UNCERTAINTY-AWARE FEDERATED GRAPH-LEVEL CLUSTERING WITH MISSING ATTRIBUTES

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph-level clustering (FGC) has attracted increasing attention for privacy-preserving multi-graph analysis. However, existing methods typically assume fully observed node attributes, which often fail in cross-domain non-independent and identically distributed (non-IID) settings due to device failures or limited data access, while FGC with missing attributes remains largely underexplored. To address this, we propose **U**ncertainty-Aware **Fed**erated **G**raph-Level **C**lustering with Missing Attributes **(UFedGC)**, which couples attribute completion and graph-level clustering through uncertainty-aware closed-loop learning and cross-client semantic alignment. Specifically, UFedGC learns disentangled shared and private graph representations and uses evolving cluster assignments to guide attribute completion; the completed graphs are re-encoded to refine the assignments, forming a bidirectional interaction between attribute inference and clustering. Local clusters are then summarized into uncertainty-aware distributional statistics for reliable global semantic alignment and knowledge aggregation across heterogeneous clients, so that uncertain or domain-specific information contributes adaptively rather than being blindly incorporated. Extensive experiments demonstrate the superiority of UFedGC.

1 INTRODUCTION

Federated Graph Learning (FGL) (Tan et al., 2023) has emerged as a paradigm for privacy-preserving learning on decentralized graph data (Liu et al., 2023; 2025b). Most FGL studies focus on supervised settings, whereas labels are often unavailable in practice, motivating unsupervised learning of global structures in decentralized graphs. To this end, FGC (Liu et al., 2025a) clusters entire graphs while preserving privacy and has shown strong potential in applications such as drug discovery (Manu et al., 2021), social network analysis (Caldarola et al., 2021), and bioinformatics (Pfeifer et al., 2023).

However, existing FGC methods typically assume fully observed node attributes. This assumption simplifies algorithm design and theoretical analysis, but deviates from real-world graph data, where missing attributes are prevalent (Chen et al., 2020). In scenarios such as social networks, a substantial proportion of node attributes is missing due to privacy constraints, user inactivity, or limited data collection (Zhao et al., 2025). Such missingness can distort the graph representations used for clustering, blur the distinction between graphs from different clusters, and yield inaccurate local cluster prototypes, as illustrated in Fig. 1. More critically, in the federated setting, these local errors are incorporated into global aggregation and fed back to clients, allowing the bias to propagate across communication rounds and progressively degrade both global knowledge and subsequent local clustering. Therefore, an effective FGC framework should not only perform graph-level clustering under incomplete node attributes, but also incorporate reliable attribute completion to prevent missingness-induced errors from propagating throughout federated learning.

An intuitive solution is to jointly address attribute missingness and cross-client semantic inconsistency. However, this requires overcoming two tightly coupled challenges. First, missing attributes can distort graph representations and clustering assignments, while independent attribute completion may introduce errors that propagate into clustering and accumulate over communication rounds. Second, client heterogeneity and completion uncertainty make local cluster knowledge unreliable and semantically inconsistent, making direct prototype aggregation susceptible to cluster mismatch, data imbalance, and noisy updates. These challenges call for two complementary mechanisms: 1) a

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.

closed-loop interaction between attribute completion and clustering, where clustering guides attribute completion and completed attributes are constrained by cluster semantics; and 2) an uncertainty-aware knowledge collaboration mechanism that evaluates the reliability of local cluster information, aligns heterogeneous cluster semantics, and aggregates only trustworthy knowledge.

Based on these insights, we propose UFedGC, an uncertainty-aware FGC framework that couples attribute completion with clustering to enable reliable cross-client collaboration without sharing raw graph data. On each client, UFedGC disentangles graph representations into shared and private features and derives soft cluster assignments from the shared features and global cluster priors. These assignments guide missing attribute completion, and the completed graphs are re-encoded to enforce clustering consistency with the original representations, forming a closed loop between completion and clustering. UFedGC further estimates completion uncertainty and incorporates it into local optimization and the construction of uncertainty-aware cluster statistics. Each client uploads only the updated shared parameters and cluster statistics. The server aligns local cluster semantics with a global cluster bank based on distributional similarity and uncertainty, then performs uncertainty-aware aggregation to update the global cluster priors and shared model. The updated global knowledge is returned to clients for the next round, closing the loop and progressively improving attribute completion and global clustering. Our main contributions can be summarized as follows:

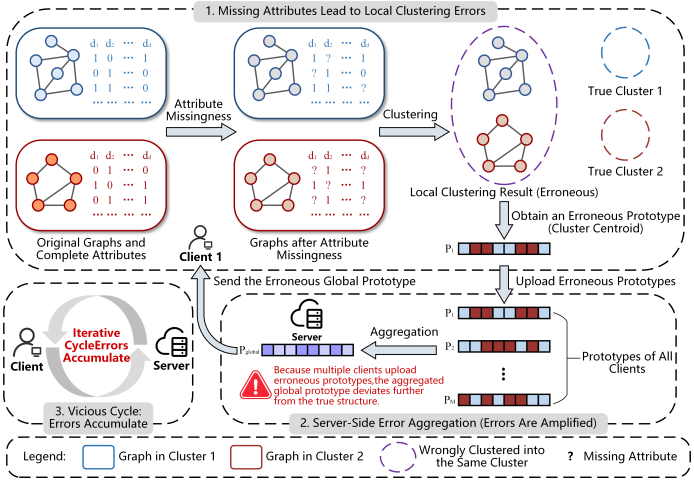

Figure 1: Missing node attributes can cause graphs from different clusters to be incorrectly grouped, leading to inaccurate local prototypes. These errors are amplified through global aggregation and propagated to subsequent rounds, forming a vicious cycle.

- **New Research Task.** To the best of our knowledge, we make the first attempt to tackle the issue of missing node attributes in the cross-domain FGC task.
- **New FGC Framework.** We propose UFedGC, a novel uncertainty-aware framework that jointly performs attribute completion and graph-level clustering.
- **Better Clustering Results.** Extensive experiments demonstrate that UFedGC consistently outperforms state-of-the-art baselines in graph clustering performance.

2 METHODOLOGY

This section presents UFedGC, an FGC framework for learning reliable cluster semantics under missing node attributes. UFedGC jointly performs attribute completion and graph-level clustering in a closed loop: cluster-aware completion improves graph representations, while clustering consistency prevents completion errors from distorting the cluster structure. To support reliable knowledge aggregation across heterogeneous clients, UFedGC quantifies the uncertainty of completed attributes and incorporates it into both local cluster statistics and global knowledge aggregation. As shown in Fig. 2, UFedGC consists of three modules: local closed-loop attribute completion and clustering, uncertainty-aware cross-client cluster alignment, and global prior and model aggregation.

2.1 NOTATIONS

We consider a federated setting with M clients, where each client privately holds a collection of graphs with missing node attributes, and the objective is to partition the graphs into C clusters. For

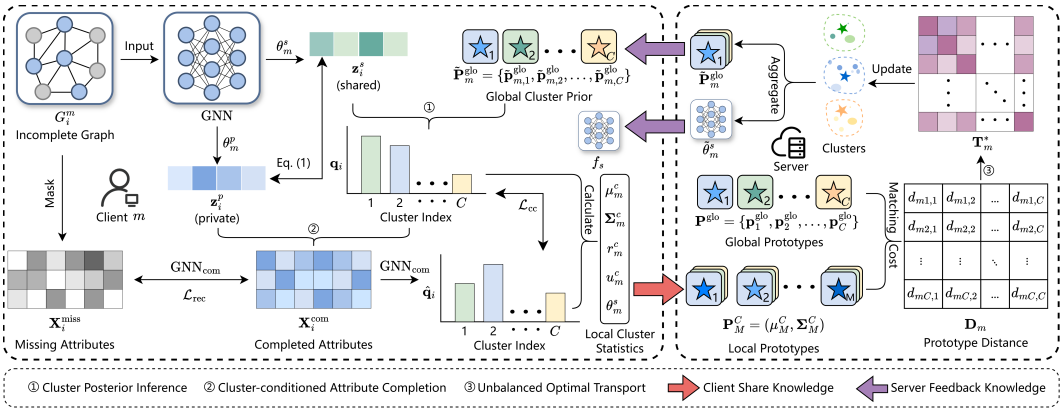

Figure 2: The framework of UFedGC. Learns shared and private representations and uncertainty-aware, cluster-conditioned attribute completion, while jointly constraining completion and clustering consistency. The server uses local cluster statistics and shared GNN parameters $\boldsymbol{\theta}_m^s$ to align clusters via unbalanced optimal transport and generate global priors for the next-round local optimization.

ease of understanding, we consider the m-th client as an example and denote the set of graphs it holds by $\mathcal{G}^m = (G_1^m, G_2^m, \ldots, G_I^m)$, where I denotes the total number of graphs held by the m-th client. For each G_i^m, the node feature matrix is denoted by $\mathbf{X}_i \in \mathbb{R}^{N\times d}$, and the normalized adjacency matrix by $\mathbf{A}_i \in \mathbb{R}^{N\times N}$, where N is the number of nodes in G_i^m and d is the dimension of the node features. Specifically, given the original adjacency matrix $\mathbf{A}_i^{\text{ori}} \in \mathbb{R}^{N\times N}$, we obtain $\mathbf{A}_i$ using symmetric normalization as $\mathbf{A}_i^{'} = \mathbf{A}_i^{\text{ori}} + \mathbf{I}$, $\mathbf{A}_i = \mathbf{D}_i^{'-\frac{1}{2}}\mathbf{A}_i^{'}\mathbf{D}_i^{'-\frac{1}{2}}$, where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix and $\mathbf{D}_i^{'}$ is the degree matrix of $\mathbf{A}_i^{'}$. A detailed list of symbols is provided in the Appendix B.

