ONE-FOR-ALL GRAPH FOUNDATION MODEL FOR FEDERATED GRAPH-LEVEL CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph-level clustering aims to discover meaningful pattern groups from decentralized graph data. However, most existing methods require training from scratch for each new non-IID setting and cannot rapidly generalize to unseen domains, resulting in substantial computational and communication overhead. To this end, we propose GFM-FGC, a One-for-All Graph Foundation Model for Federated Graph-Level Clustering, following a *pre-train once, adapt locally to each domain, and share knowledge once* paradigm. GFM-FGC is first pre-trained to acquire generalizable graph prior knowledge from public graph data. When deployed in a new non-IID setting, the pre-trained model is distributed to clients and efficiently adapted to their private local graphs through lightweight fine-tuning. The locally adapted knowledge is then shared through a single aggregation round to produce a globally coordinated clustering model. In this way, GFM-FGC can rapidly generalize to unseen non-IID settings without repeated end-to-end training or multiple-round client-server communication. Extensive experiments across multiple non-IID settings demonstrate that GFM-FGC achieves competitive clustering performance while providing improved reusability and efficiency over methods trained separately for each scenario.

1 INTRODUCTION

Graph-structured data (Liang et al., 2023; Liu et al., 2025e; Liao et al., 2025b; Liu et al., 2024b; Wang et al., 2026c) are widely used to represent complex interactions in social networks (Zhang, 2025), anomaly detection (Cai et al., 2025b;a; Qiao et al., 2025), recommendation systems (Li et al., 2025), and biological systems (Su et al., 2025). Deep Graph Clustering (DGC) (Liu et al., 2025c;d; Bo et al., 2020; Cai et al., 2024a;b) as a key technology in graph data mining, leverages representation learning to capture expressive features. DGC divides samples based on these features into latent pattern groups from graph data without requiring label supervision. In many practical applications, however, graph data are distributed across multiple organizations or devices and cannot be centralized because of privacy concerns, regulatory restrictions, and data-ownership constraints. Federated Graph Learning (FGL) (Fu et al., 2026; Liao et al., 2025a) enables clients to collaboratively train local model while protecting local graph data, thus providing a promising solution for distributed graph data mining.

Federated Graph Clustering (FGC), as an unsupervised FGL task, is a representative paradigm worthy of attention in recent years. These methods include node-level (Liu et al., 2025b; 2026b) and graph-level (Han et al., 2025; 2026; Liu et al., 2025b; Wang et al., 2026b; Liu et al., 2026a) tasks. Despite recent progress, existing FGC (Liu et al., 2025b) methods are generally designed for a single, fixed non-IID setting and lack the ability to serve unseen non-IID distributions rapidly. For every new federation setting, they typically train a domain-specific model from scratch through repeated local optimization and server–client communication. Such a workflow requires not only many communication rounds, but also sufficient graph data and globally coordinated training.

Table 1: Requirement comparison with existing FedGFMs. **Label**, **Clustering**, and **PLM/VLM** denote whether the method needs label supervision, is designed for clustering, and needs language models and vision-language models to support, respectively.

Method	Label	Clustering	PLM/VLM
FedGFM+ (Zhu et al., 2026a)	✓	×	✓
FedBook (Wu et al., 2025)	✓	×	✓
FedGALA (Zhu et al., 2026b)	✓	×	✓
FedGAMMA (Li et al., 2026)	✓	×	✓
GFM-FGC (Ours)	×	✓	×

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.

Consequently, knowledge acquired in previous scenarios is discarded rather than reused, even when it contains broadly transferable graph patterns. The resulting repeated computation and communication substantially hinder the efficient deployment of FGC across multiple applications.

Graph Foundation Models (GFMs) offer a promising direction for overcoming this limitation, as they aim to capture transferable graph knowledge that can be reused across different datasets and domain. Some recent studies, e.g., FedGFM+ (Zhu et al., 2026a), FedBook (Wu et al., 2025), FedGALA (Zhu et al., 2026b), and FedGAMMA (Li et al., 2026), have explored the combination of GFM and FGL. However, directly adapting these methods to the clustering task remains non-trivial due to three key limitations, as shown in Table 1. First, their downstream adaptation or fine-tuning procedures often rely on label supervision, which conflicts with the inherently unsupervised nature of graph clustering. Second, some of them are designed primarily for text-attributed graphs or depend on PLM/VLM models to provide semantic signal, making them less adaptable to general attributed graphs where textual information is unavailable. Third, their pre-training objectives mainly focus on learning general node-level representations rather than graph-level, clustering-oriented knowledge, and therefore do not explicitly preserve or enhance the latent patterns required for effective clustering. (See **Appendix** A for related work).

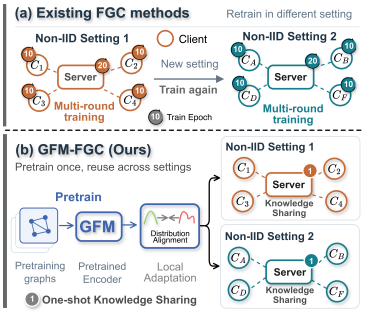

Figure 1: Compared with ours, existing FGC methods require training with repeated communication for each non-IID setting.

To address these challenges, we propose GFM-FGC, a GFM for FGC based on a pre-train, use-multiple-times paradigm. The core idea is to pretrain a GFM and reuse its transferable knowledge across diverse non-IID settings through lightweight local adaptation, as shown in Fig. 1. Specifically, in the pre-training stage, GFM-FGC filters into cluster-oriented graphs to learn transferable graph knowledge in a generative-adversarial manner. In the federated learning stage, the server distributes the resulting GFM to local clients to adapt and fine-tune the model on their private graphs locally. Finally, we execute a one-shot federated knowledge sharing to coordinate clustering-friendly information across clients. In this way, the expensive learning process is amortized over repeated deployments, allowing previously learned knowledge to benefit new domains rapidly. The main contributions of this work are summarized as follows:

- **New Perspective** To the best of our knowledge, this is the first attempt to explore *train-once, use-multiple* paradigm for unsupervised FGC. Compared with supervised methods, the absence of labels makes cross-domain knowledge alignment particularly challenging.
- **New Method** We propose GFM-FGC, a unified framework that decouples unsupervised FGL from non-IID settings, avoiding repeated end-to-end training for each new environment. First, we pretrain a GFM to learn transferable graph knowledge. Second, we perform lightweight local adaptation to accommodate each client's private graph. Finally, we aggregate the local knowledge only once to achieve global consensus.
- **Excellent Performance** Extensive experiments with state-of-the-art methods under various conditions show that GFM-FGC provides competitive performance and superior efficiency.

2 Methodology

In this section, we introduce **GFM-FGC** in detail. Its core insight is to train a generalized graph foundation model and then rapidly adapt to various non-IID settings through lightweight fine-tuning and federated sharing. It contains three components: Causal-Structural Adversarial Pretraining, Local Knowledge Adaptation, and One-shot Knowledge Sharing and Feedback. The overall architecture is shown in Fig. 2. The training process is shown in Algorithm 1 of **Appendix**.

2.1 Notation and Problem Formulation

We assume that the server has access to a public pretraining dataset $\mathcal{D}^{\text{pre}} = \{\mathrm{G}_s\}_{s=1}^{N_G}$, while N_M clients hold private graph datasets $\mathcal{D}^{L} = \{\mathrm{G}_n\}_{n=1}^{N_G}, \mathrm{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{A})$ with N_C clusters, where

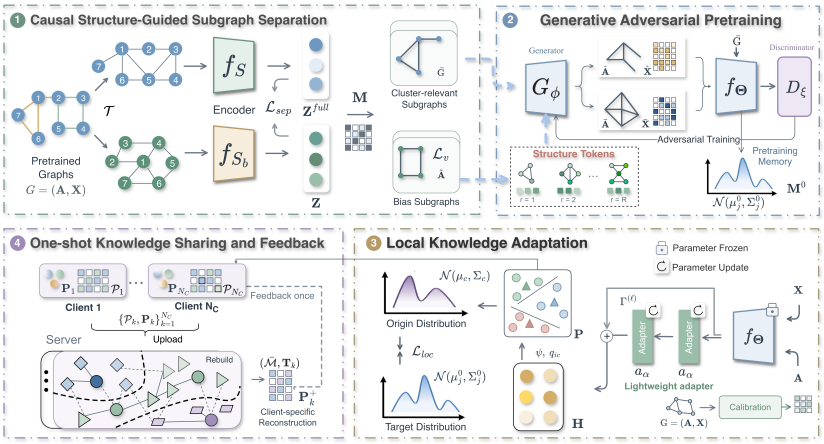

Figure 2: The architecture of GFM-FGC. Firstly, the GFM learns transferable clustering-oriented graph knowledge. The server then distributes the pretrained GFM to clients for local adaptation and prototype extraction. Finally, the server performs one-shot knowledge aggregation to discover globally shared factors, which are returned to clients to refine local clustering results.

$\boldsymbol{X} \in \mathbb{R}^{N_N \times N_d}$ and $\boldsymbol{A} \in \{0,1\}^{N_N \times N_N}$ are node attribute and graph structure matrices. Usually, each graph has N_N nodes and N_E edges, respectively. The goal of GFM-FGC is to learn a GFM for FGC, and the clients fine-tune it, then share knowledge through a single round of global communication to produce effective clustering results.

2.2 Generative Adversarial Pretraining of Graph Foundation Models

Learning graph knowledge directly from a limited set of observed graphs may cause the encoder to limit its structural coverage and weaken its ability to generalize to unseen graphs. We therefore use causally stable structures to guide the construction of diverse pretraining samples and provide a reliable guidance signal for representation learning.

