i} = j,\ y_{c,i}^{\text{gt}} = h\}, \qquad \mathbf{C}^{\text{eval}} = \text{SecAgg}_c(\mathbf{C}_c^{\text{eval}}). \tag{27}$$

Here, j indexes predicted clusters and h indexes ground-truth classes. ACC, NMI, ARI, and Macro-F1 are computed from the global table. Metrics requiring label alignment perform matching on this aggregate. The role of evaluation labels and the permitted release are defined in Sections 3.1 and 3.3.

F.4 Numerical Equivalence

At fixed mixture parameters, responsibilities are computed independently for each node. Summing local sufficient statistics therefore exactly recovers those of a centralized computation on the same directions. With identical initialization, parameter updates, clipping rules, stopping criteria, and tie handling, induction over iterations gives the same parameter sequence and final assignments in exact arithmetic. Equivalence refers to the specified updates, including the approximate concentration update, rather than an alternative vMF solver. Fixed-point and floating-point implementations may differ numerically; comparisons therefore examine aggregate statistics, parameters, and final assignments under matched initialization and precision settings. The per-aggregation encoding bound is given in Appendix E.3.

G Baseline Experimental Protocols and Additional Results

G.1 Data and Evaluation Protocol

All methods use the same fixed METIS owner assignments for each dataset and client count, with each original node owned by exactly one client. Citation networks use the Planetoid versions, retaining only the largest connected component of Cora; Amazon datasets are loaded through PyTorch Geometric. Table 6 summarizes the data. Node labels are used only for evaluation. We report the mean and sample standard deviation over five seeds at prespecified final training budgets, evaluating each original node exactly once at its owner client.

G.2 Baseline Construction

We evaluate thirteen baselines: the twelve combinations of FedAvg, FedProx, FedSGD, and FedU^2 with DCRN, DeSE, and SUBLIME, plus GCFGAE. For the twelve combinations, graph construction and message passing are restricted to client-local owner graphs, while the federated mechanism coordinates shared model parameters.

DCRN applies dual correlation reduction at the sample and feature levels (Liu et al., 2022b); our adaptation uses random initialization without pretraining. DeSE combines structural-entropy and link-reconstruction objectives to learn local graph structure and soft cluster assignments (Zhang et al., 2025). SUBLIME learns local graph structure through contrastive learning between learner and anchor views, with a bootstrapped anchor graph (Liu et al., 2022a).

FedAvg averages shared parameters after local optimization (McMahan et al., 2017). FedProx additionally penalizes deviations of shared parameters from the broadcast global model (Li et al., 2020).

Table 7: Client-count sensitivity of FedAvg + DCRN (%; mean ± sample standard deviation over five seeds).

Dataset	Clients	ACC	NMI	ARI	Macro-F1
Cora	2	64.02±3.92	48.34±3.20	40.38±4.77	58.81±3.37
	5	60.60±4.20	46.03±2.63	36.13±4.47	54.98±3.48
	10	57.38±6.20	42.76±4.37	33.58±6.36	52.46±5.41
	20	54.46±6.74	38.21±4.18	28.57±5.74	50.97±6.36
CiteSeer	2	48.51±3.32	21.20±2.48	20.73±2.57	44.40±4.19
	5	47.16±4.89	20.84±2.80	20.13±3.44	43.03±5.62
	10	46.13±5.17	20.32±3.08	19.02±3.32	42.68±6.04
	20	45.50±5.06	19.23±3.02	18.12±3.29	41.94±5.72
PubMed	2	48.20±11.59	10.92±9.66	8.95±12.07	41.30±16.19
	5	46.37±6.41	9.59±6.35	6.43±7.53	39.83±10.99
	10	49.76±4.53	11.13±5.09	8.28±6.64	44.03±8.41
	20	47.97±6.50	9.84±7.59	6.70±8.13	42.36±9.84
Amazon Photo	2	53.03±1.36	49.44±2.43	30.05±3.15	49.44±2.08
	5	62.94±3.16	51.28±3.41	40.31±3.96	56.53±5.13
	10	63.71±2.68	54.02±3.28	42.67±3.29	59.87±2.67
	20	45.36±10.71	38.93±10.17	22.61±10.49	39.56±14.46
Amazon Computers	2	48.23±1.97	42.79±2.64	30.17±1.54	43.10±2.60
	5	44.03±3.68	39.01±3.70	23.70±3.92	33.90±9.78
	10	41.47±5.22	34.73±5.74	20.29±5.87	26.97±7.81
	20	29.37±4.14	18.52±6.46	1.24±4.69	10.94±2.35