2.2 Local Closed-Loop Attribute Completion and Clustering

In this section, each client jointly performs graph representation learning, cluster inference, and missing-attribute completion instead of treating them separately. Client m uses the global cluster prior received from the previous round to infer a soft cluster posterior for each incomplete graph. The inferred cluster semantics condition attribute completion, and the completed graph is re-encoded to refine its cluster assignment. To prevent unreliable completion from dominating local clustering, UFedGC estimates graph-wise completion uncertainty and uses it to build local cluster statistics. Thus, clustering guides completion, and the completed attributes in turn refine clustering.

Representation Disentanglement For each graph G_i^m, we first feed it into a local GNN encoder f_{loc} to obtain the local graph-level representation $\mathbf{z}_i = f_{\text{loc}}(G_i^m, \boldsymbol{\theta}_{\text{loc}})$, where $\mathbf{z}_i \in \mathbb{R}^{d'}$, with d' denoting the embedding dimension (Liu et al., 2025a). Next, we initialize the parameters of the local shared GNN f_s with its own parameters in the first communication round and with the global shared parameters in subsequent rounds. We then feed G_i^m into f_s to obtain the shared representation $\mathbf{z}_i^s = f_s(G_i^m, \boldsymbol{\theta}_m^s)$, where $\mathbf{z}_i^s \in \mathbb{R}^{d'}$. Meanwhile, a private encoder f_p is employed to extract graph-specific information and produce the private representation $\mathbf{z}_i^p = f_p(G_i^m, \boldsymbol{\theta}_m^p)$, where $\mathbf{z}_i^p \in \mathbb{R}^{d'}$. To disentangle the shared and private representations, we maximize the mutual information between $\mathbf{z}_i^p$ and $\mathbf{z}_i$ while minimizing that between $\mathbf{z}_i^p$ and $\mathbf{z}_i^s$:

$$\boldsymbol{\theta}_m^p = \arg\max_{\boldsymbol{\theta}_m^p} \left[I(\mathbf{z}_i^p; \mathbf{z}_i) - I(\mathbf{z}_i^p; \mathbf{z}_i^s) \right], \tag{1}$$

where $I(\cdot;\cdot)$ denotes mutual information, estimated by a neural estimator using samples from the joint and marginal distributions (Belghazi et al., 2018).

Closed-Loop Attribute Completion and Clustering Building upon the disentangled representations obtained above, we next construct the cluster prior for client m. In the first communication round, client m initializes its cluster prior by applying the local prototype function $f_{\text{loc}}(G_i^m, \boldsymbol{\theta}_{\text{loc}})$ to its local graphs, yielding the local cluster prior. In subsequent communication rounds, the client instead uses the global cluster prior received from the server, denoted by $\tilde{\mathbf{P}}_m^{\text{glo}} = \{\tilde{\mathbf{p}}_{m,1}^{\text{glo}}, \ldots, \tilde{\mathbf{p}}_{m,C}^{\text{glo}}\}$, where

$\tilde{\mathbf{p}}^{\mathrm{glo}}_{m,c} \in \mathbb{R}^{d'}$. UFedGC measures the compatibility between the shared graph-level representation and each global cluster prior and obtains the soft posterior $\mathbf{q}_i \in \mathbb{R}^C$ as:

$$q_{ic} = \frac{\exp\big(-d\big(\mathbf{z}^s_i, \tilde{\mathbf{p}}^{\mathrm{glo}}_{m,c}\big)\big)}{\sum_{k=1}^{C} \exp\Big(-d\Big(\mathbf{z}^s_i, \tilde{\mathbf{p}}^{\mathrm{glo}}_{m,k}\Big)\Big)}, \qquad c = 1, \ldots, C, \tag{2}$$

where $d(\cdot, \cdot)$ is the cosine distance. Compared with assigning a hard pseudo-label, $\mathbf{q}_i$ retains the ambiguity of incomplete graphs and provides softer semantic guidance for subsequent completion.

We next perform cluster-conditioned attribute completion. Instead of relying solely on the incomplete graph, $\mathrm{GNN}_{\mathrm{com}}$ generates a cluster-specific candidate conditioned on local graph representations and the global cluster prior. The candidates are combined using the posterior probabilities:

$$\mathbf{X}^{\mathrm{com}}_i = \sum_{c=1}^{C} q_{ic}\, g_{\boldsymbol{\phi}}\big(\mathbf{A}_i, \mathbf{z}^s_i, \mathbf{z}^p_i, \tilde{\mathbf{p}}^{\mathrm{glo}}_{m,c}\big)\,, \tag{3}$$

where $\mathbf{X}^{\mathrm{com}}_i \in \mathbb{R}^{N \times d}$ denotes the predicted missing attributes and $\boldsymbol{\phi}$ denotes the parameters of the completion network. Completion is not treated as the endpoint of local inference. We feed the completed graph into the completion f_{com} encoder and re-estimate its cluster posterior:

$$\mathbf{z}^{\mathrm{com}}_i = f_{\mathrm{com}}(\mathbf{A}_i, \mathbf{X}^{\mathrm{com}}_i; \boldsymbol{\theta}^{\mathrm{com}})\,, \qquad \hat{q}_{ic} = \frac{\exp(-d(\mathbf{z}^{\mathrm{com}}_i, \mathbf{p}^{\mathrm{com}}_c))}{\sum_{k=1}^{C} \exp(-d(\mathbf{z}^{\mathrm{com}}_i, \mathbf{p}^{\mathrm{com}}_k))}. \tag{4}$$

Here, $\mathbf{z}^{\mathrm{com}}_i \in \mathbb{R}^{d'}$ is the completed graph-level embedding, $\hat{\mathbf{q}}_i \in \mathbb{R}^C$, and $\mathbf{p}^{\mathrm{com}}_c \in \mathbb{R}^{d'}$ denotes the prototype (i.e., cluster center) of the c-th cluster, learned by $\mathrm{GNN}_{\mathrm{com}}$ from the completed graph set $\mathcal{G}_{\mathrm{com}}$. The posterior before attribute completion captures cluster semantics from incomplete evidence, while the posterior after completion assesses whether the recovered attributes preserve them.

To jointly optimize completion and clustering, we randomly mask a subset of originally observed attributes and reconstruct them as self-supervised evidence. Meanwhile, a cluster-consistency regularizer prevents completion from moving a graph toward an inconsistent cluster:

$$\mathcal{L}_{\mathrm{com}} = \alpha \underbrace{\frac{1}{I}\sum_{i=1}^{I} \ell_{\mathrm{rec}}\big(\mathbf{X}^{\mathrm{com}}_i, \mathbf{X}^{\mathrm{miss}}_i\big)}_{\mathcal{L}_{\mathrm{rec}}} + \beta \underbrace{\frac{1}{I}\sum_{i=1}^{I} D_{\mathrm{KL}}(\mathbf{q}_i \| \hat{\mathbf{q}}_i)}_{\mathcal{L}_{\mathrm{cc}}}, \tag{5}$$

where ℓ_{rec} is evaluated only on randomly masked observed entries, $\mathbf{X}^{\mathrm{miss}}_i = \mathrm{Mask}(\mathbf{X}_i)$, $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ denotes the Kullback-Leibler divergence, and α, β are hyperparameters weighting the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ and the consistency loss $\mathcal{L}_{\mathrm{cc}}$, respectively. Hence, $\mathcal{L}_{\mathrm{rec}}$ learns recoverable structure–attribute dependencies, while $\mathcal{L}_{\mathrm{cc}}$ prevents completion errors from distorting latent cluster identity.

Uncertainty Evaluation Since the reliability of inferred attributes varies substantially across graphs, UFedGC incorporates a soft-label discrepancy measure to quantify completion uncertainty. Specifically, for graph G_i, we leverage three soft cluster assignment vectors: $\mathbf{q}_i$, $\hat{\mathbf{q}}_i$, and the previous communication round's post-completion assignment $\hat{\mathbf{q}}^{(t-1)}_i$, setting $\hat{\mathbf{q}}^{(0)}_i = \mathbf{q}^{(1)}_i$. The completion uncertainty u_i and its corresponding reliability weight w_i are defined as:

$$u_i = D_{\mathrm{KL}}(\mathbf{q}_i \parallel \hat{\mathbf{q}}_i) + D_{\mathrm{KL}}(\hat{\mathbf{q}}^{(t-1)}_i \parallel \hat{\mathbf{q}}_i), \qquad w_i = \exp(-u_i). \tag{6}$$

Here, $u_i \geq 0$ measures the divergence across feature completions and historical dynamics, ensuring that graphs with unstable or inconsistent assignments receive lower reliability weights $w_i \in (0, 1]$ during downstream aggregation and clustering. A graph with uncertain completion therefore contributes less to the subsequent estimation of local cluster semantics.

