FEDHC: FEDERATED GRAPH CLUSTERING WITH STRUCTURE MISSING VIA HYPERGRAPH CONTEXTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph-level clustering aims to group decentralized graph data without sharing private graphs. Existing methods usually summarize client knowledge with graph embeddings, cluster prototypes, or pairwise relations. However, they often assume complete graph structures, while real federated data may suffer from structure missing. Directly clustering such graphs can produce unreliable graph representations and biased prototypes, which further affects global clustering. To address this problem, we propose FedHC, a federated graph clustering method that uses hypergraph contexts for structure recovery. On each client, FedHC first builds a graph-instance hypergraph from locally observed graphs, where each hyperedge connects graphs with similar representations and structural patterns. The resulting local hypergraph contexts are used to repair damaged graph structures and obtain more reliable graph representations, and are further summarized into compact context prototypes. The server then builds a prototype hypergraph to integrate related contexts across clients and forms global hypergraph priors, which are distributed back to clients to further guide local structure recovery and clustering. Through this design, FedHC can better adapt to structure missing by exploiting both client-specific structural patterns and cross-client shared structural information within the federated learning process. Experiments on multiple graph benchmark datasets under different structure missing settings show that FedHC outperforms or remains competitive with existing federated graph clustering methods.

1 INTRODUCTION

Graph-structured data appear in social networks, recommendation platforms, molecular analysis, and financial risk detection (Liang et al., 2023; Wang et al., 2025; Agrawal et al., 2024; Zhuang et al., 2026). Deep graph clustering learns graph representations and discovers latent groups without label supervision (Bo et al., 2020; Tu et al., 2021; Cai et al., 2024a;b; Liu et al., 2025b;c). In many applications, however, graph datasets are held by different organizations or devices and cannot be centralized because of privacy, regulation, or data ownership. Federated graph-level clustering (FGC) addresses this setting by clustering decentralized graph collections while keeping each client's raw graph structures and attributes local (Liu et al., 2025a; Han et al., 2025; 2026; Liu et al., 2026a).

Recent FGC methods study this problem mainly through knowledge sharing across clients. FedGCN (Liu et al., 2025a) shares cluster prototypes for global consensus; FedPKA (Wu et al., 2025) and PERFECT (Liu et al., 2026a) perform personalized aggregation; FGCN-AKC (Han et al., 2025) compensates for heterogeneous knowledge; FedAI (Han et al., 2026) focuses on attribute missing; and FedCND (Wu et al., 2026) handles inconsistent cluster numbers. These methods improve FGC under non-IID graph distributions, but they usually summarize client knowledge as graph embeddings, point prototypes, pairwise affinities, or personalized weights. Such summaries assume that the local graph structure used to produce them is reliable enough. This assumption is weak when clients suffer from *structure missing*. Structure missing can arise from incomplete interaction records, noisy graph construction, limited institutional coverage, or client-specific collection policies. Once edges are missing, directly learning from the observed topology may distort graph representations, corrupt structural statistics, and bias uploaded prototypes. Graph structure learning has shown that observed graphs can be noisy or incomplete and that learning a refined structure can benefit downstream graph models (Zhu et al., 2021). Graph auto-encoders also show that graph representations can be used to decode edge likelihoods and reconstruct graph structure (Kipf & Welling,

2016). These studies support structure-aware refinement before clustering. Yet centralized structure-missing graph clustering (Hu et al., 2026) can access the whole graph collection, while FGC must estimate useful structural guidance from fragmented local observations without sharing private graphs.