FedU^2 adds the flexible uniform regularizer (FUR) to the local backbone objective and uses the efficient unified aggregator (EUA) for server-side coordination (Liao et al., 2024). Our FedSGD adaptation computes one full-batch gradient per client at the shared model in each round. Aggregated gradients are applied through persistent server-side Adam, without local updates to shared parameters.

GCFGAE (Guo et al., 2023) uses the same owner assignments but replicates boundary nodes to retain cross-partition edges and synchronizes overlapping-node contributions during training. Replicas are excluded from evaluation. Our adaptation simulates communication in plaintext without private set intersection, secret sharing, or secure aggregation.

GCFGAE uses a 256–128–64 encoder with first-layer $\tanh$, an inner-product decoder, and full weighted binary cross-entropy. Training uses float64 computation and Adam for 200 epochs with learning rate 0.01 and gradient clipping at 10.

Final embeddings from DCRN, SUBLIME, and GCFGAE are clustered by global k-means with the public cluster count, `n_init=20`, and `random_state=42`. DeSE obtains predictions by taking the argmax of its first-level assignment matrix.

G.3 Client-Count Sensitivity

We evaluate FedSuCo with Federated vMF, FedAvg + DCRN, FedSGD + DCRN, and GCFGAE at $C \in \{2, 5, 10, 20\}$. All methods share fixed METIS owner assignments for each dataset and client count. GCFGAE retains cross-partition edges through boundary-node replicas, which are excluded from evaluation so that each original node is counted once. Each configuration is run independently and evaluated at its prespecified final checkpoint without label-based selection.

Tables 7, 8, 9, and 10 report ACC, NMI, ARI, and Macro-F1 as percentages (mean ± sample standard deviation over five seeds).

Table 8: Client-count sensitivity of FedSGD + DCRN (%; mean ± sample standard deviation over five seeds).

Dataset	Clients	ACC	NMI	ARI	Macro-F1
Cora	2	63.95±4.16	47.71±3.72	40.04±5.46	58.70±3.58
	5	59.25±4.22	43.34±3.01	34.18±3.93	54.07±3.61
	10	55.76±6.43	40.04±3.99	30.89±5.86	50.99±8.59
	20	50.01±6.28	34.31±5.04	24.47±4.99	46.93±6.31
CiteSeer	2	47.25±2.31	20.39±1.11	19.68±1.49	42.55±3.06
	5	45.43±4.09	18.67±2.24	18.12±2.67	40.91±4.74
	10	43.29±6.50	17.81±3.14	16.65±3.50	39.43±7.16
	20	42.19±5.02	16.35±2.86	15.20±3.22	38.16±5.94
PubMed	2	41.67±6.11	2.46±5.07	1.70±3.64	31.89±6.96
	5	43.27±6.08	4.56±6.23	4.13±6.66	35.04±9.33
	10	43.32±6.32	4.53±6.19	4.23±6.61	35.29±9.22
	20	42.30±4.40	4.65±5.13	2.72±5.13	33.38±7.69
Amazon Photo	2	53.13±1.42	49.75±2.15	30.17±3.27	49.86±2.38
	5	64.44±4.24	53.44±3.54	42.05±4.69	58.24±6.08
	10	64.72±1.97	54.48±2.89	42.50±3.29	60.54±2.70
	20	47.01±11.73	39.10±11.90	23.17±11.60	41.04±15.26
Amazon Computers	2	48.43±2.33	43.07±2.60	30.50±1.91	43.26±2.98
	5	44.23±3.81	41.30±3.65	25.79±4.07	34.78±9.51
	10	42.47±6.09	35.83±6.06	20.32±5.88	28.39±9.65
	20	30.10±3.75	17.95±6.83	1.11±4.63	10.54±2.50

Table 9: Client-count sensitivity of GCFGAE (%; mean ± sample standard deviation over five independently executed runs).