Finally, rather than uploading a single centroid, client m summarizes every local cluster using local cluster statistics. Let $a_{mi,c} = w_{mi}\hat{\mathbf{q}}_{mi,c}$ and define $r^c_m = \frac{(\sum_i a_{mi,c})^2}{\sum_i a^2_{mi,c}}$, $\boldsymbol{\mu}^c_m = \frac{\sum_i a_{mi,c}\mathbf{z}^{com}_{mi}}{\sum_i a_{mi,c}}$, $\boldsymbol{\Sigma}^c_m = \frac{\sum_i a_{mi,c}(\mathbf{z}^{com}_{mi} - \boldsymbol{\mu}^c_m)^{\odot 2}}{\sum_i a_{mi,c}}$, and $u^c_m = \frac{\sum_i a_{mi,c} u_{mi}}{\sum_i a_{mi,c}}$, forming $\mathbf{S}^c_m = (\boldsymbol{\mu}^c_m, \boldsymbol{\Sigma}^c_m, r^c_m, u^c_m)$ to represent the prototype mean vector, diagonal covariance vector, effective sample size, and uncertainty, respectively.

Uploading only $\boldsymbol{\theta}_m^s$ while keeping $\boldsymbol{\theta}_m^p$ local aggregates shared knowledge across clients without leaking private data or compromising personalized representations.

In summary, local closed-loop completion and clustering: (1) form a bidirectional dependency between global semantic guidance and attribute recovery to refine cluster assignments; and (2) transform noisy graphs into reliability-aware sufficient statistics, offering richer clustering evidence than conventional prototypes.

2.3 Uncertainty-Aware Cross-Client Cluster Alignment

The previous subsection yields a set of local cluster statistics $\{\mathbf{S}_m^c\}_{c=1}^C$ together with the $\boldsymbol{\theta}_m^s$ for each client. However, the same cluster index does not necessarily carry the same semantics across heterogeneous clients, and severely incomplete or weakly supported local clusters should not contribute equally to global knowledge construction. Therefore, before global aggregation, the server aligns local cluster distributions with the global cluster space using their means, covariances, and uncertainty. UFedGC formulates this alignment as uncertainty-aware unbalanced optimal transport, avoiding artificial one-to-one matches for unreliable or absent local clusters.

Assume that the server maintains C global clusters, where the c-th global cluster is described by $(\boldsymbol{\mu}_{\text{glo}}^c, \boldsymbol{\Sigma}_{\text{glo}}^c, r_{\text{glo}}^c, u_{\text{glo}}^c)$. For client m, we first construct a matching-cost matrix $\mathbf{D}_m \in \mathbb{R}^{C\times C}$, where its (j,c)-th element measures the discrepancy between local cluster j and global cluster c:

$$[\mathbf{D}_m]_{jc} = \text{NOR}(\left\|\boldsymbol{\mu}_m^j - \boldsymbol{\mu}_{\text{glo}}^c\right\|_2^2) + \text{NOR}(\left\|\boldsymbol{\Sigma}_m^j - \boldsymbol{\Sigma}_{\text{glo}}^c\right\|_2^2) + \text{NOR}(\left|u_m^j - u_{\text{glo}}^c\right|), \tag{7}$$

where $\text{NOR}(\cdot)$ denotes a normalization function (e.g., Min-Max normalization). Consequently, two clusters are semantically compatible only when both their representations and reliability profiles are consistent. Besides modifying the matching cost, uncertainty controls the transport mass a local cluster can contribute, with source and target masses defined as:

$$[\mathbf{a}_m]_j = r_m^j \exp(-u_m^j), \qquad [\mathbf{b}]_c = r_{\text{glo}}^c \exp(-u_{\text{glo}}^c), \tag{8}$$

where $\mathbf{a}_m \in \mathbb{R}^C$ and $\mathbf{b} \in \mathbb{R}^C$. Thus, a cluster supported by more reliable local evidence carries larger transport mass, whereas a highly uncertain or weakly supported cluster contributes only limited information. UFedGC then obtains the cross-client cluster alignment matrix by minimizing the following unbalanced optimal transport objective $\mathcal{L}_{\text{UOT}}$:

$$\mathcal{L}_{\text{UOT}} = \langle \mathbf{T}_m, \mathbf{D}_m \rangle + \sum_{j,c} \mathbf{T}_{m,jc}(\log \mathbf{T}_{m,jc} - 1) + \text{KL}_{\text{gen}}(\mathbf{T}_m\mathbf{1} \parallel a_m) + \text{KL}_{\text{gen}}(\mathbf{T}_m^\top\mathbf{1} \parallel b), \tag{9}$$

where $\mathbf{T}_m^* = \arg\min_{\mathbf{T}_m \geq 0} \mathcal{L}_{\text{UOT}}$ and $\text{KL}_{\text{gen}}(p \parallel q) = \sum_k \left[p_k \log \frac{p_k}{q_k} - p_k + q_k\right]$. Unlike balanced optimal transport, the relaxed marginals allow transport mass to be created or discarded at a penalty. This property is particularly important under cross-client non-IID distributions, since a local client may contain insufficient evidence for some global clusters or may not contain them at all.

Therefore, uncertainty-aware cross-client alignment plays two complementary roles: 1) $\mathbf{T}_m^*$ resolves the semantic permutation problem among independently learned local clusters and maps them into a common global cluster space; and 2) the uncertainty-dependent costs and transport masses attenuate unreliable or weakly supported local clusters, preventing noisy completion results from being amplified in global aggregation.

2.4 Global Prior and Model Aggregation

Given the $\{\mathbf{T}_m^*\}_{m=1}^M$, the server aggregates aligned local knowledge rather than unmatched prototypes. At the cluster level, it combines transported sufficient statistics to update the global cluster prior for the next round. At the model level, it aggregates only the shared GNN parameters, weighted by each client's amount of reliable clustering evidence. The updated prior and shared encoder are sent back to clients, closing the optimization loop. For global cluster c, let $\hat{s} = \sum_{m=1}^M \sum_{j=1}^C [\mathbf{T}_m^*]_{jc}$ denote its transported evidence. The $\hat{\boldsymbol{\mu}}_{\text{glo}}^c$ are updated by moment matching:

$$\hat{\boldsymbol{\mu}}_{\text{glo}}^c = \frac{\sum_{m=1}^M \sum_{j=1}^C [\mathbf{T}_m^*]_{jc} \boldsymbol{\mu}_m^j}{\hat{s}}. \tag{10}$$

The other three cluster statistics are updated using a method similar to the local one. We use $\hat{\mathbf{p}}_c^{\text{glo}} = \hat{\boldsymbol{\mu}}_{\text{glo}}^c$ as the prototype representation of global cluster c and form $\hat{\mathbf{P}}^{\text{glo}} \in \mathbb{R}^{C\times d'}$ by stacking all global prototypes. In parallel, the server aggregates shared GNN parameters, weighted by each client's amount of reliable cluster evidence. Let $\eta_m = \sum_{c=1}^{C} r_m^c \exp(-u_m^c)$ denote the reliability of client m. The shared-model update and server feedback are:

$$\hat{\boldsymbol{\theta}}^s = \frac{\sum_{m=1}^{M} \eta_m \boldsymbol{\theta}_m^s}{\sum_{m=1}^{M} \eta_m}, \qquad \left(\tilde{\mathbf{P}}_m^{\text{glo}}, \tilde{\boldsymbol{\theta}}_m^s\right) \leftarrow \left(\hat{\mathbf{P}}^{\text{glo}}, \hat{\boldsymbol{\theta}}_m^s\right), \quad \forall m, \tag{11}$$

while the private parameters $\boldsymbol{\theta}_m^p$ remain on each client and are never globally averaged. As a result, reliable clients make stronger contributions to the shared representation space, whereas clients dominated by uncertain completions have a reduced effect on the global model.

Overall, global prior and model aggregation serve two functions: 1) it converts aligned local cluster distributions into a globally consistent semantic prior that guides the next round of cluster posterior inference and cluster-conditioned attribute completion; and 2) it builds an uncertainty-aware shared representation space while preserving client-specific information, closing the loop from global knowledge to local completion, clustering, reliable statistics, and global alignment. Appendix C describes the UFedGC learning process in detail.

2.5 Communication Efficiency Analysis

Compared with FedGCN (Liu et al., 2025a), our framework incurs only modest additional communication overhead. FedGCN uploads the model parameters and the mean prototype of each cluster, with complexity $\mathcal{O}(P + Cd')$, where P is the number of model parameters. Our method uploads only the shared parameters $\boldsymbol{\theta}_m^s$ and local cluster statistics $\mathbf{S}_m^c$, with complexity $\mathcal{O}(P_s + C(2d' + 2))$, where $P_s < P$. Here, $\boldsymbol{\mu}_m^c$ and $\boldsymbol{\Sigma}_m^c$ each contain d' elements, and r_m^c and u_m^c are scalars. Hence, the communication complexity of UFedGC is linearly related to that of FedGCN, as both are $\mathcal{O}(P+Cd')$. Our communication cost is thus only slightly higher than that of FedGCN, with the extra overhead coming from the diagonal covariance, effective sample size, and uncertainty. This modest increase enables more informative prototype aggregation and yields substantial performance gains.