The remaining question is where the recovery prior should come from. For graph-level data, different graph samples usually have no node-level correspondence, so one cannot simply copy edges from one graph to another. A more suitable clue lies in shared structural patterns: graphs from similar latent clusters often have related degree distributions, densities, motifs, or subtree patterns. Weisfeiler-Lehman graph kernels support this view by measuring graph similarity through shared subtree patterns (Shervashidze et al., 2011). Hypergraph learning further provides a natural way to represent group relations beyond pairwise edges (Zhou et al., 2006; Feng et al., 2019; Chien et al., 2022). Thus, a hyperedge over related graph samples can summarize group-level structural evidence and provide a prior for refining representations learned from damaged graphs. Figure 1 illustrates the difficulty in FGC. Even when clients contain related latent graph patterns, their observed graphs may contain different missing edge sets because each client has a different structure-retention ratio ρ_m. Point-level knowledge uploaded to the server, such as graph embeddings $\mathbf{z}_i$, prototypes $\boldsymbol{\mu}_k$, and pairwise similarities s_{ij}, can then become a biased summary of the underlying clusters. If the server aggregates such biased summaries, the returned guidance may further enlarge local errors caused by missing structures. The local and server sides therefore play distinct but connected roles: local graph groups provide recovery contexts inside each client, while server-side groups of context prototypes decide which structural priors remain reliable across clients with different missing patterns.

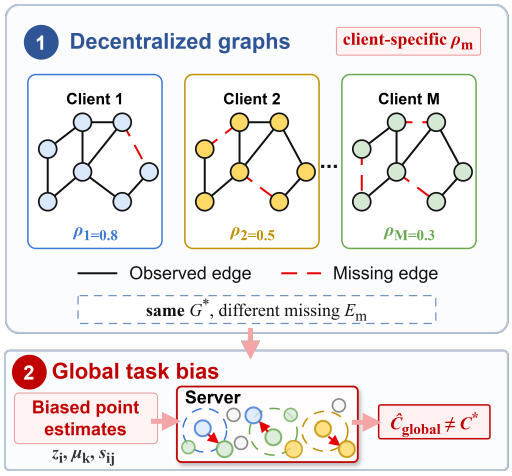


Figure 1: Client-dependent structure missing in FGC: different missing edge sets E_m bias uploaded $\{\mathbf{z}_i, \boldsymbol{\mu}_k, s_{ij}\}$, leading to $\widehat{C}_{\text{global}} \neq C^*$.

We propose **FedHC**, a federated graph-level clustering framework that uses hypergraph contexts to mitigate structure-missing bias. On each client, FedHC builds a graph-instance hypergraph from locally observed graphs. Each hyperedge gathers graphs with similar representations and structural patterns, producing a local hypergraph context used to refine structure-aware representations before clustering. Clients do not upload private graphs or graph neighborhoods; they upload compact hypergraph context prototypes. The server builds a prototype hypergraph over these uploaded contexts and forms global hypergraph priors from related recovery contexts across clients. These priors are returned to clients to guide local representation refinement and clustering. Local and global hypergraphs are both needed: the former estimates reliable context from related graphs inside a client, while the latter filters client-specific missing bias and supplies cross-client structural information.

Our main contributions are as follows:

- **Problem Setting.** We study federated graph-level clustering with structure missing and argue that damaged graphs require structure-aware refinement before clustering, since isolated embeddings, single prototypes, and pairwise affinities can carry structure-missing bias.
- **Method.** We propose FedHC, which uses client-side graph-instance hypergraphs as local structural-context priors and server-side prototype hypergraphs as global hypergraph priors.
- **Evaluation.** Experiments on multiple graph benchmark datasets and six structure-missing federated settings show that FedHC outperforms or remains competitive with recent federated graph clustering methods.