Cluster-oriented and Biased Subgraph Separation. In this stage, our goal is to identify a compact cluster-oriented subgraph that contains the key pattern evidence required to determine the latent partition, while separating graph components that are irrelevant. We call this cluster-oriented component the causal subgraph and the remaining component the biased subgraph. We first utilize a graph projector f_S and learnable cluster prototypes $\mathcal{C} = \{\boldsymbol{c}_k\}_{k=1}^{N_C}$ to encode the all graphs and compute a prototype reference assignment for each graph $\mathcal{G}$: $\boldsymbol{H} = f_S(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{Z} = \psi(\boldsymbol{H}), q_{ik} = \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{c}_k)/\tau_t)}{\sum_{\ell=1}^{N_C} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{c}_\ell)/\tau_t)}$, where $\mathrm{sim}(\cdot,\cdot)$, $\psi(\cdot)$, and τ_t denote cosine similarity, readout function and sharpness rate of the cluster assignment distribution $\boldsymbol{q}_i$, respectively. $\boldsymbol{q}_i$ inferred from the embeddings of complete graphs to evaluate whether a retained subgraph preserves the same partition.

Based on this paradigm, we independently sample two perturbations $(\boldsymbol{X}^a, \boldsymbol{A}^a) = \mathcal{T}_a(\boldsymbol{X}, \boldsymbol{A})$, where $a \in \{1,2\}$ indexes two independently augmented views of the same graph. The edge gate produces a mask $\boldsymbol{M}^a$ for each perturbed view, which decomposes its topology into

$$\bar{\boldsymbol{A}}^a = \boldsymbol{A}^a \odot \boldsymbol{M}^a, \quad \tilde{\boldsymbol{A}}^a = \boldsymbol{A}^a \odot (\boldsymbol{1} - \boldsymbol{M}^a), \tag{1}$$

where $\bar{\boldsymbol{A}}^a$ and $\tilde{\boldsymbol{A}}^a$ denote the retained cluster-defining topology and its complementary topology, respectively. The corresponding graph-level representations are obtained as $\boldsymbol{z}^{\mathrm{full},a} = \psi(f_S(\boldsymbol{X}^a, \boldsymbol{A}^a)), \bar{\boldsymbol{z}}^a = \psi(f_S(\boldsymbol{X}^a, \bar{\boldsymbol{A}}^a)), \tilde{\boldsymbol{z}}^a = \psi(f_{S_b}(\boldsymbol{X}^a, \tilde{\boldsymbol{A}}^a))$. Their soft cluster assignments are $\boldsymbol{q}^{\mathrm{full},a}$, $\bar{\boldsymbol{q}}^a$, and $\tilde{\boldsymbol{q}}^a$, respectively. Moreover, let $\bar{\boldsymbol{Z}}^a = [\bar{\boldsymbol{z}}_1^a, \ldots, \bar{\boldsymbol{z}}_{N_G}^a]^\top$ and $\tilde{\boldsymbol{Z}}^a = [\tilde{\boldsymbol{z}}_1^a, \ldots, \tilde{\boldsymbol{z}}_{N_G}^a]^\top$ denote the graph-level representation matrices. The separator is optimized by

$$\begin{aligned}\mathcal{L}_{\mathrm{sep}} = &\frac{1}{2N_G}\sum_{a=1}^{2}\sum_{i=1}^{N_G}\mathrm{JS}(\bar{\boldsymbol{q}}_i^a, \mathrm{sg}[\boldsymbol{q}_i^{\mathrm{full},a}]) + \lambda_{\mathrm{inv}}\frac{1}{N_G}\sum_{i=1}^{N_G}\mathrm{JS}(\bar{\boldsymbol{q}}_i^1, \bar{\boldsymbol{q}}_i^2) - \lambda_b\frac{1}{2N_G}\sum_{a=1}^{2}\sum_{i=1}^{N_G}\mathcal{H}(\tilde{\boldsymbol{q}}_i^a) \\ &+ \frac{\lambda_h}{2}\sum_{a=1}^{2}\mathrm{HSIC}(\bar{\boldsymbol{Z}}^a, \tilde{\boldsymbol{Z}}^a) + \frac{\lambda_\rho}{2N_G}\sum_{a=1}^{2}\sum_{i=1}^{N_G}(|\mathcal{E}_i^a|^{-1}\sum_{(u,v)\in\mathcal{E}_i^a} m_{i,uv}^a - \rho)^2.\end{aligned} \tag{2}$$

The first term requires the retained subgraph to recover the reference partition inferred from the corresponding full-graph view, thus enforcing cluster sufficiency. The second term encourages consistent cluster assignments across two independent interventions. The third term suppresses confident cluster predictions from the complementary topology, while the HSIC regularizer reduces the statistical dependence between the retained and complementary graph representations. Finally, the edge-ratio constraint controls the size of the retained topology and prevents the trivial solution of assigning every edge to the cluster-defining subgraph.

After training, the learned mask $\boldsymbol{M}$ induces a soft decomposition of the original graph. The deterministic retention probability $p_{ij} = \sigma(g_\omega(a_{ij}))$ quantifies the contribution of edge (i, j) to the cluster-oriented subgraphs. The soft probabilities are retained during training to enable differentiable optimization. The resulting causal subgraph $\bar{\mathrm{G}} = (\bar{\boldsymbol{A}}, \bar{\boldsymbol{X}}, \bar{\boldsymbol{H}})$ preserves the structural relations that consistently support the latent partition, whereas the biased branch absorbs environment-specific correlations and unstable shortcuts. Consequently, $\bar{\mathrm{G}}$ provide both a high-confidence graph cluster-oriented prior for the subsequent topology-conditioned graph generation.

Structure-conditioned Subgraph Generation and Discrimination In this stage, our goal is to utilize $\bar{\mathrm{G}}$ and $\tilde{\mathrm{G}}$ to construct a structurally synthetic subgraph set in adversarial manner. The generator provides diverse yet causally consistent training instances for the subsequent discrimination stage. This preserves fine-grained local patterns across different neighborhood scales and broadens the structural support of the pretraining distribution. Specifically, we extract classic structural patterns to create reusable token vocabularies from $\tilde{\mathrm{G}}$. These patterns are then compositionally injected into $\bar{\mathrm{G}}$, expanding the pattern support of the pretraining distribution. For each node v and radius r, we extract its r-hop subgraph $\tilde{\mathrm{G}}_v^{(r)} = \tilde{\mathrm{G}}[\mathcal{N}_r(v)]$ from the bias subgraph and encode it as

$$\boldsymbol{u}_v^{(r)} = \psi(f_S(\tilde{\mathrm{G}}_v^{(r)})), \quad k_v^{(r)} = \arg\min_{m\in[M_r]} \|\boldsymbol{u}_v^{(r)} - \boldsymbol{e}_m^{(r)}\|_2^2, \tag{3}$$

where $\mathscr{E}^{(r)} = \{\boldsymbol{e}_m^{(r)}\}_{m=1}^{M_r}$ is the structural vocabulary at scale r. The structural token assigned to the corresponding subgraph is defined as $\boldsymbol{e}^{(r)}$. The vocabularies are learned by

$$\mathcal{L}_{\mathrm{v}} = \sum\nolimits_{r=1}^{R} [\|\mathrm{sg}[\boldsymbol{U}_r] - \boldsymbol{Q}_r\|_F^2 + \beta\|\boldsymbol{U}_r - \mathrm{sg}[\boldsymbol{Q}_r]\|_F^2 + \lambda_o\|\tilde{\boldsymbol{E}}_r\tilde{\boldsymbol{E}}_r^\top - \boldsymbol{I}_{M_r}\|_F^2] + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}}. \tag{4}$$

where $\boldsymbol{U}_r$ stacks the node representations at scale r, and $\tilde{\boldsymbol{E}}_r$ denotes the normalized corresponding codebook containing M_r codewords. $\boldsymbol{S}_r$ is the one-hot assignment matrix. Accordingly, $\boldsymbol{Q}_r = \boldsymbol{S}_r\boldsymbol{E}_r$ collects the quantized representations of all nodes. $\boldsymbol{I}_{M_r}$ is an identity matrix, and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. The coefficients β, λ_o, and λ_{rec} control the commitment, codeword orthogonality, and reconstruction terms, respectively. The first two terms update the vocabulary entries and encourage the encoder outputs to commit to their assigned tokens, respectively. The diversity regularizer discourages different entries from encoding redundant patterns, $\mathcal{L}_{\mathrm{rec}}$ preserves the local connectivity of each bias subgraphs from its quantized representation. Consequently, each token represents a recurring bias pattern at a particular neighborhood scale.

We then utilize these structure tokens to guide graph generation. For each causal subgraph, we independently sample one token from different scale vocabularies, as $\mathcal{T}_v = \{\boldsymbol{e}_{s_v^{(r)}}^{(r)}\}_{r=1}^{R}, s_v^{(r)} \sim p_r(\mathscr{E}^{(r)})$. The conditional generator produces a structural perturbation mask, as

$$\boldsymbol{M}_v^b = G_\phi(\boldsymbol{\epsilon}_v, \bar{\boldsymbol{H}}_v, \bar{\boldsymbol{z}}_v, \mathcal{T}_v), \quad \hat{\boldsymbol{A}}_v = \bar{\boldsymbol{A}}_v + (1 - \bar{\boldsymbol{A}}_v) \odot \boldsymbol{M}_v^b, \tag{5}$$

where $\boldsymbol{\epsilon}_v \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, while the node attributes and node correspondence are inherited from the causal subgraph, i.e., $\hat{\boldsymbol{X}}_v = \bar{\boldsymbol{X}}_v$. $\bar{\boldsymbol{z}}_v$ provides the cluster-relevant pattern context of the subgraph. The noise variable $\boldsymbol{\epsilon}_v$ introduces stochastic variation, allowing the generator to produce multiple structurally diverse subgraphs under the same basic patterns. The synthetic subgraphs generated across different graphs and center nodes are collected as additional graph instances for foundation-model pretraining, expanding the structural coverage of the pretraining distribution without directly reproducing the observed source graphs.

After graph synthesis, we pretrain the foundation encoder f_Θ with an auxiliary discriminator D_ξ in a generative-adversarial manner. D_ξ operates on the graph-level representations produced by the foundation encoder. D_ξ is attached to the output of f_Θ and is defined as

$$D_\xi(\bar{\mathrm{G}}, c) = d_\xi(\bar{\boldsymbol{Z}}, c), \quad D_\xi(\hat{\mathrm{G}}, c) = d_\xi(\hat{\boldsymbol{Z}}, c), \tag{6}$$

where c denotes the pseudo-cluster label of the original graph, and $d_\xi(\cdot, c)$ outputs a scalar discrimination score conditioned on c. In this way, the discriminator directly connects the subgraph separation and generation modules, encouraging generated samples to approximate the causal pretraining distribution while forcing the encoder to capture clustering-friendly information for pretraining.