3 Experiments

3.1 Experimental Settings

Benchmark Datasets We evaluate UFedGC on 17 public datasets from the TUDataset collection, spanning small molecules, bioinformatics, computer vision, and social networks. These datasets are organized into six non-IID settings: same-domain (SM, SN, CV) and cross-domain (SM-BIO, SM-BIO-SY, SN-SY), following the non-IID settings adopted in FedGCN (Liu et al., 2025a). Detailed dataset information and non-IID settings are provided in Appendix D.

Baseline Methods To evaluate the effectiveness of UFedGC, we consider two types of baselines. The first group includes the SOTA FGC methods FedGCN (Liu et al., 2025a), FGCN-AKC (Han et al., 2026), FedCND (Wu et al., 2026), PERFECT (Liu et al., 2026), and FedPKA (Wu et al., 2025), as well as several representative FGL methods adapted to FGC, such as FedSage (Zhang et al., 2021), GCFL (Xie et al., 2021), FedStar (Tan et al., 2023), LG-FGAD (Cai et al., 2024a), FGAD (Cai et al., 2024b), and AGDiff (Cai et al., 2025). The second group comprises advanced centralized attributed graph completion methods with conventional FGC methods, including FedGCN + GLAD (Zuo et al., 2026), FedGCN + FSD-CAP (Qiao et al., 2026), and FedGCN + FISF (Um et al., 2025). The descriptions of the baselines and the implementation details of UFedGC are provided in Appendix E.

Implementation details For fair comparison, all methods are implemented in PyTorch and run on a single NVIDIA GeForce RTX 4090 GPU. Each client uses a three-layer GIN (Xu et al., 2018) to extract graph-level embeddings, with a batch size of 256 and a learning rate of 1e-3. For attribute-incomplete graphs, 50% of node attributes are randomly masked, with no feature entirely missing. Both local and global models are optimized using Adam (Kingma & Ba, 2014). Each client communicates with the server for 10 rounds, with its local model trained for 10 epochs per round.

Evaluation Metrics To comprehensively assess the performance of UFedGC, we adopt four standard evaluation metrics for unsupervised clustering: Accuracy (ACC) (Xie et al., 2016), Normalized

Table 1: Performance Comparison of Different FGC Methods under six non-IID settings. For fairness, all methods are evaluated on attribute-incomplete graphs unsupervisedly.

Methods	$\mathbf{SM}^2(7)$				$\mathbf{SN}^3(2)$				$\mathbf{SM\text{-}BIO}^2(9)$			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
FedSage*	54.3±2.4	10.4±1.7	6.5±0.5	45.5±1.2	49.0±1.9	12.8±1.7	10.1±3.1	43.5±1.9	51.5±0.9	4.7±0.2	3.5±0.4	45.4±0.7
GCFL*	57.2±1.6	7.4±0.7	7.8±0.3	46.3±2.0	46.1±2.6	10.7±2.3	11.7±2.7	46.0±1.5	54.2±2.0	4.0±0.7	2.7±0.3	44.2±1.2
FedStar*	53.2±1.5	10.3±1.7	0.1±0.1	44.1±2.5	48.6±2.6	12.0±1.0	10.2±1.3	47.8±2.3	56.1±1.8	4.4±0.1	3.1±0.2	47.4±2.2
FGAD†	61.5±2.4	17.9±1.5	3.5±0.2	56.7±2.1	37.0±0.5	5.2±0.1	0.4±0.1	34.0±0.6	55.8±1.0	12.0±0.6	1.7±0.2	53.7±1.2
LG-FGAD†	57.3±1.2	16.1±1.0	3.0±0.1	53.8±0.6	35.7±0.2	7.9±0.5	0.4±0.2	24.4±0.6	51.0±0.2	7.5±0.4	6.8±0.3	50.7±0.5
AGDiff†	62.5±1.6	16.5±1.6	13.8±1.4	61.7±2.7	36.7±0.6	6.3±0.7	7.1±0.2	31.7±0.8	53.5±1.4	8.9±0.1	2.8±0.1	51.9±0.7
FedGCN	66.5±0.7	19.3±1.3	26.5±3.8	59.3±1.5	57.3±2.3	25.8±2.5	30.3±2.3	43.8±1.1	59.3±2.1	11.6±1.6	15.2±1.8	50.3±0.6
FGCN-AKC	68.6±1.9	24.2±1.9	31.1±1.3	68.4±1.6	57.9±1.8	30.6±1.7	31.2±1.9	55.9±2.0	59.2±2.3	15.1±2.6	17.5±2.3	52.9±1.5
FedPKA	67.3±0.4	22.2±2.1	25.7±2.9	61.8±2.5	59.5±1.8	20.9±2.1	27.0±2.4	51.5±1.1	61.0±1.9	13.0±1.5	17.2±1.0	53.0±1.7
FedCND	61.6±1.5	21.6±0.6	25.9±0.6	60.5±1.5	54.8±0.9	27.0±2.0	31.5±2.5	49.0±1.5	58.8±1.7	12.5±1.4	15.9±2.1	49.8±2.1
PERFECT	62.6±1.1	20.7±1.9	26.8±1.3	60.8±2.3	55.4±0.3	29.9±1.9	30.3±1.0	43.5±1.3	56.5±0.6	14.2±1.1	18.2±1.6	54.5±2.8
FedGCN+GLAD	70.4±0.5	20.0±1.9	26.4±1.8	58.2±1.3	59.8±2.5	25.4±2.4	29.0±2.3	42.8±2.6	61.6±0.9	12.4±1.0	14.3±1.1	51.9±0.5
FedGCN+FSD-CAP	71.6±0.5	19.4±1.2	25.1±1.7	57.4±1.9	60.1±2.3	26.5±1.7	30.1±1.6	45.2±2.1	63.9±0.3	12.6±1.5	14.0±1.0	47.5±0.9
FedGCN+FISF	69.5±0.6	18.9±1.9	27.5±2.3	60.8±1.9	58.5±2.3	27.0±2.8	28.9±2.1	45.3±2.1	61.0±0.6	11.5±2.2	14.1±1.7	55.8±1.2
Ours	**74.4±0.4**	**24.8±2.2**	**31.2±1.8**	**68.8±1.7**	**65.9±2.0**	**33.4±2.2**	**35.8±2.0**	**59.7±2.0**	**69.5±2.4**	**16.5±1.2**	**19.6±1.7**	**62.9±1.0**
	$\mathbf{SM\text{-}BIO\text{-}SY}^2(10)$				$\mathbf{SN\text{-}SY}^{11}(2)$				$\mathbf{CV}^{15}(3)$			
FedSage*	51.2±0.1	14.8±1.5	15.5±2.8	43.2±1.8	14.5±0.9	6.5±0.8	3.0±0.2	2.5±0.2	15.8±0.8	19.4±0.4	10.4±1.4	14.8±1.0
GCFL*	54.9±1.0	12.0±2.3	11.4±2.4	47.2±1.9	17.2±0.4	3.9±0.2	1.1±0.1	7.0±0.6	23.1±2.0	19.3±2.3	11.0±1.4	20.5±1.3
FedStar*	53.7±0.4	13.1±2.4	13.2±1.1	47.3±1.3	17.2±0.7	3.4±0.3	1.9±0.2	7.0±0.9	20.5±1.0	16.7±1.2	8.4±1.2	18.7±3.4
FGAD†	51.8±0.9	12.0±1.9	2.6±0.2	45.6±0.5	13.9±0.1	6.1±0.5	2.3±0.8	7.5±0.3	22.1±1.3	27.5±1.5	6.2±0.4	21.6±0.8
LG-FGAD†	52.2±0.3	6.6±0.1	5.3±0.4	45.7±0.5	15.6±1.3	5.6±0.3	2.9±0.1	6.3±0.5	25.0±1.5	26.1±2.4	7.7±0.4	19.9±2.1
AGDiff†	53.3±0.3	13.7±1.1	11.8±1.0	46.1±1.2	14.2±0.9	3.6±0.1	2.8±0.2	8.2±0.3	21.0±1.2	22.6±1.0	7.6±0.4	18.8±1.6
FedGCN	58.8±1.6	11.5±1.9	14.5±1.0	55.1±3.5	17.0±1.5	3.9±0.3	2.0±0.2	10.3±0.9	31.6±2.6	30.4±2.2	15.9±1.4	26.9±2.6
FGCN-AKC	60.0±1.0	16.0±1.3	17.5±1.7	56.5±2.2	19.8±1.9	10.9±1.0	7.5±0.4	13.7±0.8	31.7±1.3	29.7±1.8	20.7±1.3	29.3±1.6
FedPKA	58.7±0.8	14.5±0.9	17.3±1.0	52.4±2.4	13.2±1.1	4.7±0.2	5.3±0.2	6.9±0.3	29.6±1.4	28.1±1.7	16.7±1.4	28.2±1.7
FedCND	60.3±1.8	14.4±0.5	15.5±0.9	53.5±2.6	17.3±1.5	4.2±0.2	3.9±0.1	9.9±1.1	29.2±0.9	30.3±1.5	17.5±0.8	28.3±1.0
PERFECT	58.2±0.1	14.8±1.3	18.0±1.2	56.1±2.7	18.2±1.8	9.3±1.0	8.8±0.7	9.2±1.2	29.6±1.3	30.7±0.9	17.5±1.1	29.4±0.9
FedGCN+GLAD	62.2±1.1	11.6±1.8	14.7±3.7	54.4±4.3	20.4±1.4	13.0±0.8	10.1±1.4	13.9±0.8	31.9±2.2	27.9±2.6	17.3±2.2	27.3±2.5
FedGCN+FSD-CAP	62.9±1.4	12.1±2.5	14.3±3.6	48.2±3.5	21.7±1.2	13.9±1.2	11.0±1.7	14.6±0.6	33.1±2.8	29.0±2.0	18.9±1.6	29.6±2.0
FedGCN+FISF	61.9±1.1	11.6±2.2	14.3±3.1	49.6±4.0	21.0±1.5	13.5±0.9	10.2±1.3	14.0±1.4	32.1±3.2	30.6±2.8	15.8±1.9	28.8±2.4
Ours	**69.4±1.3**	**17.6±1.8**	**19.4±0.7**	**63.1±0.4**	**23.0±1.6**	**19.5±1.5**	**12.5±2.1**	**21.7±1.1**	**33.3±1.9**	**31.7±2.5**	**21.2±1.9**	**29.9±1.4**

* Note that these supervised FGL methods are adapted to the unsupervised scenario.