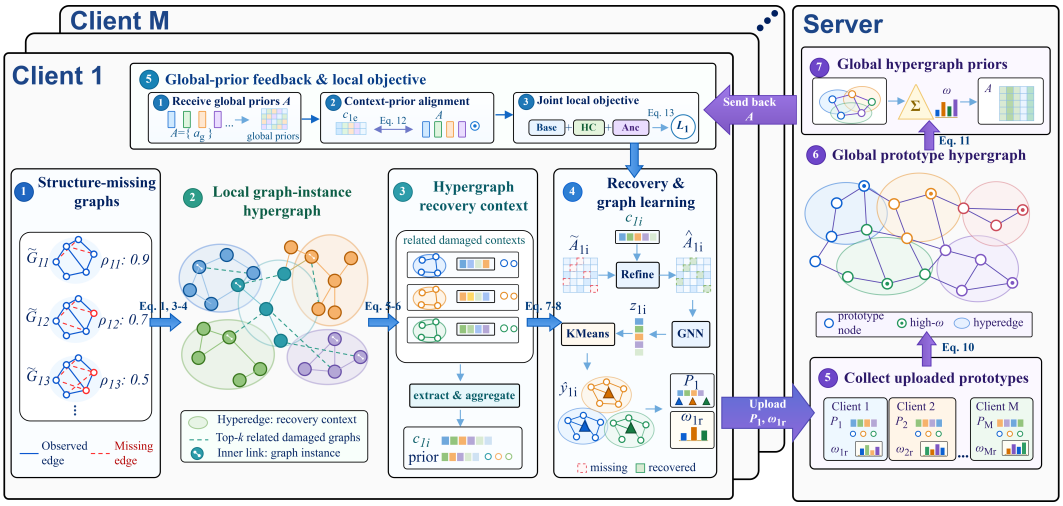


Figure 2: Overall framework of FedHC. Clients build local hypergraph contexts from structure-missing graphs to refine structure-aware representations before clustering, and upload compact context/cluster prototypes. The server builds a prototype hypergraph and broadcasts global priors for the next refinement round.

2 METHODOLOGY

We present **FedHC** for federated graph-level clustering with structure missing. Existing FGC methods usually exchange graph embeddings, cluster prototypes, or personalized knowledge (Liu et al., 2025a; Wu et al., 2025; Han et al., 2026; Liu et al., 2026a; Wu et al., 2026). Under structure missing, however, a single embedding or prototype can be a biased observation of the underlying cluster semantics. FedHC therefore describes a cluster as a *hypergraph context*: locally, a group of related graphs jointly supports one cluster pattern; globally, a group of local contexts jointly supports one global structural prior. This design follows the view that hypergraphs naturally encode higher-order relations beyond pairwise edges (Feng et al., 2019; Chien et al., 2022).

FedHC contains two coupled innovation modules, as illustrated in Figure 2. The *Local Hypergraph Context module (LHC)* groups damaged graphs into reliability-aware recovery contexts and refines structure-aware representations before KMeans clustering. Each client uploads only compact context/cluster prototypes with reliability scores, while raw graphs, graph neighborhoods, and graph-level embeddings remain local. The *Global Prototype Hypergraph module (GPH)* builds a server-side prototype hypergraph from these uploads and aggregates reliable prototype hyperedges into global priors for the next refinement round. Thus, LHC provides local group-wise structural evidence, while GPH redistributes cross-client priors to reduce client-specific missing bias. Algorithm 1 summarizes the main procedure.

Compared with point-prototype collaboration, FedHC shares context-level knowledge: LHC summarizes groups of related structure-missing graphs, and GPH relates these context prototypes across clients. Thus, global information supports robust clustering without exposing raw graphs or forcing identical prototypes across heterogeneous clients.

2.1 PROBLEM FORMULATION

We consider M clients. Client m owns a private graph dataset $\mathcal{D}_m = \{G_{mi}\}_{i=1}^{N_m}$, where $G_{mi} = (\mathbf{X}_{mi}, \mathbf{A}_{mi})$ is one graph instance. During training, only a structure-missing observation $\widetilde{G}_{mi} = (\mathbf{X}_{mi}, \widetilde{\mathbf{A}}_{mi})$ is available, where part of the edge set is unobserved. Client-dependent missing rates bias local structural statistics and graph embeddings. The goal is to learn clustering assignments $\{\hat{y}_{mi}\}_{i=1}^{N_m}$ without exchanging raw graphs.