The optimization objectives of G_ϕ and D_ξ are defined as

$$\mathcal{L}_D = \mathbb{E}_{(\hat{G},c)}[d_\xi(\hat{\boldsymbol{Z}}, c)] - \mathbb{E}_{(\bar{G},c)}[d_\xi(\bar{\boldsymbol{Z}}, c)] + \lambda_{\text{gp}}\mathcal{L}_{\text{gp}}, \quad \mathcal{L}_Q = \mathcal{L}_{\text{fid}} - \lambda_{\text{adv}}\mathbb{E}_{(\hat{G},c)}[d_\xi(\hat{\boldsymbol{Z}}, c)]. \tag{7}$$

To preserve the essential structure of the real subgraph, we define the fidelity loss as $\mathcal{L}_{\text{fid}} = \frac{1}{|\mathcal{V}|^2}\|\hat{\boldsymbol{A}} - \boldsymbol{A}^c\|_1$. The gradient penalty is imposed in the subgraph representation space: $\mathcal{L}_{\text{gp}} = \mathbb{E}_{\tilde{\boldsymbol{z}}}[(\|\nabla_{\tilde{\boldsymbol{z}}} d_\xi(\tilde{\boldsymbol{z}}, c)\|_2 - 1)^2], \tilde{\boldsymbol{z}} = \epsilon\bar{\boldsymbol{z}} + (1-\epsilon)\hat{\boldsymbol{z}}, \epsilon \sim \mathcal{U}(0,1)$. In the opposite direction, the foundation encoder is trained to resist these perturbations:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N_G}\sum_i \log \frac{\exp(\text{sim}(\bar{\boldsymbol{h}}_i, \hat{\boldsymbol{h}}_i)/\tau_t)}{\sum_j \exp(\text{sim}(\bar{\boldsymbol{h}}_i, \hat{\boldsymbol{h}}_j)/\tau_t)}, \quad \mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{sep}} + \lambda_c\mathcal{L}_{\text{v}} + \lambda_n\mathcal{L}_{\text{NCE}}. \tag{8}$$

We alternately minimize $\mathcal{L}_D$, $\mathcal{L}_Q$, and $\mathcal{L}_{pre}$. Through the resulting adversarial game, the generator progressively improves the structural fidelity of synthetic subgraphs, whereas the foundation encoder learns representations that are sensitive to relations yet robust across diverse graph structures.

2.3 Local Knowledge Adaptation

After pretraining, we fit a Gaussian mixture with C_0 components to the graph embeddings produced by f_{Θ^0}, forming a pretrained memory $\mathcal{M}^0 = \{(\boldsymbol{\mu}_j^0, \boldsymbol{\Sigma}_j^0)\}_{j=1}^{C_0}$. The server distributes Θ^0, $\mathcal{E}^{1:R}$, and $\mathcal{M}^0$ to all clients. Each client adapts the pretrained model using its local private graph set $\mathcal{D}^L$. Specifically, we keep the backbone encoder and $\mathcal{M}^0$ fixed and train the lightweight alignment layer. Moreover, we insert a residual adapter into each message-passing layer, as

$$\boldsymbol{H}^{(\ell+1)} = \text{MP}^{(\ell)}_{\boldsymbol{\Theta}^0}(\boldsymbol{H}^{(\ell)}, \boldsymbol{A}) + \boldsymbol{\Gamma}^{(\ell)} a^{(\ell)}_\alpha(\boldsymbol{H}^{(\ell)}), \tag{9}$$

where $a^{(\ell)}_\alpha$ is a client-specific adapter and $\boldsymbol{\Gamma}^{(\ell)} \in [0,1]$ controls its contribution to each node. We utilize graph-level embedding $\mathbf{Z}$ to learn N_C cluster centers and compute soft assignment distribution q_{ic} of graph i to cluster c by Student's t-distribution. We align distribution at the cluster level. Specifically, we assign each graph a weight $w_{ic} = a_i q_{ic}$, and summarize each local cluster using its assignment mass, embedding distribution, and structural pattern as

$$\begin{aligned} n_c &= \sum\nolimits_i w_{ic}, & \boldsymbol{\mu}_c &= \frac{1}{n_c}\sum\nolimits_i w_{ic}\boldsymbol{z}_i, \\ \boldsymbol{\Sigma}_c &= \frac{1}{n_c}\sum\nolimits_i w_{ic}(\boldsymbol{z}_i - \boldsymbol{\mu}_c)(\boldsymbol{z}_i - \boldsymbol{\mu}_c)^\top + \epsilon\boldsymbol{I}, & \boldsymbol{s}_c &= \frac{1}{n_c}\sum\nolimits_i w_{ic}\boldsymbol{r}_i. \end{aligned} \tag{10}$$

where $\boldsymbol{r}_i$ concatenates the soft assignments of graph i to the pretrained structural vocabulary. Thus, $(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ describes the cluster in the embedding space, whereas $\boldsymbol{s}_c$ characterizes its local topology. The optimization objective on each client is as

$$\mathcal{L}_{\text{loc}} = N^{-1}\sum\nolimits_i \mathcal{H}(\boldsymbol{q}_i) - \mathcal{H}(N^{-1}\sum\nolimits_i \boldsymbol{q}_i) + \mathcal{L}_{align} + \lambda_p\|\alpha\|_2^2. \tag{11}$$

where $\mathcal{L}_{align} = {N_C}^{-1}\sum_{c=1}^{N_C} \min_{1\le j\le C_0} W_2^2(\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \mathcal{N}(\boldsymbol{\mu}_j^0, \boldsymbol{\Sigma}_j^0))$. The mutual-information objective encourages confident assignments while preventing all graphs from collapsing into the same cluster. Together, these objectives adapt the model to each client.

2.4 One-shot Knowledge Sharing and Feedback

After local adaptation, each client uploads $\mathcal{P}_c = (n_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c, \boldsymbol{s}_c, r_c)$ as a describer of the cluster c. The reliability score is defined as $r_c = e^{-\bar{\mathcal{H}}_c}$, where $\bar{\mathcal{H}}_c$ is the assignment entropy within the cluster. To describe the relations of local clusters, we further construct a cluster relation matrix $\mathbf{R}$, as

$$r(c,c') = \eta_r \frac{\boldsymbol{\mu}_c^\top \boldsymbol{\mu}_{c'}}{\|\boldsymbol{\mu}_c\|_2\|\boldsymbol{\mu}_{c'}\|_2} + (1-\eta_r)\frac{\boldsymbol{s}_c^\top \boldsymbol{s}_{c'}}{\|\boldsymbol{s}_c\|_2\|\boldsymbol{s}_{c'}\|_2}. \tag{12}$$

Then we jointly infer the alignment and the global prototype memory $\bar{\mathcal{M}} = \{\bar{\mathcal{P}}_j\}_{j=1}^{N_C}$. For local cluster c and latent global slot j, we define $C(c,j) = W_2^2(\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \mathcal{N}(\overline{\boldsymbol{\mu}}_j, \overline{\boldsymbol{\Sigma}}_j)) + \beta_s \,\mathrm{JS}(\boldsymbol{s}_c, \overline{\boldsymbol{s}}_j)$. The transport plan $\boldsymbol{\Pi}_k$ and the barycenter are obtained by minimizing

$$\begin{aligned}\min_{\bar{\mathcal{M}},\bar{\boldsymbol{R}},\boldsymbol{b},\{\boldsymbol{\Pi}_k\}} \sum_{k=1}^{K} \big[\langle \boldsymbol{\Pi}_k, \boldsymbol{C}_k\rangle + \lambda_R \sum_{c,c',j,j'} |r_k(c,c') - \bar{r}(j,j')|^2 \,\Pi_{k,cj}\Pi_{k,c'j'} \\ + \varepsilon\Omega(\boldsymbol{\Pi}_k) + \tau_c \,\mathrm{KL}(\boldsymbol{\Pi}_k^\top \mathbf{1} \| \boldsymbol{b}) + \tau_r \,\mathrm{KL}(\boldsymbol{\Pi}_k \mathbf{1} \| \boldsymbol{a}_k)\big].\end{aligned} \tag{13}$$

where $\boldsymbol{a}_k$ is the normalized local cluster mass vector, with $a_{kc} = n_{kc} r_{kc} / \sum_{c'} n_{kc'} r_{kc'}$, and $\boldsymbol{b}$ is the learnable global mass vector. The first term aligns semantic Gaussian prototypes and structural-token distributions. The fused Gromov-Wasserstein term aligns the relation between prototypes and is therefore invariant to local cluster permutations.

The server returns $\bar{\mathcal{M}}$ and transport plan $T_k(c,j) = \Pi_{k,cj} / \sum_{j'} \Pi_{k,cj'}$ to client k. Its matching confidence is $\Gamma_{kc} = 1 - \frac{\mathcal{H}(\boldsymbol{T}_k(c,:))}{\log C}$. Each graph obtains a soft federated target and a confidence score, as

$$\tilde{y}_{k|ij} = \sum\nolimits_c q_{k|ic} T_k(c,j), \quad \Gamma_{ki} = \sum\nolimits_c q_{k|ic} \Gamma_{kc}. \tag{14}$$

Its global-prototype prediction is calculated as

$$p^g_{kij} \propto \exp\big[-2\tau_f{}^{-1} (\boldsymbol{z}_{ki} - \overline{\boldsymbol{\mu}}_j)^\top (\overline{\boldsymbol{\Sigma}}_j + \epsilon \boldsymbol{I})^{-1} (\boldsymbol{z}_{ki} - \overline{\boldsymbol{\mu}}_j) - \beta_f / \tau_f \mathrm{JS}(\boldsymbol{r}_{ki}, \overline{\boldsymbol{s}}_j)\big], \tag{15}$$

where $\boldsymbol{r}_{ki}$ be the structural-code distribution of graph i. The adapter and clustering head are refined with

$$\mathcal{L}^k_{\mathrm{proto}} = -\frac{1}{N_G} \sum\nolimits_{i,j} \Gamma_{ki} \tilde{y}_{kij} \log p^g_{kij}, \quad \mathcal{L}^k_{\mathrm{keep}} = \frac{1}{N_G} \sum\nolimits_i (1 - \Gamma_{ki}) \mathrm{JS}(\boldsymbol{q}^+_{ki}, \boldsymbol{q}^{\mathrm{before}}_{ki}). \tag{16}$$

The overall feedback loss is defined as: $\mathcal{L}^k_{\mathrm{fb}} = \mathcal{L}^k_{\mathrm{proto}} + \lambda_{kp} \mathcal{L}_{\mathrm{keep}}$. High-confidence alignments therefore transfer cross-client consensus, while low-confidence local assignments are regularized toward their pre-feedback values.