† Note that these unsupervised FGL tasks are manually converted into FGC tasks.

Table 2: Module ablation study of UFedGC is conducted under four non-IID settings.

Variants	SM			SM-BIO			SM-BIO-SY			CV		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Basic	55.7±0.6	17.4±1.7	12.5±1.0	54.3±1.1	13.5±1.9	11.3±2.4	53.3±0.9	6.6±0.9	11.3±1.3	25.1±2.4	15.5±1.9	14.3±1.3
w/o LCC	62.6±0.6	21.3±2.3	27.9±2.0	61.2±2.0	14.7±0.7	17.1±1.9	61.4±1.1	14.4±1.1	16.6±0.3	29.2±2.1	28.0±2.7	19.0±1.8
w/o UNC	69.3±0.4	23.4±2.5	28.7±1.5	63.6±2.5	15.4±1.3	18.4±2.1	63.2±1.0	14.3±1.6	17.7±0.6	31.4±2.0	29.5±2.6	19.4±1.6
w/o CCA	62.1±0.8	20.0±2.5	25.6±1.7	57.6±2.4	14.0±1.5	16.5±1.7	58.2±1.2	14.1±1.3	15.7±1.2	27.4±1.4	26.7±2.0	18.0±2.3
Ours	**74.4±0.4**	**24.8±2.2**	**31.2±1.8**	**69.5±2.4**	**16.5±1.2**	**19.6±1.7**	**69.4±1.3**	**17.6±1.8**	**19.4±0.7**	**33.3±1.9**	**31.7±2.5**	**21.2±1.9**

Mutual Information (NMI) (Strehl & Ghosh, 2002), Adjusted Rand Index (ARI) (Hubert & Arabie, 1985), and F1 score (Powers, 2020). Details of these metrics are provided in Appendix G.

3.2 Comparison Experiments

Comparison with FGL Methods We compare UFedGC with advanced FGL methods for FGC under missing attributes. As shown in Table 1, UFedGC consistently achieves the best clustering performance under non-IID settings. This gain stems from jointly optimizing attribute completion and clustering, where cluster assignments guide completion and completed attributes refine representations. Uncertainty-aware prototype statistics and unbalanced optimal transport further align heterogeneous client semantics and reduce error accumulation from noisy local prototypes. Supervised methods depend on labels and thus cannot model representations or align semantics under missing attributes, while unsupervised anomaly detection targets rare graph patterns rather than general clustering. UFedGC also significantly outperforms SOTA FGC methods, indicating that jointly optimizing completion and clustering better addresses cross-client semantic heterogeneity.

Comparison with Centralized Attribute-Completion-Based Methods To determine whether the gains come solely from attribute completion, we compare UFedGC with three centralized completion methods combined with FedGCN. As shown in Table 1, UFedGC consistently outperforms these baselines, showing that completion before clustering is insufficient. UFedGC instead jointly optimizes completion and clustering: cluster assignments guide attribute inference, while completed attributes

preserve cluster semantics. It also models completion uncertainty in prototype and model aggregation, reducing the propagation of unreliable completions across clients.

Comparison with Supervised FGL Methods To evaluate UFedGC without label supervision, we compare it with supervised FGL methods provided with partial labels. As shown in Fig. 3, UFedGC achieves competitive or superior clustering performance despite using no labels. This is attributed to its uncertainty-aware closed-loop learning mechanism, which jointly exploits cluster assignments, attribute reconstruction, and cross-client semantic alignment to progressively refine graph representations. Consequently, UFedGC extracts discriminative and reliable clustering information directly from graph structure and attributes, reducing reliance on label supervision.

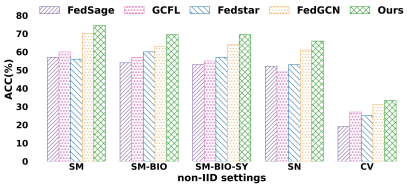

Figure 3: Comparison experiment results on the supervised methods with few labels under five non-IID settings.

3.3 Module Ablation Studies

We evaluate UFedGC under four non-IID settings by removing three mechanisms: local closed-loop completion and clustering, uncertainty modeling, and cross-client cluster alignment. We obtain **w/o LCC**, **w/o UNC**, **w/o CCA**, and **Basic**, which removes all three; **Ours** uses all. As shown in Table 2, removing any mechanism degrades performance, confirming their complementary roles. **w/o LCC** shows that coupling completion with clustering lets recovered attributes refine assignments; decoupling them weakens representations under missing attributes. **w/o UNC** shows that uncertainty modeling suppresses unreliable evidence; without it, noisy representations can be amplified under heterogeneity. **w/o CCA** shows that cluster indices may have different semantics across clients, so direct aggregation can mix mismatched statistics; CCA aligns local clusters into a common global space. Overall, **Ours** achieves the strongest performance on all three metrics. The gains from **Basic** to the enhanced variants and then to the full model show that local recovery, uncertainty awareness, and semantic alignment are jointly important for clustering with missing attributes.

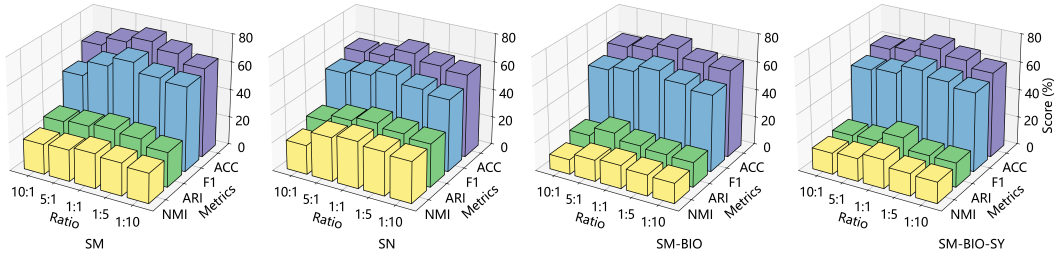

Figure 4: Hyper-parameters α and β sensitivity analysis results under four non-IID settings.

3.4 Hyper-parameter Sensitivity Analysis

To examine UFedGC's sensitivity to the weighting of reconstruction and consistency losses, we vary α:β from 1:10 to 10:1 under four non-IID settings. As shown in Fig. 4, UFedGC remains stable across a wide range of ratios, showing limited dependence on a specific setting. Excessive β may hinder attribute reconstruction, while excessive α weakens cluster consistency and causes completed attributes to deviate from their cluster identity. Overall, α:β = 1:1 provides the most stable performance across ACC, NMI, ARI, and F1.

3.5 Convergence Analysis

To assess the stability and convergence behavior of UFedGC, we track the trajectories of ACC, NMI, ARI, and F1 values across communication rounds. As shown in Fig. 5, our model converges smoothly and rapidly with stable performance on all metrics. This stable convergence stems from joint attribute completion and clustering, with consistency constraints reducing error propagation. Uncertainty-aware weighting limits unreliable updates, while prototype alignment reduces cross-client semantic

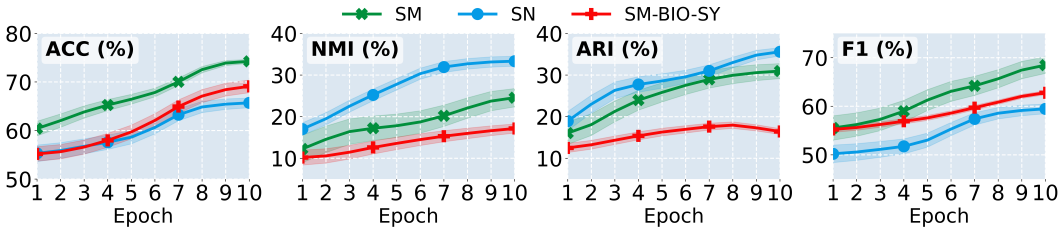

Figure 5: Convergence curves under three non-IID settings.