Let f_{θ_m} be the local graph encoder. To construct local hyperedges, FedHC first extracts an auxiliary descriptor from the observed graph,

$$\widetilde{\mathbf{z}}_{mi} = f_{\theta_m}(\mathbf{X}_{mi}, \widetilde{\mathbf{A}}_{mi}) \in \mathbb{R}^{d_z} \tag{1}$$

and a lightweight structural summary $\mathbf{s}_{mi}$, such as node count, density, average degree, and structure-retention statistics. We denote the reliability of the observed structure by $\rho_{mi} \in (0, 1]$. These descriptors are used to organize damaged graphs into local hypergraph recovery contexts, while the clustering representation is computed with context-guided structure-aware refinement:

$$\mathbf{z}_{mi} = f_{\theta_m}(\mathbf{X}_{mi}, \widehat{\mathbf{A}}_{mi}). \tag{2}$$

Here $\widehat{\mathbf{A}}_{mi}$ denotes the effective structure used by the encoder, covering explicit refinement, soft structural views, or context regularization. In simulations, ρ_{mi} is the realized edge-retention ratio; otherwise, it can be estimated from observable structural quality indicators.

2.2 Local Hypergraph Context Module (LHC)

Graph-instance hypergraph construction. The local module addresses the first source of structure-missing bias: an isolated damaged graph may not contain enough reliable topology to support a stable cluster decision. FedHC therefore constructs a graph-instance hypergraph $\mathcal{H}_m = (\mathcal{V}_m, \mathcal{E}_m)$ on each client, where each hypergraph node corresponds to one local graph. For graph i, we concatenate its auxiliary descriptor and structural summary as $\mathbf{h}_{mi} = [\widetilde{\mathbf{z}}_{mi} \| \mathbf{s}_{mi}]$. The reliability-aware context affinity between graphs i and j is

$$a_{ij}^m = \rho_{mj} \frac{\mathbf{h}_{mi}^\top \mathbf{h}_{mj}}{\|\mathbf{h}_{mi}\|_2 \|\mathbf{h}_{mj}\|_2}. \tag{3}$$

For each graph i, FedHC creates one local hyperedge by connecting its top-k neighbors under this affinity:

$$e_{mi} = \mathrm{TopK}_j(a_{ij}^m, k). \tag{4}$$

Each hyperedge forms a local recovery context over structurally and semantically related graphs, rather than merely a neighborhood for smoothing embeddings.

Reliability-aware local contexts. For each hyperedge $e \in \mathcal{E}_m$, FedHC computes a local recovery context by reliability-weighted pooling:

$$\mathbf{c}_{me} = \frac{\sum_{i \in e} \rho_{mi} \widetilde{\mathbf{z}}_{mi}}{\sum_{i \in e} \rho_{mi} + \epsilon}. \tag{5}$$

For graph i, the contexts of its incident hyperedges are pooled into a graph-specific recovery prior

$$\bar{\mathbf{c}}_{mi} = \frac{1}{|\mathcal{E}_{mi}|} \sum_{e \in \mathcal{E}_{mi}} \mathbf{c}_{me}, \qquad \mathcal{E}_{mi} = \{e \in \mathcal{E}_m \mid i \in e\}. \tag{6}$$

The damaged structure is then refined, explicitly or implicitly, by

$$\widehat{\mathbf{A}}_{mi} = \mathrm{Refine}(\widetilde{\mathbf{A}}_{mi}; \mathbf{X}_{mi}, \bar{\mathbf{c}}_{mi}, \rho_{mi}, \mathcal{A}), \tag{7}$$

where $\mathrm{Refine}(\cdot)$ denotes explicit edge recovery, a soft adjacency, or an implicit context regularizer, without copying edges or assuming node alignment across graph samples. Raw graph neighborhoods are not uploaded. LHC further imposes a hyperedge consistency loss:

$$\mathcal{L}_{\mathrm{hc}}^m = \frac{1}{|\mathcal{E}_m|} \sum_{e \in \mathcal{E}_m} \frac{\sum_{i \in e} \rho_{mi} \left[1 - \cos(\mathbf{z}_{mi}, \mathbf{c}_{me})\right]}{\sum_{i \in e} \rho_{mi} + \epsilon}. \tag{8}$$

Algorithm 1: Algorithm Procedure of FedHC

Require: $\mathcal{D}_{1:M}$, encoders $f_{\theta_{1:M}}$, rounds T.
Require: Sizes k, k_s; weights $\lambda_{\mathrm{hc}}, \lambda_{\mathrm{anc}}$.
Ensure: Clustering result RS.