2.5 Efficiency Analysis

Pretraining. Let L, N_d, and T_{pre} denote the number of GNN layers, the hidden dimension, and the number of pretraining epochs, respectively. A graph-encoding operation costs $\mathcal{O}(L(N_E N_d + N_N N_d^2))$. Causal subgraph separation has the same order, while hierarchical tokenization requires $\mathcal{O}(N_N N_d R M)$ for codeword assignment. Graph generation and adversarial discrimination introduce another constant number of graph-encoding operations. Therefore, the overall pretraining complexity is $\mathcal{O}(T_{\mathrm{pre}} N_G [L(N_E N_d + N_N N_d^2) + N_N N_d \sum_{r=1}^{R} M_r])$.

Local Adaptation. Let T_{ad} denote the number of adaptation epochs. The frozen backbone requires $\mathcal{O}(L(N_E N_d + N_N N_d^2))$ for forward propagation, the adapters require $\mathcal{O}(L N_N N_d^2)$, and cluster assignment requires $\mathcal{O}(N_C N_d)$ per graph. Hence, the adaptation complexity is $\mathcal{O}(T_{\mathrm{ad}} N_G [L(N_E N_d + N_N N_d^2) + L N_N N_d^2 + N_C N_d])$.

3 Experiments

3.1 Experimental Setup

Benchmark Datasets and non-IID Settings. In settings of non-IID for federated graph-level learning Wang et al. (2026b), we use 16 public graph datasets from five domains: small molecules (MUTAG, BZR, COX2, DHFR, PTC-MR, AIDS and BZR-MD), bioinformatics (DD and PROTEINS), social networks (COLLAB and IMDB-MULTI), synthetic graphs (SYNTHETIC,SYNTHIE), and computer vision (LETTER-LOW, LETTER-MED and LETTER-HIGH). Each client holds a private dataset. The details of datasets and non-IID settings as summarized in **Appendix** B.1.

Baseline Methods. We compare GFM-FGC with four groups of baselines: 1) centralized end-to-end DGC methods, including DGLC (Cai et al., 2024b), DCGLC (Cai et al., 2024a) and GPC (Chen et al., 2025); 2) supervised FGL frameworks, including GCFL (Xie et al., 2021), FedStar (Tan et al., 2023), CFEGL (Liu et al., 2026c), LG-FGAD (Cai et al., 2024c); 3) unsupervised FGC

Table 2: Performance comparison across different FGL methods under six non-IID settings. * denotes supervised methods adapted for unsupervised learning. † denotes graph anomaly detection methods adapted for clustering. ‡ denotes centralized DGC methods adapted for FGL. Underlined values denote the best performance in each domain and metric.

Models	SM^2(7)				SN^3(2)				$SM\text{-}BIO^2$(9)			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
GCFL*	57.0±0.8	2.2±0.2	2.5±0.5	53.6±0.6	47.7±1.7	0.2±1.9	1.0±1.5	21.4±1.6	54.1±0.5	0.8±0.1	0.9±0.2	51.7±0.4
FedStar*	60.8±3.1	7.2±4.0	6.3±4.3	56.4±3.2	42.3±0.7	4.1±0.6	1.2±0.8	30.6±0.9	58.5±1.7	4.4±1.9	2.6±1.2	52.0±3.8
CEFGL*	61.9±3.9	3.8±4.0	4.6±4.2	53.4±8.3	45.2±4.0	3.1±1.7	2.3±2.1	34.0±7.2	56.6±0.5	1.9±1.1	1.5±0.2	49.1±4.3
LG-FGAD†	56.4±1.9	2.8±1.6	1.6±1.3	54.9±2.2	45.2±2.8	9.5±4.3	6.8±5.4	41.1±3.0	59.7±1.5	8.3±1.6	6.0±1.7	55.5±1.9
DGLC‡	66.3±1.2	3.6±3.3	6.2±4.9	49.4±4.4	51.1±1.9	7.3±4.7	3.4±3.1	28.2±3.8	64.8±1.6	1.6±1.3	2.2±1.9	44.0±1.8
DCGLC‡	65.5±3.2	4.2±5.1	0.6±5.6	44.1±5.3	48.3±0.6	1.0±1.4	2.1±1.5	21.4±1.2	63.6±1.6	4.0±6.7	5.1±8.3	43.5±5.1
GPC‡	63.9±0.1	4.7±0.3	8.4±0.2	52.5±0.4	47.1±1.4	0.2±0.1	0.0±1.6	21.5±0.2	67.6±1.8	0.1±1.9	-0.2±0.1	40.3±0.2
FedGCN	55.3±1.3	4.6±2.1	1.1±0.8	51.4±1.6	45.8±2.1	10.9±4.0	5.2±2.7	39.7±3.3	62.0±0.9	3.5±0.4	3.9±0.7	49.1±0.3
FGCN-DKS	55.6±0.2	7.0±0.1	0.2±0.3	51.7±0.4	42.7±1.3	3.8±3.1	4.1±0.2	29.9±1.9	65.6±1.0	1.9±0.9	1.5±1.1	47.0±2.4
FedAI	65.8±2.5	9.5±2.9	10.1±3.6	57.2±2.4	44.9±2.9	7.2±2.2	2.6±1.8	29.5±3.3	63.0±1.1	8.6±1.6	7.5±1.2	57.1±1.9
Perfect	60.9±0.1	1.4±2.6	2.5±1.7	52.3±0.2	43.1±1.2	7.2±4.3	3.3±0.8	36.5±2.5	64.6±0.4	1.0±1.1	0.5±0.9	46.2±2.7
FedCND	58.3±3.7	8.5±1.2	3.4±3.6	55.1±2.8	48.7±2.3	10.7±2.2	9.3±1.9	42.6±0.8	57.3±2.5	4.8±1.1	2.9±2.2	53.4±3.7
FedGFM	55.4±2.0	3.5±0.9	-1.6±1.1	46.2±2.2	OOM				56.2±1.9	2.7±1.1	0.8±0.5	51.0±1.6
FedGFM+	67.5±2.3	0.7±0.4	0.6±1.5	45.8±1.7	47.7±0.1	0.0±0.0	0.1±0.2	21.5±0.3	62.0±3.4	1.6±0.6	0.3±1.3	46.1±1.5
FedBook	61.7±0.8	0.2±0.2	0.0±0.5	31.8±0.3	47.7±1.5	1.3±0.9	0.8±1.2	21.4±0.3	67.6±0.1	0.2±0.3	0.3±0.6	40.5±0.7
FedGALA	59.8±2.3	3.6±2.7	4.2±2.4	54.5±2.9	41.1±0.9	8.3±2.2	4.4±2.3	40.5±3.4	61.7±1.3	8.5±1.4	7.2±1.8	59.0±0.9
Ours	**72.0±2.0**	**12.4±1.7**	**13.5±1.6**	**61.3±2.2**	**54.3±1.7**	**16.7±1.8**	**11.5±2.0**	**45.2±2.4**	**71.2±1.4**	**11.8±2.3**	**10.0±2.4**	**62.6±1.9**
	SM-BIO-SY^2(10)				**SN-SY^4(3)**				**CV^{15}(3)**			
GCFL*	65.5±0.0	10.6±0.1	10.7±0.2	55.8±0.3	37.2±0.4	0.1±0.0	0.3±0.1	34.4±0.5	9.1±3.9	4.7±1.2	3.6±0.8	6.9±2.5
CEFGL*	55.6±0.5	5.1±0.2	4.2±0.3	28.4±0.4	43.1±3.5	6.4±1.3	5.2±2.0	34.6±1.9	18.9±0.5	19.8±0.7	5.2±0.3	16.0±0.8
FedStar*	61.7±1.6	12.8±1.8	9.3±1.5	52.9±2.7	50.5±1.3	15.9±1.2	9.6±1.6	43.7±2.4	37.7±0.9	22.8±1.4	23.2±1.1	31.6±1.0
LG-FGAD†	OOM				39.1±1.3	7.7±1.9	1.9±0.0	29.3±2.4	35.6±1.3	41.7±1.9	18.0±1.4	30.2±1.5
DGLC‡	62.9±0.7	1.1±0.3	0.6±0.4	42.2±0.8	47.2±2.1	9.5±3.9	4.9±3.0	32.6±3.2	23.0±2.4	28.8±4.1	11.9±2.2	15.1±2.5
DCGLC‡	66.9±2.3	12.2±0.8	13.3±1.5	51.1±6.9	46.9±1.6	13.3±3.3	7.8±2.7	35.7±5.5	20.4±4.3	26.7±7.6	8.2±4.7	13.1±3.2
GPC‡	62.7±3.3	2.4±2.0	0.5±0.8	41.8±3.1	46.7±0.0	0.0±0.1	0.1±0.2	21.3±0.3	10.0±3.9	7.9±9.6	1.2±1.7	4.1±3.5
FedGCN	62.9±1.2	2.7±1.6	4.2±1.1	50.1±1.3	42.0±0.0	4.8±0.1	2.6±0.2	35.9±0.3	7.6±0.1	0.8±0.0	0.0±0.2	3.4±0.3
FGCN-DKS	58.2±1.0	12.9±0.7	10.1±0.6	55.3±0.8	23.5±1.5	13.4±1.0	8.7±1.6	15.6±1.2	36.2±1.3	36.1±1.5	21.5±1.2	34.3±1.4
FedAI	50.3±1.7	6.7±0.8	5.5±0.6	33.8±1.9	41.2±2.8	6.1±0.9	3.3±0.7	32.4±5.0	23.4±5.1	22.8±4.0	10.3±3.8	21.5±5.6
Perfect	60.3±0.5	6.9±2.7	4.7±2.8	52.8±2.4	44.3±3.7	3.8±1.5	1.2±2.0	29.9±5.3	35.9±1.4	35.2±0.7	21.0±1.2	33.1±1.3
FedCND	55.8±1.6	9.2±2.9	6.3±2.1	52.4±1.2	49.7±3.9	14.1±2.5	8.3±2.6	47.4±3.1	34.7±2.0	33.4±1.7	19.5±2.1	33.6±2.9
FedGFM	58.1±2.3	1.9±0.4	0.2±1.8	47.3±1.5	OOM				29.3±2.8	35.8±2.2	18.0±2.5	21.4±3.0
FedGFM+	59.3±2.7	1.2±0.8	0.4±1.6	41.9±2.5	46.8±0.2	0.2±0.3	0.1±0.1	21.5±0.4	12.3±0.0	0.0±0.0	0.0±0.0	11.3±0.9
FedBook	64.8±3.2	1.4±1.1	2.1±0.7	38.3±0.3	48.7±3.2	3.6±6.3	2.3±4.4	25.4±6.5	18.6±8.6	24.5±6.7	9.7±7.0	11.2±8.1
FedGALA	59.3±0.9	8.4±2.6	7.0±1.4	56.5±1.2	46.9±5.7	11.4±4.2	7.3±4.4	46.6±5.8	34.7±0.9	42.8±0.5	20.9±0.6	30.3±1.2
Ours	**70.5±2.3**	**14.8±1.9**	**16.9±1.6**	**62.4±1.8**	**56.2±1.5**	**18.7±3.4**	**11.9±1.8**	**51.3±2.3**	**41.4±2.0**	**47.5±1.6**	**30.7±1.7**	**39.6±2.1**