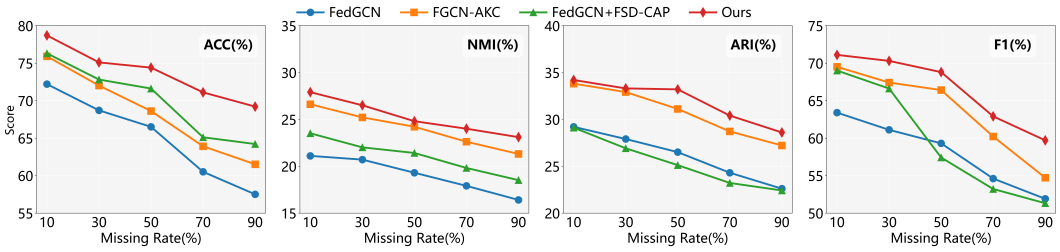

Figure 6: Clustering performance comparison under different missing rates in the SM non-IID setting.

inconsistency. As training proceeds, global representations and cluster priors are refined and converge to stable values, demonstrating reliable convergence under missing attributes and client heterogeneity.

3.6 Robustness Analysis under Varying Missing Rates

To evaluate robustness to missing features, we conduct experiments under the SM non-IID setting with missing rates from 10% to 90%. Fig. 6 shows the clustering performance. Overall, UFedGC consistently outperforms three SOTA methods across all missing rates. As the missing rate increases, all methods degrade due to feature information loss. However, baseline methods experience sharper performance drops at high missing rates, whereas our method degrades more gradually. Even with 90% missing features, UFedGC maintains competitive accuracy and stable performance, demonstrating strong robustness to incomplete non-IID graph data.

3.7 Runtime Comparison

As shown in Table 3, UFedGC achieves the lowest one global-epoch runtime across all settings. Compared with the baselines, it reduces runtime by at least 6.5%, demonstrating its computational efficiency. This advantage is mainly attributed to the integrated design, which avoids the substantial overhead introduced by separately performing attribute completion and graph clustering.

Table 3: One global-epoch training runtime (in seconds) of the evaluated baseline approaches.

Methods	SM	SM-BIO	SM-BIO-SY	CV
FedCND	44.3	220.8	253.5	42.2
FGCN-AKC	36.0	207.5	217.0	38.3
FedGCN+FSD-CAP	53.7	198.5	217.8	43.5
Ours	**33.0**	**185.6**	**190.9**	**28.3**

4 Conclusion

In this paper, we propose UFedGC, a novel framework for federated graph-level clustering under severe node attribute missingness. UFedGC tightly couples attribute completion and clustering through a closed-loop interaction, while leveraging uncertainty estimation to prevent error propagation across federated rounds. On the server, an unbalanced optimal transport mechanism aligns heterogeneous client semantics for reliable knowledge aggregation. Extensive experiments across non-IID benchmarks demonstrate that UFedGC consistently outperforms state-of-the-art baselines in clustering accuracy, convergence speed, and robustness to missing attributes.

AI USE STATEMENT

The authors declare that no Large Language Models (LLMs) or other AI tools were used in any stage of this work, including research ideation, execution, data processing, or manuscript writing.

REFERENCES

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Jinyu Cai, Yunhe Zhang, Jicong Fan, and See-Kiong Ng. Lg-fgad: An effective federated graph anomaly detection framework. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3760–3769, 2024a.

Jinyu Cai, Yunhe Zhang, Zhoumin Lu, Wenzhong Guo, and See-kiong Ng. Fgad: Self-boosted knowledge distillation for an effective federated graph anomaly detection framework. 2024b.

Jinyu Cai, Yunhe Zhang, Fusheng Liu, and See-Kiong Ng. Leveraging diffusion model as pseudo-anomalous graph generator for graph-level anomaly detection. In *Forty-Second International Conference on Machine Learning*, 2025.

Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2749–2758, 2021.

Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W Tsang. Learning on attribute-missing graphs. 44(2):740–757, 2020.

Renda Han, Xinyuan Li, Guangzhen Yao, Mengfei Li, Wenxin Zhang, Ronghao Fu, Zeyu Zhang, and Junlong Wu. Federated graph-level clustering network with adaptive knowledge compensation. *Neural Networks*, 195:108212, 2026. ISSN 0893-6080.

Lawrence Hubert and Phipps Arabie. Comparing partitions. 2:193–218, 1985.

D.P. Kingma and J.B. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

Jingxin Liu, Jieren Cheng, Renda Han, Wenxuan Tu, Jiaxin Wang, and Xin Peng. Federated graph-level clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18870–18878, 2025a.

Jingxin Liu, Wenxuan Tu, Haotian Wang, Renda Han, Junlong Wu, Guohui Liu, Xiangyan Tang, and Yang Yue. Personalized federated graph-level clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2026.

Meng Liu, Ke Liang, Dayu Hu, Hao Yu, Yue Liu, Lingyuan Meng, Wenxuan Tu, Sihang Zhou, and Xinwang Liu. TMac: Temporal multi-modal graph learning for acoustic event classification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3365–3374, 2023.

Meng Liu, Ke Liang, Hao Yu, Lingyuan Meng, Siwei Wang, Sihang Zhou, and Xinwang Liu. Multiview Temporal Graph Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):18383–18396, 2025b.

Daniel Manu, Yi Sheng, Junhuan Yang, Jieren Deng, Tong Geng, Ang Li, Caiwen Ding, Weiwen Jiang, and Lei Yang. Fl-disco: Federated generative adversarial network for graph-based molecule drug discovery: Special session paper. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–7. IEEE, 2021.

Bastian Pfeifer, Hryhorii Chereda, Roman Martin, Anna Saranti, Sandra Clemens, Anne-Christin Hauschild, Tim Beißbarth, Andreas Holzinger, and Dominik Heider. Ensemble-gnn: federated ensemble learning with graph neural networks for disease module discovery and classification. *Bioinformatics*, 39(11):btad703, 2023.

David MW Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

Xin Qiao, Shijie Sun, Anqi Dong, Cong Hua, Xia Zhao, Longfei Zhang, Guangming Zhu, et al. Fsd-cap: Fractional subgraph diffusion with class-aware propagation for graph feature imputation. In *International Conference on Learning Representations*, volume 2026, pp. 79504–79535, 2026.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9953–9961, 2023.

Daeho Um, Sunoh Kim, Jiwoong Park, Jongin Lim, Seong Jin Ahn, and Seulki Park. Propagate and inject: Revisiting propagation-based feature imputation for graphs with partially observed features. In *Forty-second International Conference on Machine Learning*, 2025.

Junlong Wu, Haotian Wang, Jingxin Liu, Wenxuan Tu, Renda Han, Jieren Cheng, and Xiangyan Tang. Fedpka: Federated graph-level clustering network with personalized knowledge aggregation. In *International Conference on Intelligent Computing*, pp. 27–38. Springer, 2025.

Junlong Wu, Renda Han, Wenxuan Tu, Jingxin Liu, Haotian Wang, and Jieren Cheng. Fedcnd: Federated graph-level clustering under inter-client cluster number discrepancy. In *Proceedings of the InternationalWorld Wide Web Conference*. WWW, 2026.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. 34: 18839–18852, 2021.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pp. 478–487. PMLR, 2016.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *arXiv preprint arXiv:1810.00826*, 2018.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. 34:6671–6682, 2021.

Bowen Zhao, Qianqian Wang, Zhengming Ding, and Quanxue Gao. Attribute-missing multi-view graph clustering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25832–25841, 2025.

Jiankai Zuo, Yu Zhang, Yang Zhang, Zihao Yao, and Yaying Zhang. Glad: Bidirectional structure-attribute alignment via latent graph diffusion models. In *Forty-third International Conference on Machine Learning*, 2026.

A Related Work

A.1 Federated Graph-level Clustering

FGC aims to collaboratively group graphs across multiple clients without sharing raw graph data, thereby addressing privacy concerns while handling non-IID graph distributions. Existing methods mainly focus on learning global consensus representations or personalized knowledge aggregation to improve clustering performance. For instance, FedGCN (Liu et al., 2025a) generates global prototypes from multi-source clustering structures, which in turn guide client models in unsupervised settings. Building on this idea of guiding client models, FedPKA (Wu et al., 2025) incorporates prior confidence as a mechanism to mitigate knowledge drift and heterogeneity in non-IID graph data. Similarly addressing client discrepancies, FGCN-AKC (Han et al., 2026) proposes local knowledge enhancement and global prototype alignment to further reduce personalized knowledge discrepancies among clients. Analogously, PERFECT (Liu et al., 2026) leverages personalized aggregation of representative samples and clustering-wise gradient updates, which effectively enhances domain-specific cluster recognition. Moreover, FedCND (Wu et al., 2026) enhances FGC by decoupling

Table 4: Basic notations for the proposed UFedGC.