Initialize global priors $\mathcal{A} \leftarrow \emptyset$.
for $t = 0, \ldots, T-1$ **do**
  Local Hypergraph Context module (LHC).
  for client m **do**
    Extract $\widetilde{\mathbf{z}}_{mi}$, $\mathbf{s}_{mi}$, and reliability ρ_{mi}.
    Build local hyperedges $\mathcal{E}_m$ by Eq. 3.
    Pool recovery contexts $\mathbf{c}_{me}$ by Eq. 5.
    Refine structure-aware graph representations with local/global priors.
    Encode graphs and run KMeans on $\mathbf{z}_{mi}$.
    Optimize Eq. 13 with two FedHC weights.
    Upload context/cluster prototypes $\mathcal{P}_m$.
  end for
  Global Prototype Hypergraph module (GPH).
  Collect client prototypes $\{\mathcal{P}_m\}_{m=1}^M$.
  Compute prototype affinities b_{uv} by Eq. 10.
  Build top-k_s prototype hyperedges on the server.
  Update global priors $\mathcal{A}$ by Eq. 11.
  Broadcast $\mathcal{A}$ for next-round refinement.
  for client m **do**
    Infer $\hat{y}_{mi}$ from the refined graph embeddings.
  end for
end for
Return $RS = \{\hat{y}_{mi}\}$ from all clients.

This loss makes the recovery context affect the learned graph representation, so the server receives less noisy summaries from damaged structures.

After local optimization and KMeans assignment, client m compresses hyperedge contexts and cluster centers into R_m context/cluster prototypes

$$\mathcal{P}_m = \{(\mathbf{p}_{mr}, \omega_{mr})\}_{r=1}^{R_m}, \tag{9}$$

where $\mathbf{p}_{mr}$ is a local recovery-context prototype associated with the current cluster summary and ω_{mr} is its reliability. Only these compact prototypes are uploaded; raw graphs, graph neighborhoods, and graph-level embeddings remain private.

2.3 Global Prototype Hypergraph Module (GPH)

The global module handles the second source of structure-missing bias: different clients may observe the same latent cluster through different missing rates. Directly averaging uploaded prototypes can mix reliable and unreliable contexts. FedHC instead treats each uploaded prototype as a node in a server-side prototype hypergraph.

Let $\mathcal{P} = \{(\mathbf{p}_u, \omega_u)\}_{u=1}^{R}$ collect all uploaded context/cluster prototypes. The server constructs the prototype affinity

$$b_{uv} = \omega_v \frac{\mathbf{p}_u^\top \mathbf{p}_v}{\|\mathbf{p}_u\|_2 \|\mathbf{p}_v\|_2}, \tag{10}$$

and forms E_u from the top-k_s related prototypes. Each server hyperedge groups related local contexts that jointly support one global structural prior:

$$\mathbf{a}_g = \frac{\sum_{u \in E_g} \omega_u \mathbf{p}_u}{\sum_{u \in E_g} \omega_u + \epsilon}. \tag{11}$$

When the number of priors exceeds the global memory size, the server further clusters them and keeps normalized centers, producing $\mathcal{A} = \{\mathbf{a}_g\}_{g=1}^{G}$.

The server broadcasts $\mathcal{A}$ to clients. Each local recovery context is softly aligned to its most relevant global prior:

$$\mathcal{L}_{\text{anc}}^m = \frac{1}{|\mathcal{E}_m|} \sum_{e \in \mathcal{E}_m} \bar{\rho}_{me} \left[1 - \max_g \cos(\mathbf{c}_{me}, \mathbf{a}_g)\right], \tag{12}$$

where $\bar{\rho}_{me}$ is the hyperedge reliability. Thus, reliable local contexts receive stronger global guidance, while uncertain ones are weakly constrained. The returned priors then guide the next local representation refinement step.