approach, including FedGCN, FedAI (Han et al., 2026), FGCN-DKS (Wang et al., 2026b), FedCND (Wu et al., 2026), and Perfect; and 4) federated GFMs, including FedGFM (Zhu et al., 2026a), FedGFM+, FedBook, and FedGALA. For a fair unsupervised comparison, we adapt the supervised FGL baselines to graph-level clustering without using graph labels during training.

Evaluation Metrics. We report clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and macro-F1. More details of the metrics are shown in **Appendix** B.3.

Implementation Details. All experiments are implemented using PyTorch 2.4.0 with CUDA 12.4. For supervised FGL methods and supervised federated GFMs, we replace label-dependent training objectives with an unsupervised clustering objective to adapt them to graph-level clustering. Further implementation details are provided in **Appendix** B.2.

3.2 Overall Clustering Performances

Table 2 demonstrates the clustering performance on six non-IID settings. GFM-FGC outperforms other categories of methods. This is mainly attributed to the following factors: (1) Compared with existing FGC methods, this advantage comes from its ability to learn and transfer knowledge across domains. Existing FGC methods are typically trained for specific non-IID settings and lack reusable knowledge across domains, thus requiring multiple rounds of training and communication when applied to new domains. (2) Comparison with supervised FGL methods adapted for unsupervised learning: although we adapt these baselines to an unsupervised setting, their original components are primarily designed for supervised tasks and cannot learn clustering-friendly guidance signals. Moreover, they also depend on specific non-IID settings. (3) Comparison with centralized DGC methods. the reason is that they primarily learn representations and cluster structures from the

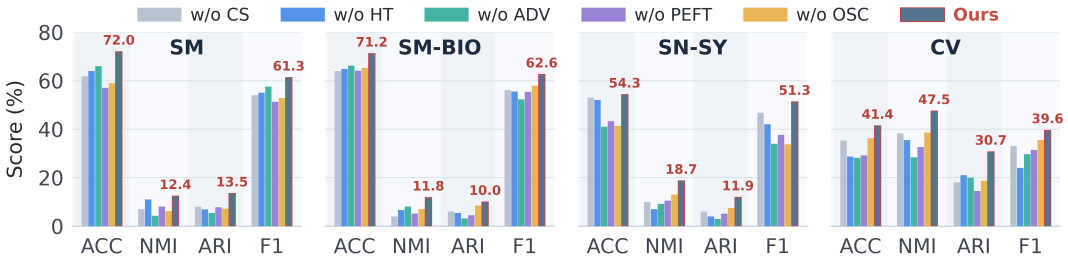

Figure 3: Model clustering performance with its five variants under four non-IID settings.

current dataset (not non-IID settings), with limited accumulation and use of knowledge from other datasets or domains. (4) Compared with existing federated GFM in clustering. This advantage is related to the difference in objectives between general supervised representation learning and graph clustering. They focus more on learning general representations with labels, whose transferability does not necessarily guarantee their suitability for clustering. Topological noise can disrupt node representations through message passing, affecting cluster structure.

3.3 Ablation Studies

Module Ablation. To evaluate the contributions of each proposed components, we design the following model variants: **(1) w/o CS**, which remove subgraph separation and directly generate samples by the origin graphs; **(2) w/o HT**, which remove the structural token dictionary; **(3) w/o ADV**, which removes adversarial graph generation; **(4) w/o PEFT**, which freezes all model parameters during deployment and do not perform client adaptation; and **(5) w/o OSC**, which removes one-shot coordination across clients. The results shown in Fig. 3 lead to the following findings. The complete GFM-FGC consistently achieves the best overall performance. Removing subgraph separation (**w/o CS**) causes degradation because encoding the entire graph introduces cluster-irrelevant noise signal, which misleads the pretraining optimization direction. The inferior performance of (**w/o HT**) indicates that directly generate graph samples without a token dictionary provides limited generalization to unseen graph domains. The token dictionary organizes recurring cluster-biased structures across different neighborhood ranges into reusable discrete tokens, which enables diverse structural compositions and broadens the domain space covered during pretraining. The declines of (**w/o ADV**) arise because limited pretraining graphs cannot sufficiently cover unseen structural variations. During deployment, (**w/o PEFT**) cannot correct the mismatch between the pretrained distribution and each client's local topology, whereas (**w/o OSC**) leaves independently adapted clients with misaligned cluster semantics.

Table 3: Ablation study results of various collaboration strategies.

Settings	Strategy	ACC	NMI	ARI
SM	FedProx	68.5±2.3	6.3±2.0	7.3±1.2
	FedPer	62.4±1.9	8.9±1.4	9.7±1.3
	FedNova	63.9±2.5	7.0±1.8	8.5±2.1
	Ours	72.0±2.0	12.4±1.7	13.5±1.6
SM-BIO-SY	FedProx	63.8±2.2	7.1±1.5	10.5±1.9
	FedPer	64.1±2.1	10.3±1.9	11.6±1.8
	FedNova	66.4±2.0	7.9±2.2	10.4±1.4
	Ours	70.5±2.3	14.8±1.9	16.9±1.6
SN	FedProx	47.3±2.4	12.1±1.8	8.9±2.0
	FedPer	50.8±2.5	13.4±1.6	8.7±2.3
	FedNova	51.5±2.7	12.9±1.8	7.0±2.2
	Ours	54.3±1.7	16.7±1.8	11.5±2.0
CV	FedProx	32.7±2.1	27.0±2.0	23.4±2.2
	FedPer	35.6±1.8	28.2±1.6	25.5±1.5
	FedNova	31.9±2.4	30.5±2.5	21.4±2.2
	Ours	41.4±2.0	47.5±1.6	30.7±1.7

Sharing Strategy Ablation. To investigate whether one-shot knowledge sharing is more effective than the existing multi-round communication method, we replace it with FedProx (Li et al., 2020), FedPer (Arivazhagan et al., 2019), and FedNova (Wang et al., 2020). Based on the experimental results shown in Table 3, the following conclusions can be drawn. These conventional methods consistently yield inferior performance, as they are primarily designed to progressively align client models through repeated communication and parameter aggregation. Consequently, they struggle to extract sufficiently transferable knowledge from a single exchange, particularly under substantial structural and distributional heterogeneity. In contrast, our one-shot method integrates heterogeneous client prototypes in a shared latent factor space, preserving complementary cluster knowledge to improve global clustering performance.

3.4 Communication Overhead Analysis

To evaluate communication efficiency, we compare the communication time and payload of GFM-FGC with SOTA methods in Table 4. Communication overhead increases with the number of clients for all methods. Although GFM-FGC transmits a larger per-round payload to share client-level statistics, its communication time remains comparable to the baselines. More importantly, its main advantage lies in requiring only one aggregation round, eliminating repeated client-server exchanges and reducing cumulative communication costs relative to multi-round FGC methods.

Table 4: Communication overhead comparison under the CV non-IID setting. "T" and "C" denote Time (s) and Cost (KB), respectively.

Clients	Ours		FGCN-DKS		FedGCN	
	T (s)	C (KB)	T (s)	C (KB)	T (s)	C (KB)
1	17.2	240.6	16.8	32.3	15.9	30.5
2	36.2	493.9	35.5	67.3	30.8	61.8
3	57.9	728.2	53.0	96.7	44.8	93.6

3.5 Hyperparameter Sensitivity Analysis

Token Dictionary Capacity We vary the maximum token radius $R \in \{1, 2, 3, 4\}$ and the structural token dictionary size $M \in \{16, 32, 64, 128, 256\}$. Fig. 4 reports the corresponding model performance. The best performance is obtained in balanced settings. Increasing R initially improves the results by incorporating broader neighborhood structures, whereas further enlarging it leads to a slight performance degradation. This is plausibly because excessively large receptive fields introduce irrelevant high-order connectivity and dilute local cluster-relevant patterns. A similar trend is observed for M: a small dictionary provides insufficient capacity to characterize diverse structural patterns, while an excessively large dictionary produces redundant or sparsely activated tokens, making their representations difficult to learn reliably.