Notations	Meaning	Notations	Meaning
M	Number of clients	C	Number of clusters
d'	Embedding dimension	I	Number of graphs held by client m
$\mathcal{G}^m$	Graph collection held by client m	G_i^m	i-th graph of client m
N	Number of nodes	d	Dimension of node features
$\mathbf{X}_i$	Node feature matrix	$\mathbf{A}_i$	Normalized adjacency matrix
$\mathbf{z}_i$	Local graph-level representation	$\mathbf{z}_i^s$	Shared graph representation
$\mathbf{z}_i^p$	Private graph representation	f_{loc}	Local GNN encoder
f_s	Shared GNN encoder	f_p	Private GNN encoder
$\boldsymbol{\theta}_{loc}$	Parameters of local encoder	$\boldsymbol{\theta}_m^s$	Shared GNN parameters of client m
$\boldsymbol{\theta}_m^p$	Private GNN parameters of client m	$\mathbf{P}_m^{glo}$	Global cluster prior for client m
$\mathbf{q}_i$	Soft cluster posterior assignment	$\mathrm{GNN}_{\mathrm{com}}$	Attribute completion network
$\mathbf{X}_i^{\mathrm{com}}$	Predicted missing attributes	$\mathbf{z}_i^{com}$	Re-encoded graph representation
$\hat{\mathbf{q}}i$	Re-estimated soft cluster posterior	$\mathbf{p}_c^{\mathrm{com}}$	Cluster prototype from completed graphs
$\mathcal{G}_{com}$	Completed graph set	$\mathbf{X}_i^{\mathrm{miss}}$	Masked node attribute matrix
$\hat{\mathbf{q}}_i^{(t-1)}$	Assignment from previous round $t-1$	$\mathcal{L}_{com}$	Total attribute completion loss
$\mathcal{L}_{rec}$	Reconstruction loss	$\mathcal{L}_{cc}$	Cluster consistency loss
u_i	Completion uncertainty of graph i	w_i	Reliability weight of graph i
S_m^c	Local cluster statistics for cluster c	$\boldsymbol{\mu}_m^c$	Prototype mean of local cluster c
$\boldsymbol{\Sigma}_m^c$	Prototype diagonal covariance	r_m^c	Effective sample size of local cluster c
u_m^c	Uncertainty of local cluster c	$\boldsymbol{\mu}_{glo}^c$	Mean of global cluster c
$\mathbf{D}_m$	Prototype matching-cost matrix	$\mathbf{T}_m^*$	Optimal OT cluster alignment matrix
a_m	Source transport mass vector	b	Target transport mass vector
$\hat{s}$	Aggregated transport evidence	$\hat{\boldsymbol{\mu}}_{glo}^c$	Updated mean of global cluster c
η_m	Reliability weight of client m	$\hat{\mathbf{P}}^{glo}$	Updated global prototype matrix

graph data into public and private patterns to align heterogeneous cluster distributions across clients while preserving privacy. Despite these advances, existing methods largely overlook the impact of missing attributes and completion uncertainty on FGC. In contrast, our work jointly performs uncertainty-aware attribute completion and graph-level clustering for reliable cross-client cluster alignment.

A.2 Graph Learning with Missing Attributes

Graph learning with missing attributes aims to recover meaningful node representations when a substantial fraction of feature vectors is unavailable, exploiting graph topology and partial attributes to compensate for the missing information. For example, GLAD (Zuo et al., 2026) proposes a latent graph diffusion framework with bidirectional alignment to capture the conditional distribution of missing attributes while preserving topological consistency. FSD-CAP (Qiao et al., 2026) instead combines fractional subgraph diffusion with class-aware propagation to progressively impute missing attributes while enforcing intra-class consistency. FISF (Um et al., 2025) takes a propagation-based approach, leveraging a two-stage propagate-and-inject mechanism that injects domain-specific feature variance into graph channels to accurately restore missing node attributes across complex topological structures. Although effective for centralized graph learning, these methods assume access to complete graph data and are therefore unsuitable for privacy-preserving federated settings. Applying them directly to FGC risks propagating imputation errors into graph representations and cluster assignments, further exacerbating cross-client cluster inconsistency.

B Notations

All notations used in the proposed UFedGC are summarized in Table 4.

C Algorithm

The overall process is detailed in Algorithm 1.

Algorithm 1 Training Procedure of UFedGC

input: Graph collections $\{\mathcal{G}^m\}_{m=1}^M$; node feature matrices $\{\mathbf{X}_i\}$; normalized adjacency matrices $\{\mathbf{A}_i\}$; number of clients M; number of clusters C; global communication rounds T; local training epochs E.
Output: Graph clustering assignments $\mathbf{R}$ across clients.
Initialize parameters $\boldsymbol{\theta}^{\text{glo}}$ at the server.
for $t = 1$ **to** T **do**
  `# Local Closed-Loop Attribute Completion and Clustering`
  for $m = 1$ **to** M **in parallel do**
    Receive global prior $\tilde{\mathbf{P}}_m^{\text{glo}}$ and update shared encoder f_s with $\boldsymbol{\theta}^s$.
    for epoch $= 1$ **to** E **do**
      Extract representations $\mathbf{z}_i, \mathbf{z}_i^s, \mathbf{z}_i^p$ and disentangle via Eq. (1).
      Infer soft cluster posterior $\mathbf{q}_i$ using prior $\tilde{\mathbf{P}}_m^{\text{glo}}$ via Eq. (2).
      Generate predicted attributes $\mathbf{X}_i^{\text{com}}$ using completion network g_ϕ via Eq. (3).
      Re-encode completed graph $\mathbf{z}_i^{\text{com}}$ and estimate posterior $\hat{\mathbf{q}}_i$ via Eq. (4).
      Optimize local parameters by minimizing loss $\mathcal{L}_{\text{com}}$ via Eq. (5).
    end for
    Estimate completion uncertainty u_i and reliability weights w_i via Eq. (6).
    Construct local cluster statistics $S_m^c = (\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c, r_m^c, u_m^c)$ for $c = 1, \ldots, C$.
    Upload shared parameters $\boldsymbol{\theta}_m^s$ and cluster statistics $\{S_m^c\}_{c=1}^C$ to server.
  end for
  `# Uncertainty-Aware Cross-Client Cluster Alignment`
  Construct matching-cost matrix $\mathbf{D}_m$ and mass vectors a_m, b via Eqs. (7) – (8).
  Solve unbalanced optimal transport for cross-client alignment matrix $\mathbf{T}_m^*$ via Eq. (9).
  `# Global Prior and Model Aggregation`
  Update global cluster priors $\hat{\mathbf{P}}^{\text{glo}}$ by moment matching via Eq. (10).
  Aggregate shared model parameters $\hat{\boldsymbol{\theta}}^s$ weighted by client reliability η_m via Eq. (11).
  Send $\tilde{\mathbf{P}}^{\text{glo}}$ and $\tilde{\boldsymbol{\theta}}^s$ back to clients.
end for
Obtain $\mathbf{R}$ through clustering.
RETURN R

D Benchmark Datasets and non-IID Setting

The dataset statistics are summarized in Table 5. The non-IID settings are summarized in Table 6. A detailed introduction to the dataset follows:

- **MUTAG** The MUTAG consists of 188 molecular graphs representing nitroaromatic compounds, where each node corresponds to an atom and edges represent chemical bonds. The ground-truth label classifies each compound as mutagenic or non-mutagenic for Salmonella typhimurium.
- **BZR_MD** The BZR_MD dataset is a graph dataset containing 306 molecules, where each graph represents a chemical compound. The nodes in each graph correspond to atoms, and the edges represent chemical bonds between them. The dataset is used to evaluate graph-based machine learning models, particularly for tasks like molecular property prediction. It contains binary labels, making it suitable for binary clustering tasks. The BZR_MD dataset is often used to test the ability of models to capture chemical structure information in molecular graphs.
- **PTC_MR** The PTC_MR dataset comprises 344 molecular graphs of dihydrofolate reductase inhibitors, each labeled according to its binding activity, serving as an indicator of potential antimicrobial or anticancer action.
- **BZR** The BZR dataset contains 405 molecular graphs of benzodiazepine receptor ligands. Each graph is labeled according to its biological activity, indicating whether the compound binds to the receptor.

Table 5: Description of benchmark datasets.

	Basic Attributes				
Datasets	**Domain**	**Classes**	**Graphs**	**Average Nodes**	**Average Edges**
MUTAG	SM	2	188	17.93	19.79
BZR_MD		2	306	21.30	225.06
PTC_MR		2	344	14.29	14.69
BZR		2	405	35.75	38.36
COX2		2	467	41.22	43.45
DHFR		2	756	42.43	44.54
AIDS		2	2000	15.69	16.20
DD	BIO	2	1178	284.32	715.66
PROTEINS		2	1113	39.06	72.82
SYNTHETIC	SY	2	300	100.00	196.00
COLORS-3		11	10500	61.31	91.03
REDDIT-MULTI-12K	SN	11	11929	391.41	456.89
COLLAB		3	5000	74.49	2457.78
IMDB-MULTI		3	1500	13.00	65.94
Letter-high	CV	15	2250	4.67	4.50
Letter-low		15	2250	4.68	3.13
Letter-med		15	2250	4.67	3.21

Table 6: Description of the non-IID settings.

	non-IID Setting					
Datasets	**SM**	**SN**	**SM-BIO**	**SM-BIO-SY**	**SN-SY**	**CV**
MUTAG	✓	-	✓	✓	-	-
BZR_MD	✓	-	✓	✓	-	-
PTC_MR	✓	-	✓	✓	-	-
BZR	✓	-	✓	✓	-	-
COX2	✓	-	✓	✓	-	-
DHFR	✓	-	✓	✓	-	-
AIDS	✓	-	✓	✓	-	-
DD	–	-	✓	✓	-	-
PROTEINS	-	-	✓	✓	-	-
SYNTHETIC	-	-	-	✓	-	-
COLORS-3	-	-	-	-	✓	-
REDDIT-MULTI-12K	-	-	-	-	✓	-
COLLAB	-	✓	-	-	-	-
IMDB-MULTI	-	✓	-	-	-	-
Letter-high	-	-	-	-	-	✓
Letter-low	-	-	-	-	-	✓
Letter-med	-	-	-	-	-	✓

- **COX2** The COX2 dataset includes 467 drug-like molecules annotated based on their inhibitory activity toward the cyclooxygenase-2 enzyme. The binary ground-truth distinguishes inhibitors from non-inhibitors.
- **DHFR** The DHFR dataset comprises 756 molecular graphs of dihydrofolate reductase inhibitors, each labeled according to its binding activity, serving as an indicator of potential antimicrobial or anticancer action.
- **AIDS** The AIDS dataset encompasses 2,000 molecular graphs labeled by their anti-HIV activity. Each graph is classified as either active or inactive in inhibiting HIV replication.