The local objective optimized on client m is

$$\mathcal{L}_m = \mathcal{L}_{\text{base}}^m + \lambda_{\text{hc}} \mathcal{L}_{\text{hc}}^m + \lambda_{\text{anc}} \gamma_t \mathcal{L}_{\text{anc}}^m, \tag{13}$$

where $\mathcal{L}_{\text{base}}^m$ is the backbone clustering objective. λ_{hc} and λ_{anc} weight local context consistency and global-prior feedback. The non-tunable gate $\gamma_t = \mathbb{I}(|\mathcal{A}| > 0)$ disables global feedback before the first server update. The objective therefore has three losses but only two loss weights, and optimizes LHC and GPH within the same training process.

2.4 Efficiency Analysis

FedHC uploads only R_m context prototypes and reliability scores per client, leading to $\mathcal{O}(R_m d_z)$ communication. Local hypergraph construction requires $\mathcal{O}(N_m^2(d_z + d_s))$ similarity computation in the direct form, where d_s is the dimension of the structural summary, and can be implemented with top-k nearest-neighbor search for larger clients. Server-side prototype hypergraph construction costs $\mathcal{O}(R^2 d_z)$ with $R = \sum_m R_m$, where R is much smaller than the total number of private graphs. Thus, the extra cost is concentrated on compact context prototypes, while raw graphs and structure-level graph information remains on clients.

Table 1: Performance comparison across different FGL methods under six non-IID settings. * denotes supervised methods adapted for unsupervised learning. † denotes anomaly detection methods adapted for clustering. ‡ denotes centralized deep graph-level clustering methods adapted for FGL. **Bold** and underlined values denote the best and second-best performance, respectively.