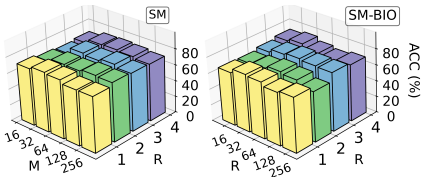

Figure 4: Clustering performance of varying R and M on 2 non-IID settings.

3.6 Generalization Ability Test

To assess whether the pretrained GFM depends on a specific dataset to acquire generalizable knowledge, we evaluate its clustering performance using different pretraining datasets. As shown in Fig. 5, GFM-FGC consistently achieves competitive performance across pretraining sources, with some variations that may reflect source–target distribution differences. These results suggest that pretraining captures transferable structural patterns rather than dataset-specific knowledge. Moreover, structurally similar source and target datasets tend to yield better performance, suggesting that source–target compatibility can further facilitate local adaptation.

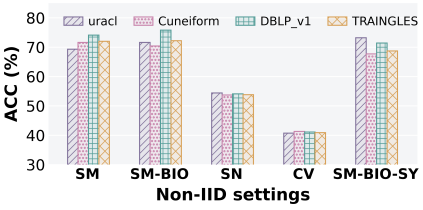

Figure 5: Performance on four pretrain datasets under 5 non-IID settings.

4 Conclusion

In this paper, we present GFM-FGC, a GFM framework for FGC that avoids repeatedly training for new settings through multi-round communication. GFM-FGC follows a train-once, adapt-locally, and aggregate-once paradigm. During pretraining, subgraph separation, topology-conditioned tokenization, and adversarial graph generation enable the model to capture transferable and clustering-friendly graph patterns. For various settings, local adaptation efficiently accommodates client-personalized distribution shifts, while factor aggregation extracts globally shared clustering knowledge without exchanging raw graphs or requiring iterative federated retraining. Extensive experiments on six non-IID settings from diverse domains demonstrate that GFM-FGC achieves consistently competitive clustering performance. These results show that a pretrained graph foundation model can be efficiently transferred to unseen non-IID settings with only lightweight local adaptation and a single global aggregation, providing a practical direction for communication-efficient and FGL with rapid domain adaptation.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, improving the readability of the manuscript, LaTeX formatting, and debugging author-written code. They were not used to develop the proposed methodology, design the experiments, formulate or prove mathematical claims, process the datasets, or interpret the experimental results. All AI-assisted content was carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated Learning with Personalization Layers. *ArXiv Preprint ArXiv:1912.00818*, 2019.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural Deep Clustering Network. In *Proceedings of the Web Conference*, pp. 1400–1410, 2020.

Jinyu Cai, Yi Han, Wenzhong Guo, and Jicong Fan. Deep Graph-level Clustering Using Pseudo-label-guided Mutual Information Maximization Network. *Neural Computing and Applications*, 36(16):9551–9566, 2024a.

Jinyu Cai, Yunhe Zhang, Jicong Fan, Yali Du, and Wenzhong Guo. Dual Contrastive Graph-Level Clustering with Multiple Cluster Perspectives Alignment. In *International Joint Conference on Artificial Intelligence*, pp. 3770–3779, 2024b.

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, 2024c.

Jinyu Cai, Yunhe Zhang, and Jicong Fan. Self-Discriminative Modeling for Anomalous Graph Detection. In *Proceedings of the International Conference on Machine Learning*, pp. 6387–6408, 2025a.

Jinyu Cai, Yunhe Zhang, Fusheng Liu, and See-Kiong Ng. Leveraging Diffusion Model as Pseudo-Anomalous Graph Generator for Graph-Level Anomaly Detection. In *Proceedings of the International Conference on Machine Learning*, pp. 3782–3794, 2025b.

Man-Sheng Chen, Pei-Yuan Lai, De-Zhang Liao, Chang-Dong Wang, and Jian-Huang Lai. Graph Prompt Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7): 5794–5805, 2025.

Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52464–52489, 2023.

Lele Fu, Tianchi Liao, Sheng Huang, Bowen Deng, Shirui Pan, and Chuan Chen. Unsupervised federated graph learning. *Advances in Neural Information Processing Systems*, 38:97431–97457, 2026.

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*, 195:108212–108212, 2025.

Renda Han, Junlong Wu, Wenxuan Tu, Jingxin Liu, Haotian Wang, and Jieren Cheng. Federated Graph-level Clustering Network with Attribute Inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21585–21593, 2026.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced Masked Self-supervised Graph Learner. In *Proceedings of the ACM Web Conference*, pp. 737–746, 2023.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative Pre-training of Graph Neural Networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1857–1867, 2020.

Runze Li, Di Jin, Xiaobao Wang, Dongxiao He, Bingdao Feng, and Zhen Wang. Single-node trigger backdoor attacks in graph-based recommendation systems. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Machine Learning and Systems*, pp. 429–450, 2020.

Xunkai Li, Guohao Fu, Yuming Ai, Zhengyu Wu, Hongchao Qin, Rong-Hua Li, and Guoren Wang. Toward federated multimodal graph foundation models: A topology-aware multimodal alignment framework. *arXiv preprint arXiv:2607.15687*, 2026.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. Learn from Relational Correlations and Periodic Events for Temporal Knowledge Graph Reasoning. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1559–1568, 2023.

Tianchi Liao, Ziyue Xu, Qing Hu, Hong-Ning Dai, Huaiwei Huang, Zibin Zheng, and Chuan Chen. FedBrb: A solution to the Small-to-Large Scenario in Device-Heterogeneity Federated Learning. *IEEE Transactions on Mobile Computing*, pp. 1–14, 2025a.

Tianchi Liao, Lei Zhang, Jinghua Yang, Chuan Chen, and Zibin Zheng. CoaTr: A convolutional Autoregressive Tensor-Ring Decomposition Method for Sparse Spatio-temporal Traffic Data. *Neurocomputing*, 617:129006, 2025b.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *International Conference on Learning Representations*, 2024a.

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*, number 18, pp. 18870–18878, 2025a.

Jingxin Liu, Renda Han, Wenxuan Tu, Haotian Wang, Junlong Wu, and Jieren Cheng. Federated Node-Level Clustering Network with Cross-Subgraph Link Mending. In *International Conference on Machine Learning*, pp. 38540–38556, 2025b.

Jingxin Liu, Xiangyan Tang, Renda Han, Wenxuan Tu, and Ruili Wang. Adaptive Feature Boosting and Distribution Refinement for Graph Clustering. *Pattern Recognition*, pp. 112309, 2025c.

Jingxin Liu, Wenxuan Tu, Renda Han, Junlong Wu, Haotian Wang, Guohui Liu, Xiangyan Tang, and Yue Yang. Personalized Federated Graph-Level Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23721–23729, 2026a.

Jingxin Liu, Wenxuan Tu, Haotian Wang, Renda Han, Haoyi Li, Junlong Wu, and Xiangyan Tang. Causally-Aware Attribute Completion for Incomplete Federated Graph Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23730–23738, 2026b.

Meng Liu, Ke Liang, Siwei Wang, Xingchen Hu, Sihang Zhou, and Xinwang Liu. Deep Temporal Graph Clustering: A comprehensive Benchmark and Datasets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11561–11578, 2025d.

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

Ruyue Liu, Rong Yin, Xiangzhen Bo, Xiaoshuai Hao, Xingrui Zhou, Yong Liu, Liang Li, Jinwen Zhong, Can Ma, and Weiping Wang. Communication-efficient personalized federal graph learning via low-rank decomposition. *Pattern Recognition*, pp. 114360, 2026c.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying Pre-training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference*, pp. 417–428, 2023.

Zhaowei Liu, Dong Yang, Shenqiang Wang, and Hang Su. Adaptive Multi-Channel Bayesian Graph Attention Network for Iot Transaction Security. *Digital Communications and Networks*, 10(3): 631–644, 2024b.

Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. Anomalygfm: Graph Foundation Model for Zero/few-shot Anomaly Detection. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2326–2337, 2025.

Xiaorui Su, Pengwei Hu, Dongxu Li, Bowei Zhao, Zhaomeng Niu, Thomas Herget, Philip S Yu, and Lun Hu. Interpretable identification of cancer genes across biological networks via transformer-powered graph representation learning. *Nature biomedical engineering*, 9(3):371–389, 2025.

Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1717–1727, 2022. doi: 10.1145/3534678.3539249.

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*, pp. 9953–9961, 2023.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500, 2024. doi: 10.1145/3626772.3657775.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Lingren Wang, Wenxuan Tu, Jiaxin Wang, Xiong Wang, Jieren Cheng, and Jingxin Liu. Fedigl: Federated invariant graph learning for non-iid graphs. *Advances in Neural Information Processing Systems*, 38:117917–117940, 2026a.

Xiaobao Wang, Renda Han, Ronghao Fu, and Di Jin. Federated Graph-level Clustering Network with Dual Knowledge Separation. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Xiaobao Wang, Ruoxiao Sun, Yujun Zhang, Bingdao Feng, Dongxiao He, Luzhi Wang, and Di Jin. Stealthy yet effective: Distribution-preserving backdoor attacks on graph classification. *Advances in Neural Information Processing Systems*, 38:65501–65529, 2026c.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V. Chawla, Chuxu Zhang, and Yanfang Ye. Towards graph foundation models: Learning generalities across graphs via task-trees. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 65518–65555, 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 ACM Web Conference*, pp. 925–934, 2026.

Zhengyu Wu, Yinlin Zhu, Xunkai Li, Ziang Qiu, Rong-Hua Li, Guoren Wang, and Chenghu Zhou. Fedbook: A unified federated graph foundation codebook with intra-domain and inter-domain knowledge modeling. *arXiv preprint arXiv:2510.07755*, 2025.

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated Graph Classification Over Non-Iid Graphs. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 18839–18852, 2021.

Yiwei Zhang. Social network user profiling for anomaly detection based on graph neural networks. In *2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA)*, pp. 1197–1201. IEEE, 2025.

Yinlin Zhu, Xunkai Li, Jishuo Jia, Miao Hu, Di Wu, and Meikang Qiu. Towards effective federated graph foundation model via mitigating knowledge entanglement. *Advances in Neural Information Processing Systems*, 38:64599–64628, 2026a.