- **DD** The DD dataset contains 1,178 protein structure graphs, where labels denote enzyme or non-enzyme status, reflecting fundamental functional properties.
- **PROTEINS** The PROTEINS dataset includes 1,113 labeled protein graphs, with binary ground truth indicating enzyme function, widely used for GNN-based protein classification.
- **SYNTHETIC** The SYNTHETIC dataset is a collection of 300 artificially generated graphs, half of which contain structural anomalies. The ground truth labels differentiate normal from anomalous instances.
- **COLORS-3** The COLORS-3 dataset includes 10,500 graphs, each with node features that represent colors. The graphs are designed to evaluate the performance of graph-based learning models, especially in multi-class classification tasks. The dataset is often used in graph classification experiments to test algorithms' ability to capture structural and feature-based patterns in graphs.
- **IMDB-MULTI** The IMDB-MULTI dataset comprises 1,500 actor co-appearance networks. Labels distinguish between three movie genres, with graphs representing actor collaboration patterns.
- **COLLAB** The COLLAB dataset comprises 5,000 scientific collaboration networks. Labels distinguish between three research fields, with graphs representing collaborations among researchers.
- **REDDIT-MULTI-12K** The REDDIT-MULTI-12K dataset is a large-scale graph dataset, consisting of 12,000 social network graphs derived from Reddit. Each graph represents a community or discussion thread, where nodes correspond to posts, and edges represent interactions (such as replies or mentions) between them. The dataset is multi-class, with each graph labeled by one of 11 different categories. It is commonly used to evaluate graph-based models, especially in tasks like community detection and classification in social network analysis. The size and complexity of the dataset make it particularly useful for testing the scalability and effectiveness of graph neural networks (GNNs).
- **Letter-low/med/high** The Letter-low/med/high datasets are subsets of 2,250 character-rendered graphs per group, each labeled with one of 15 alphabet letters. The three versions vary by distortion level (high, medium, low), enabling study of structural robustness under different noise conditions.

E Detailed Introduction to Baseline Methods

It is worth noting that in the comparative experiments of Table 1, the procedures for adapting supervised FGL methods into unsupervised FGC methods, converting unsupervised FGL (federated anomaly detection) methods into unsupervised FGC methods, and applying a small amount of labels to the converted unsupervised FGC methods in Fig. 3, all follow FedGCN (Liu et al., 2025a) and FGCN-AKC (Han et al., 2026). The various baseline methods are described in detail below:

- **FedGCN.** FedGCN is an unsupervised federated graph-level clustering framework that extracts topology-oriented graph representations and preserves cluster prototypes at the client side. The server aggregates these multi-source prototypes through Gaussian estimation to construct global consensus representations, which are then returned to the clients to facilitate local clustering.
- **FGCN-AKC.** FGCN-AKC addresses consensus failure caused by personalized knowledge discrepancies across clients through adaptive knowledge compensation. It introduces a Local Knowledge Enhancement strategy to extract and refine reliable clustering knowledge, together with a Global Prototype Alignment mechanism that forms semantically related client communities for adaptive knowledge aggregation.
- **FedCND.** FedCND focuses on federated graph-level clustering under inter-client discrepancies in the number of clusters, relaxing the common assumption that all clients share the same cluster cardinality. It decouples graph information into public and private patterns to align heterogeneous knowledge across clients and alleviate the misalignment of cluster-specific information during federated aggregation.

- **PERFECT.** PERFECT is a personalized federated graph-level clustering framework designed to mitigate structural heterogeneity across clients. It reconstructs privacy-preserving representative samples for inter-client correlation modeling, derives a personalized aggregation strategy at the server, and further employs clustering-wise gradients to selectively refine the returned client models.
- **FedPKA.** FedPKA is an unsupervised federated graph-level clustering framework that performs personalized knowledge aggregation under non-IID graph distributions and knowledge drift. It evaluates the confidence of local domain knowledge, constructs a dynamic community-aware aggregation strategy for each client, and further introduces adaptive prototype adjustment to improve prototype alignment.
- **FedSage.** FedSage is a subgraph federated learning method that trains a GraphSAGE model under the FedAvg framework to collaboratively exploit node features, graph structures, and task labels from distributed subgraphs. In our experiments, its federated representation learning mechanism is adapted from node-level classification to graph-level clustering to provide a representative baseline for conventional FGL.
- **GCFL.** GCFL tackles non-IID federated graph learning by dynamically clustering clients according to the gradients of their local GNNs, allowing clients with similar learning dynamics to share model parameters. Its enhanced variant, GCFL+, further exploits gradient sequences with dynamic time warping to obtain more stable client grouping, and the same client-clustering principle is adapted here for graph-level clustering.
- **FedStar.** FedStar addresses non-IID graph learning by explicitly separating structure-related and feature-related knowledge. It employs an independent structure encoder whose parameters are shared across clients, while feature-based knowledge is learned in a personalized manner, thereby reducing feature misalignment and facilitating the transfer of domain-invariant structural information.
- **LG-FGAD.** LG-FGAD is a federated graph anomaly detection framework that combines self-adversarial graph generation with local-global anomaly awareness. It further adopts dual knowledge distillation over both logits and embedding distributions, enabling collaborative learning while preserving client-specific characteristics and alleviating the impact of non-IID data.
- **FGAD.** FGAD introduces an anomaly generator that perturbs normal graphs to construct pseudo-anomalous samples and trains a detector to distinguish them from normal graphs. A student model subsequently distills knowledge from the local teacher model to preserve personalization while reducing communication costs, and this federated learning mechanism is adapted to graph-level clustering in our experiments.
- **AGDiff.** AGDiff leverages a latent diffusion model to generate pseudo-anomalous graphs through subtle perturbations of normal graph representations. By jointly training a classifier to distinguish generated anomalies from normal graphs, it learns more discriminative graph representations without relying on abundant real anomalous samples, and its representation learning framework is adapted to the clustering setting.
- **FedGCN + GLAD.** GLAD is a centralized graph attribute completion method based on latent diffusion and bidirectional structure-attribute alignment. It reconstructs missing node attributes by conditioning latent diffusion on graph structure while enforcing structural consistency, and the completed graphs are subsequently processed by FedGCN as the federated clustering backbone.
- **FedGCN + FSD-CAP.** FSD-CAP is a two-stage graph feature imputation framework designed for highly incomplete node attributes. It first localizes diffusion using graph-distance-guided subgraph expansion and a fractional diffusion operator, and then refines the imputed features through class-aware propagation; the resulting completed attributes are used as input to FedGCN.
- **FedGCN + FISF.** FISF revisits propagation-based graph feature imputation and identifies the low-variance channel problem caused by feature propagation under missing attributes. It injects synthetic features to increase the variance and informativeness of the completed channels, after which the imputed graphs are fed into FedGCN for federated graph-level clustering.

F More Implementation Details

All experiments are implemented in PyTorch 2.4.0 and executed on an NVIDIA RTX 4090 GPU with 24 GB of memory. The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, a batch size of 256, and a weight decay of 1×10^{-3}. Each training run lasts for 10 epochs based on the validation loss. We employ a 3-layer GIN backbone with hidden dimensions set to 10. ReLU activation is applied after each layer. All input features are normalized to zero mean and unit variance. Self-loops are added to the adjacency matrices before propagation. For fair comparison, all baseline methods are either reproduced using publicly available implementations or directly adapted from PyG libraries. Hyperparameters for each method are refined. Our experiments are conducted on Windows 10 with CUDA 11.8. The reported results are averaged over five independent runs with different random seeds.

G Evaluation Metrics

We evaluate clustering quality with ACC, NMI, ARI, and macro-F1. Labels are never used for training and are used only for post-hoc evaluation.

- **Clustering Accuracy (ACC).** ACC is computed after optimal one-to-one matching between predicted clusters and ground-truth classes:

$$\mathrm{ACC} = \max_{\pi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(y_i = \pi(\hat{y}_i)\right), \tag{12}$$

  where π is the optimal cluster-to-label permutation.

- **Normalized Mutual Information (NMI).** NMI measures the normalized mutual dependence between the predicted partition $\hat{Y}$ and ground-truth labels Y:

$$\mathrm{NMI} = \frac{I(Y, \hat{Y})}{\sqrt{H(Y)H(\hat{Y})}}. \tag{13}$$

- **Adjusted Rand Index (ARI).** ARI evaluates pairwise agreement between two partitions while correcting for chance:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}. \tag{14}$$

- **Macro-F1.** Macro-F1 averages class-wise F1 scores after the same optimal label matching used for ACC:

$$\mathrm{F1} = \frac{1}{C} \sum_{c=1}^{C} \frac{2\,\mathrm{Precision}_c\,\mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}. \tag{15}$$

For federated settings with multiple clients, we report weighted averages across client datasets together with the standard deviation over repeated seeds.