Models	SM^2(7)				SN^3(2)				SM-BIO^2(9)			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
FedSage*	45.6±1.9	8.8±1.8	5.2±0.9	40.2±1.4	33.0±1.9	5.8±1.0	2.6±1.2	33.1±2.2	47.1±2.9	3.7±2.8	2.9±3.5	39.9±0.8
GCFL*	50.7±2.3	6.4±3.1	6.5±3.1	40.2±3.0	32.6±2.3	4.9±1.6	3.0±1.0	35.7±1.9	49.9±2.4	3.4±3.1	2.2±3.0	38.3±2.0
FedStar*	49.5±3.1	8.9±1.7	0.1±1.2	40.8±3.6	32.8±2.6	5.4±1.9	2.8±0.8	35.0±2.5	50.0±2.2	3.9±2.0	2.7±2.6	42.4±2.9
LG-FGAD†	56.6±1.2	14.7±2.5	2.4±1.5	52.8±0.9	24.6±2.4	4.0±2.0	0.1±0.4	18.6±3.8	51.3±2.0	7.0±1.9	5.6±2.3	47.0±3.4
FGAD†	57.8±3.0	16.0±3.4	3.1±4.1	54.2±3.4	27.1±1.9	2.5±1.7	0.1±0.6	25.6±1.9	55.2±1.4	11.6±1.5	1.5±2.6	51.6±2.0
AGDiff†	61.8±1.9	15.4±3.7	11.5±5.0	57.9±3.6	28.1±1.9	3.2±0.9	2.1±0.8	26.9±1.2	53.9±3.2	8.5±2.5	2.5±0.4	49.2±2.2
UDGC‡	47.2±4.3	8.0±3.2	6.7±4.3	45.9±3.0	33.3±2.3	4.6±1.7	2.4±0.6	35.0±2.5	48.9±2.4	7.3±1.9	5.3±0.7	43.3±1.4
DGLC‡	54.1±2.0	11.9±1.7	8.5±1.9	45.4±1.9	37.3±1.5	5.2±1.9	3.2±0.7	38.1±2.4	51.6±2.5	10.2±1.9	9.2±2.3	46.5±2.0
DCGLC‡	56.8±2.3	14.7±2.0	14.1±2.3	51.4±2.6	40.5±1.9	6.2±1.4	4.1±0.8	42.1±2.5	54.4±2.2	11.1±1.5	12.6±2.5	49.8±1.7
GPC‡	59.7±2.6	16.1±2.9	11.7±2.8	49.8±3.0	43.8±1.8	11.1±0.8	3.7±0.7	39.0±1.6	60.6±1.1	8.8±1.7	13.7±1.4	51.0±1.5
FedGCN	69.1±1.2	19.9±2.2	25.8±4.3	59.7±2.0	45.8±2.3	14.1±4.0	9.2±2.2	38.0±2.6	63.0±0.9	12.0±3.5	14.5±4.0	52.6±1.3
FedPKA	70.8±0.4	23.3±4.8	26.2±4.2	60.6±2.6	46.9±1.5	12.1±1.6	8.9±1.0	42.1±1.7	65.1±1.9	13.4±3.4	16.5±4.3	54.5±2.8
FGCN-AKC	75.2±0.8	26.0±1.5	31.5±1.3	68.0±1.5	50.4±0.5	17.0±1.2	10.9±0.6	47.8±2.1	70.1±2.9	16.3±3.1	19.4±2.6	57.6±2.3
FedAI	72.3±0.7	23.4±1.4	27.6±1.8	63.5±1.5	46.6±1.3	10.4±0.6	7.0±0.7	40.5±1.9	67.3±2.3	17.4±5.3	19.5±3.1	56.1±5.9
Perfect	70.7±2.0	21.8±6.3	27.0±5.9	63.3±3.4	47.0±1.0	16.3±1.2	9.7±0.7	39.0±1.6	64.8±1.3	15.4±4.3	18.1±4.2	56.5±4.2
FedCND	70.0±0.7	21.1±0.6	27.0±0.8	61.2±1.1	46.6±1.4	14.5±1.6	9.6±1.0	40.2±1.9	65.0±1.8	13.1±2.2	15.1±2.3	54.8±3.0
Ours	**75.5±0.4**	**36.5±1.1**	**41.3±0.9**	**73.4±0.4**	**51.7±0.4**	**17.4±0.5**	**11.3±0.5**	**49.0±3.3**	**75.0±0.3**	**30.8±1.0**	**35.2±0.1**	**73.5±0.1**
	SM-BIO-SY^2(10)				SN-SY^4(3)				CV^{15}(3)			
FedSage*	47.2±2.5	14.8±2.5	12.0±3.1	37.4±2.4	12.8±1.5	5.5±1.4	2.3±3.5	2.3±2.4	12.5±0.9	14.3±0.6	5.8±1.2	12.9±1.1
GCFL*	49.1±2.6	10.5±2.9	9.5±3.6	42.4±2.5	16.0±0.9	3.3±3.0	0.8±1.5	7.0±1.3	18.1±1.7	17.6±2.5	6.1±1.7	19.6±1.6
FedStar*	48.6±3.4	11.6±3.1	11.3±3.9	42.9±2.9	16.0±3.7	3.0±3.2	1.6±3.2	6.5±2.9	14.9±1.1	13.2±2.0	4.9±1.9	14.6±3.6
LG-FGAD†	50.2±0.8	5.9±0.7	4.6±1.2	45.9±1.1	16.4±1.9	5.2±0.9	2.4±1.4	6.2±1.5	18.4±1.7	21.5±4.5	4.7±2.8	18.2±3.7