Dataset	Clients	ACC	NMI	ARI	Macro-F1
Cora	2	60.10±8.94	49.20±4.53	36.84±8.60	59.16±10.68
	5	60.10±8.94	49.20±4.53	36.84±8.60	59.16±10.68
	10	60.10±8.94	49.20±4.53	36.84±8.60	59.16±10.68
	20	60.10±8.94	49.20±4.53	36.84±8.60	59.16±10.68
CiteSeer	2	34.91±4.09	18.04±4.23	6.84±2.01	31.61±4.51
	5	34.91±4.09	18.04±4.23	6.84±2.01	31.61±4.51
	10	34.91±4.09	18.04±4.23	6.84±2.01	31.61±4.51
	20	34.91±4.09	18.04±4.23	6.84±2.01	31.61±4.51
PubMed	2	60.88±2.96	25.40±2.58	21.13±3.62	60.57±2.54
	5	60.88±2.96	25.40±2.58	21.13±3.62	60.57±2.54
	10	60.88±2.96	25.40±2.58	21.13±3.62	60.57±2.54
	20	60.88±2.96	25.40±2.58	21.13±3.62	60.57±2.54
Amazon Photo	2	19.16±0.21	2.65±0.20	0.26±0.02	13.21±0.06
	5	21.43±3.10	5.86±4.84	2.11±2.54	16.38±4.97
	10	19.25±0.09	2.51±0.30	0.28±0.03	13.45±0.41
	20	19.09±0.40	2.36±0.71	0.26±0.12	13.54±0.40
Amazon Computers	2	19.92±0.14	2.53±0.87	0.59±0.14	10.57±0.28
	5	19.30±1.23	2.39±0.77	0.57±0.22	10.96±0.65
	10	19.31±1.19	1.77±0.73	0.41±0.17	10.65±0.34
	20	19.11±1.77	2.27±1.85	0.55±0.48	11.08±1.05

Table 10: Client-count sensitivity of FedSuCo with Federated vMF (%; mean ± sample standard deviation over five seeds). A dash indicates an out-of-memory (OOM) failure.

Dataset	Clients	ACC	NMI	ARI	Macro-F1
Cora	2	68.30±1.95	53.28±1.05	46.72±1.38	63.91±3.38
	5	71.56±1.15	56.22±1.06	51.00±1.50	67.99±2.14
	10	74.13±1.89	56.20±1.64	52.49±2.98	71.11±3.80
	20	68.59±3.93	54.96±2.10	47.80±4.48	64.00±5.63
CiteSeer	2	63.09±1.98	36.99±0.92	36.69±0.95	59.41±1.41
	5	62.81±0.67	35.37±0.77	35.34±1.01	58.96±1.06
	10	58.77±2.59	32.45±1.64	30.81±2.18	55.48±2.23
	20	58.91±3.25	31.73±2.08	31.66±2.85	54.99±3.26
PubMed	2	—	—	—	—
	5	—	—	—	—
	10	64.69±1.34	23.68±1.27	23.40±1.90	64.84±1.24
	20	61.22±1.43	20.89±0.80	18.69±1.58	61.53±1.37
Amazon Photo	2	66.70±7.07	61.25±4.95	49.42±6.46	60.79±7.41
	5	63.13±7.51	59.05±2.86	47.36±5.29	56.09±9.39
	10	64.61±8.70	58.71±6.80	44.67±10.57	61.69±9.65
	20	64.73±8.35	59.53±3.63	45.54±7.88	59.93±9.16
Amazon Computers	2	—	—	—	—
	5	45.06±2.99	45.34±1.53	26.86±3.62	38.71±3.29
	10	40.19±4.32	41.32±2.57	22.04±1.57	35.13±5.35
	20	42.31±3.50	42.58±2.33	24.37±3.41	35.39±2.57