Yinlin Zhu, Di Wu, Xianzhi Zhang, Yuming Ai, Xunkai Li, Miao Hu, and Guocong Quan. Rethinking federated graph foundation models: A graph-language alignment-based approach. *arXiv preprint arXiv:2601.21369*, 2026b.

A Related Work

A.1 Graph Foundation Model

Graph Foundation Models (GFMs) aim to learn transferable graph knowledge from large-scale graph data and adapt it to diverse downstream tasks. Early graph pre-training methods (Hu et al., 2020) mainly obtain general graph representations. For example, GraphMAE (Hou et al., 2023) adopts masked attribute reconstruction to pre-train graph autoencoders. To better bridge the gap between pre-training and downstream adaptation, GPPT (Sun et al., 2022) introduces graph prompt tuning by transforming downstream node classification into an edge-prediction-like task, while GraphPrompt (Liu et al., 2023) further unifies pre-training and downstream tasks under a common template with learnable graph prompts. GPF proposes a more general prompt tuning strategy by adding learnable prompt features to node attributes, making prompt-based adaptation applicable to different pre-trained GNNs Fang et al. (2023). Recent studies further explore more general-purpose graph models across domains and tasks. OFA (Liu et al., 2024a) represents heterogeneous graph data as text-attributed graphs and introduces nodes-of-interest with graph prompts, enabling one model to handle node-, link-, and graph-level classification tasks across different domains. GraphGPT (Tang et al., 2024) incorporates graph structural knowledge into large language models through graph instruction tuning, improving the ability of LLMs to understand and reason over graph-structured data. More recently, GIT (Wang et al., 2025) learns cross-task generalities through task-tree reconstruction and supports fine-tuning, in-context learning, and zero-shot transfer across many graph datasets. Recent methods have explored combining federated learning with foundation models. FedGFM+ mitigates cross-domain knowledge entanglement, while FedBook improves codebook aggregation across clients. FedGALA replaces discrete quantization with continuous graph-language alignment, and FedGAMMA further extends this paradigm to multimodal graphs.

Although the former exhibit strong reusability and transferability, they are primarily developed for centralized graph learning or supervised downstream tasks. The latter, moreover, either require label supervision, are driven by vision–language models, or learn general-purpose representations without focusing on specific clustering tasks.

A.2 Federated Graph Clustering

Federated graph clustering (FGC) aims to collaboratively discover latent groups from distributed and unlabeled graph collections while keeping raw graph data local. FedGCN (Liu et al., 2025a) makes an early attempt in this direction by extracting topology-oriented cluster prototypes from non-IID graphs at each client and generating global consensus prototypes at the server to guide local clustering. Beyond directly sharing clustering prototypes, FedIGL (Wang et al., 2026a) explores invariant graph learning for heterogeneous federated graphs by separating client-agnostic and client-specific subgraph patterns, and demonstrates its effectiveness on graph-level clustering tasks. Building on this line, FGCN-DKS (Wang et al., 2026b) separates cluster-oriented and personalized subgraph patterns and leverages the former to guide personalized aggregation, addressing both intra- and inter-client heterogeneity. PERFECT (Liu et al., 2026a) introduces personalized federated graph-level clustering by estimating inter-client affinities from privacy-preserving representative samples and constructing client-specific aggregation strategies, while FGCN-AKC (Han et al., 2025) alleviates consensus failure caused by personalized knowledge discrepancies through local knowledge enhancement and global prototype alignment. In addition to structural heterogeneity, FedAI (Han et al., 2026) leverages structure-guided attribute inference to recover missing node features and facilitate cross-client knowledge transfer. More recently, FedCND (Wu et al., 2026) further relaxes the common assumption that all clients share the same number of clusters by decoupling common and

Algorithm 1: GFM-FGC: Pretrain Once, Adapt and Federate Once

Input: Pretraining graphs $\mathcal{D}^{\mathrm{pre}}$; private client graphs $\{\mathrm{G}_k\}_{k=1}^{K}$; number of latent clusters C.
Output: Client cluster assignments $\{\hat{\boldsymbol{y}}_k\}_{k=1}^{K}$.

1. Learn intervention-aware edge masks;
2. Split each pretraining graph into $(\bar{\mathrm{G}}, \tilde{\mathrm{G}})$ by Eqs. (1)-(2).
3. Learn structural vocabularies by Eqs. (3)-(4).
4. Generate conditional subgraphs $\hat{G}$ by Eq. (5).
5. Alternately optimize the discriminator, generator by Eqs. (7)-(8).
6. The server distributes $\boldsymbol{\Theta}^0$, $\mathscr{E}$ and $\mathcal{M}^0$ to each client.
7. On each client k in parallel:
   (a) Freeze the backbone and optimize the local adapters and cluster head using Eq. (11).
   (b) Construct $(\mathcal{P}_k, \boldsymbol{R}_k)$ by Eqs. (10) and (12).
   (c) Upload them.
8. `On the server.`
9. Solves the relational fused GW barycenter in Eq. (13).
10. Returns $(\bar{\mathcal{M}}, \boldsymbol{T}_k)$ to each client k.
11. Each client performs confidence-gated prototype feedback by Eq. (16)
12. Outputs clustering results $\hat{y}_{ki} = \arg\max_j p^g_{kij}$.

personalized graph patterns and employing dual-prototype optimization to accommodate heterogeneous cluster numbers across clients. Although these methods have advanced FGC under various forms of heterogeneity, most rely on task-specific federated optimization and iterative communication between clients and the server. In contrast, our GFM-FGC investigates how transferable knowledge learned by a reusable graph foundation model can be efficiently adapted to unsupervised federated graph-level clustering through lightweight local adaptation and one-round aggregation, reducing repeated domain training and communication overhead.

B Details of Experimental Settings

B.1 Benchmark Datasets

The details of the statistics of the datasets and the non-IID setting are shown in Table 5 and Table 6.

B.2 More Details of Implementation

For all experiments, we utilize the IMDB-BINARY dataset to pre-train the foundation model. We only perform this process once, then the backbone and the hierarchical structural token dictionaries are frozen and reused in each non-IID settings. The graph encoder contains 3 layers with hidden dimension 64. Hierarchical structural tokens are extracted up to radius R using structural token dictionaries, each containing M entries. Pretraining uses AdamW with learning rate 1e-3, weight decay 5e-4, batch size 128, and 100 epochs. At deployment, we fine-tune parameters for 10 local epochs; the backbone and structural token dictionaries remain frozen. Each client then participates in exactly one upload-aggregation-download cycle. All competing methods use their official implementations when available. Their hyperparameters are taken from the original papers or tuned over the same budget using only unsupervised validation objectives. Experiments are run on PyTorch 2.4.0 and CUDA 12.4. For supervised FGL methods and federated supervised GFMs, we remove labels and adapt them to a clustering objective. The aggregation mechanisms are driven by the same unsupervised local objective.

Table 5: Statistics of the 16 benchmark datasets from five domains. Avg. Nodes and Avg. Edges denote the average numbers of nodes and edges per graph, respectively.

Domain	Dataset	#Graphs	#Classes	Avg. Nodes	Avg. Edges
Small molecules	MUTAG	188	2	17.93	19.79
	BZR	405	2	35.75	38.36
	COX2	467	2	41.22	43.45
	DHFR	756	2	42.43	44.54
	PTC_MR	344	2	14.29	14.69
	AIDS	2,000	2	15.69	16.20
	BZR_MD	306	2	21.30	225.06
Bioinformatics	DD	1,178	2	284.32	715.66
	PROTEINS	1,113	2	39.06	72.82
Social networks	COLLAB	5,000	3	74.49	2457.78
	IMDB-MULTI	1,500	3	13.00	65.94
Synthetic graphs	SYNTHETIC	300	2	100.00	196.00
	SYNTHIE	400	4	95.00	172.93
Computer vision	LETTER-LOW	2,250	15	4.68	3.13
	LETTER-MED	2,250	15	4.67	3.21
	LETTER-HIGH	2,250	15	4.67	4.50

Table 6: Federated evaluation scenarios. Each constituent dataset is held by one client.

Setting	#Clients	Domains	Heterogeneity type
SM	7	Small molecule	Cross-dataset
SN	2	Social network	Cross-dataset
CV	3	Computer vision	Cross-dataset
SM-BIO	9	Molecule, bioinformatics	Cross-domain
SM-BIO-SY	10	Molecule, bioinformatics, synthetic	Cross-domain
SN-SY	3	Social network, synthetic	Cross-domain

B.3 DETAILS OF EVALUATION METRICS.

We evaluate the model performance using four complementary metrics: clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and macro-F1. All metrics measure the agreement between the predicted clusters and the ground-truth graph categories, and higher values indicate better clustering performance. Ground-truth labels are used *only* for evaluation and are never involved in model training.

Notation. Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ be the set of N graphs. For each graph v_i, let $y_i \in \{1, \ldots, C\}$ denote its ground-truth class label, and let $\hat{y}_i \in \{1, \ldots, K\}$ denote the predicted cluster assignment. The contingency table between ground-truth labels and predicted clusters is

$$n_{ij} = \big|\{v \in \mathcal{V} : y_v = i,\ \hat{y}_v = j\}\big|, \quad i = \{1, \ldots, C, \}, j = \{1, \ldots, K\}. \tag{17}$$

We further define the row and column marginals

$$a_i = \sum_{j=1}^{K} n_{ij}, \qquad b_j = \sum_{i=1}^{C} n_{ij}, \tag{18}$$

so that $\sum_{i=1}^{C} a_i = \sum_{j=1}^{K} b_j = N$. Here a_i is the size of ground-truth class i, and b_j is the size of predicted cluster j.

Clustering Accuracy (ACC). ACC first finds the optimal one-to-one matching between predicted clusters and ground-truth classes, and then reports the fraction of correctly matched graphs. Formally, let Π be the set of all injective mappings from the ground-truth classes to the predicted

clusters. Π is the set of all permutations of $\{1, \ldots, K\}$. Then

$$\mathrm{ACC} = \max_{\pi \in \Pi} \frac{1}{N_G} \sum_{i=1}^{C} n_{i,\pi(i)}. \tag{19}$$

The optimal assignment $\pi^\star$ can be obtained efficiently by the Hungarian algorithm in $O(\max(C, K)^3)$ time. The resulting $\pi^\star$ is also reused for the macro-F1 computation described below.

Normalized Mutual Information (NMI). NMI measures the statistical dependence between the ground-truth labels Y and the predicted clusters $\hat{Y}$ via mutual information, normalized by their entropies. Let

$$H(Y) = -\sum_{i=1}^{C} \frac{a_i}{N_G} \log \frac{a_i}{N_G}, \tag{20}$$

$$H(\hat{Y}) = -\sum_{j=1}^{K} \frac{b_j}{N_G} \log \frac{b_j}{N_G}, \tag{21}$$

be the entropy of the ground-truth partition and the predicted partition, respectively, and let

$$I(Y; \hat{Y}) = \sum_{i=1}^{C} \sum_{j=1}^{K} \frac{n_{ij}}{N_G} \log \frac{n_{ij}/N_G}{\big(a_i/N_G\big)\big(b_j/N_G\big)} \tag{22}$$

be their mutual information. We adopt the arithmetic-mean normalization:

$$\mathrm{NMI} = \frac{2\, I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}. \tag{23}$$

By convention, $0 \log 0 = 0$. If both partitions are trivial, i.e., $H(Y) = H(\hat{Y}) = 0$, we set $\mathrm{NMI} = 1$. NMI lies in $[0, 1]$, with 1 indicating perfect agreement.

Adjusted Rand Index (ARI). ARI measures the agreement between two partitions while correcting for chance. Let $\binom{n}{2} = n(n-1)/2$ denote the number of unordered pairs. Define

$$\mathrm{Index} = \sum_{i=1}^{C} \sum_{j=1}^{K} \binom{n_{ij}}{2}, \tag{24}$$

which counts the number of graph pairs that are grouped together in both partitions. The expected index under a random model with fixed marginals is

$$\mathrm{Expected} = \frac{\left(\sum_{i=1}^{C} \binom{a_i}{2}\right)\left(\sum_{j=1}^{K} \binom{b_j}{2}\right)}{\binom{N_G}{2}}, \tag{25}$$

and the maximum achievable index is

$$\mathrm{Max} = \frac{1}{2}\left[\sum_{i=1}^{C} \binom{a_i}{2} + \sum_{j=1}^{K} \binom{b_j}{2}\right]. \tag{26}$$

The adjusted Rand index is then

$$\mathrm{ARI} = \frac{\mathrm{Index} - \mathrm{Expected}}{\mathrm{Max} - \mathrm{Expected}}. \tag{27}$$

ARI takes values in $[-1, 1]$, where 1 indicates perfect agreement, 0 corresponds to random labeling, and negative values indicate agreement worse than chance.

Macro-F1. Macro-F1 evaluates the quality of the matched clusters by treating the clustering result as a classification outcome after the optimal assignment $\pi^\star$ obtained for ACC. Assuming $K = C$ (as in our experiments), for each ground-truth class i we define

$$\mathrm{TP}_i = n_{i,\pi^\star(i)}, \tag{28}$$

$$\mathrm{FP}_i = b_{\pi^\star(i)} - \mathrm{TP}_i, \tag{29}$$

$$\mathrm{FN}_i = a_i - \mathrm{TP}_i. \tag{30}$$

The corresponding precision, recall, and F1 score for class i are

$$P_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}, \tag{31}$$

$$R_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{32}$$

$$F1_i = \frac{2P_iR_i}{P_i + R_i}. \tag{33}$$

When a denominator is zero, the corresponding quantity is set to 0. Macro-F1 is the unweighted average over all C classes:

$$\text{Macro-F1} = \frac{1}{C}\sum_{i=1}^{C} F1_i. \tag{34}$$

Macro-F1 lies in $[0, 1]$, with 1 indicating perfect clustering. Unlike micro-averaged F1, macro-F1 weights all classes equally and is therefore robust to class imbalance.

Summary. ACC and macro-F1 are based on the optimal Hungarian matching between clusters and classes and therefore directly measure assignment correctness; NMI and ARI measure partition agreement from an information-theoretic and a pairwise perspective, respectively, and do not require an explicit cluster-to-class matching. Together, these four metrics provide a comprehensive evaluation of clustering quality. Ground-truth labels are used solely to compute these scores and are never accessed during training.

C Theoretical Analysis

We analyze how one-shot prototype alignment affects the final clustering error. For clarity, we fix one client with positive aggregation weight and omit its index. Let N_G denote the number of local graphs and $C \geq 2$ the number of global prototypes. We use

$$\Omega(\Pi) = \sum_{c,j} \Pi_{cj}(\log \Pi_{cj} - 1), \qquad \mathrm{KL}(x\|y) = \sum_{\ell} \left[x_\ell \log \frac{x_\ell}{y_\ell} - x_\ell + y_\ell \right].$$

Assume finite costs, positive reference masses a, b, and nonnegative regularization coefficients with $\varepsilon > 0$.

Let $\Pi^\star > 0$ be a stationary transport plan of Eq. (13) with the server variables held fixed. Define $u_c = \sum_j \Pi^\star_{cj}$, $v_j = \sum_c \Pi^\star_{cj}$, and $T_{cj} = \Pi^\star_{cj}/u_c$. For symmetric relation matrices R and $\bar{R}$, define the effective alignment cost as

$$\begin{aligned} D_{cj} &= C_{cj} + \lambda_R G_{cj} + \tau_c \log \frac{v_j}{b_j}, \\ G_{cj} &= 2\sum_{c',j'} (R_{cc'} - \bar{R}_{jj'})^2 \Pi^\star_{c'j'}. \end{aligned} \tag{35}$$

This cost accounts for prototype discrepancy, relational consistency, and target-mass regularization.

Theorem 1 (Error bound for one-shot knowledge sharing)**.** *Let $\sigma(c)$ denote the correct global slot of local cluster c. Suppose that, for some $\kappa > 0$,*

$$D_{cj} - D_{c,\sigma(c)} \geq \kappa, \qquad j \neq \sigma(c). \tag{36}$$

Then the incorrect alignment mass satisfies

$$1 - T_{c,\sigma(c)} \le \delta := \frac{(C-1)e^{-\kappa/\varepsilon}}{1 + (C-1)e^{-\kappa/\varepsilon}}. \tag{37}$$

Let y_i denote the true class of graph i after fixing a common global label permutation, and define the average local assignment error as

$$\bar{\eta} = \frac{1}{N_G} \sum_{i=1}^{N_G} \left(1 - \sum_{c:\sigma(c)=y_i} q_{ic} \right).$$

Keep the targets $\tilde{\mathbf{y}}_i$ and weights Γ_i from Eq. (14) fixed during feedback. Define

$$\mathcal{E}_{\mathrm{fb}} = \frac{1}{N_G} \sum_{i=1}^{N_G} \Gamma_i \, \mathrm{KL}(\tilde{\mathbf{y}}_i \| \mathbf{p}_i^g), \qquad \gamma = \min_i \Gamma_i.$$

Then $\gamma > 0$, and the empirical clustering error satisfies

$$\widehat{\mathrm{Err}} := \frac{1}{N_G} \sum_{i=1}^{N_G} \mathbf{1}\{\arg\max_j p_{ij}^g \neq y_i\} \le 2\bar{\eta} + 2\delta + \sqrt{\frac{2\mathcal{E}_{\mathrm{fb}}}{\gamma}}. \tag{38}$$

Proof. The stationarity condition of Eq. (13) gives

$$D_{cj} + \varepsilon \log \Pi_{cj}^\star + \tau_r \log \frac{u_c}{a_c} = 0.$$

Subtracting the condition for slot $\sigma(c)$ from that for slot j cancels the source-mass term, yielding

$$\frac{T_{cj}}{T_{c,\sigma(c)}} = \exp\left(-\frac{D_{cj} - D_{c,\sigma(c)}}{\varepsilon} \right) \le e^{-\kappa/\varepsilon}.$$

Summing over the $C-1$ incorrect slots proves Eq. 37. Since $\kappa > 0$, every row of T is nonuniform. Hence $\Gamma_c = 1 - H(T_{c,:})/\log C > 0$ and $\Gamma_i = \sum_c q_{ic}\Gamma_c > 0$, implying $\gamma > 0$.

For each graph, let $\eta_i = 1 - \sum_{c:\sigma(c)=y_i} q_{ic}$. Using $\tilde{y}_{ij} = \sum_c q_{ic} T_{cj}$, we obtain

$$\tilde{y}_{i,y_i} \ge \sum_{c:\sigma(c)=y_i} q_{ic} T_{c,\sigma(c)} \ge (1-\eta_i)(1-\delta).$$

Therefore, $1 - \tilde{y}_{i,y_i} \le \eta_i + \delta$. By Pinsker's inequality, with natural logarithms,

$$1 - p_{i,y_i}^g \le \eta_i + \delta + \sqrt{\frac{1}{2}\,\mathrm{KL}(\tilde{\mathbf{y}}_i \| \mathbf{p}_i^g)}.$$

Averaging over graphs, applying Jensen's inequality, and using $\Gamma_i \ge \gamma$ gives

$$\frac{1}{N_G} \sum_i (1 - p_{i,y_i}^g) \le \bar{\eta} + \delta + \sqrt{\frac{\mathcal{E}_{\mathrm{fb}}}{2\gamma}}.$$

An incorrect prediction implies $p_{i,y_i}^g \le 1/2$, so $\mathbf{1}\{\arg\max_j p_{ij}^g \neq y_i\} \le 2(1 - p_{i,y_i}^g)$. This proves Eq. 38. □

Theorem 1 decomposes the final error into local assignment error, alignment error, and feedback fitting error. Under the effective separation condition, the alignment error decreases exponentially with κ/ε. Moreover, the feedback error is directly related to Eq. (16):

$$\mathcal{E}_{\mathrm{fb}} = \mathcal{L}_{\mathrm{proto}} - \frac{1}{N_G} \sum_i \Gamma_i H(\tilde{\mathbf{y}}_i).$$

Thus, fitting the fixed federated targets controls the feedback term in the bound.