FGAD†	54.1±1.7	11.5±3.4	2.2±3.7	48.2±1.2	14.3±0.8	5.8±0.8	2.0±0.6	7.1±1.1	17.6±1.2	22.1±1.5	3.6±1.0	18.8±1.3
AGDiff†	54.0±1.8	12.5±3.6	10.0±1.9	43.3±3.9	13.9±2.6	3.4±2.6	2.5±0.4	7.1±3.6	15.8±1.5	21.2±1.6	4.7±0.9	14.9±1.7
UDGC‡	48.9±3.7	10.4±3.1	8.9±3.4	46.5±2.9	15.4±1.7	6.6±2.9	5.0±4.5	8.3±3.7	14.0±2.3	7.5±2.6	4.3±1.5	11.1±2.1
DGLC‡	51.4±2.6	12.0±1.8	8.5±2.2	47.2±1.7	16.2±1.4	7.6±1.9	5.9±1.5	7.5±1.8	20.5±2.4	15.7±1.6	7.8±1.3	16.9±2.0
DCGLC‡	54.1±1.9	13.1±2.3	10.5±1.7	52.5±2.4	15.8±1.7	6.9±1.9	5.2±2.6	9.2±3.4	20.2±2.0	17.9±1.3	10.1±1.2	19.0±1.6
GPC‡	60.6±3.0	15.9±2.5	13.1±2.4	53.4±2.8	17.4±2.3	8.8±1.4	4.4±1.8	8.2±2.4	22.4±2.2	17.8±2.1	9.5±1.6	20.6±2.3
FedGCN	62.4±1.8	11.6±2.8	14.3±4.6	52.9±4.8	16.7±4.0	4.1±6.3	1.9±3.4	10.0±4.5	24.5±2.8	26.2±2.7	10.8±2.0	24.7±3.3
FedPKA	64.5±1.3	15.0±1.2	18.6±1.5	55.4±3.0	15.1±3.4	5.0±3.0	5.0±2.6	7.4±3.2	26.1±1.2	26.2±1.9	11.6±1.2	26.6±1.6
FGCN-AKC	69.9±1.9	20.9±1.7	21.4±2.5	60.5±2.3	27.7±2.4	12.1±1.9	8.1±1.7	17.2±2.0	29.0±1.4	29.9±1.8	15.1±1.2	29.2±1.7
FedAI	66.5±1.5	18.4±2.5	18.7±3.4	55.1±1.8	18.9±1.1	5.9±0.6	3.3±0.9	18.4±1.2	24.1±1.5	25.5±1.9	10.8±1.5	24.8±1.7
Perfect	64.7±0.8	14.5±3.5	17.3±3.5	55.4±3.1	40.8±4.7	3.3±2.0	1.0±2.6	26.9±6.7	25.7±1.5	26.9±1.0	11.9±1.2	26.9±1.6
FedCND	64.5±2.2	14.5±1.4	15.2±1.8	54.4±2.9	20.1±7.2	7.4±1.3	2.0±0.8	15.5±5.1	25.5±1.3	27.2±1.9	12.1±0.6	25.5±0.8
Ours	**74.4±0.0**	**29.8±1.0**	**34.1±0.5**	**73.0±0.2**	**54.2±2.6**	**17.5±0.8**	**12.4±1.6**	**52.0±3.4**	**29.7±0.4**	**30.7±0.6**	**15.5±0.7**	**29.9±0.3**

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Benchmark Datasets. We evaluate FedHC on 15 public graph-level benchmarks from TUDataset Morris et al. (2020). Each dataset is treated as one federated client, so raw graphs are never centralized. Following recent federated graph-level clustering protocols Liu et al. (2025a); Wu et al. (2025); Han et al. (2025); Liu et al. (2026a), we build six non-IID settings: SM^2(7), SN^3(2), SM-BIO^2(9), SM-BIO-SY^2(10), SN-SY^4(3), and CV^{15}(3). These settings cover same-domain and cross-domain federations over molecular, bioinformatics, social-network, synthetic, and computer-vision graphs, reflecting both semantic and structural heterogeneity across clients. Dataset statistics and setting memberships are given in Appendix Tables 4 and 5.

Structure Missing Settings. To evaluate client-dependent structure missing, we corrupt only local graph structures before training while keeping node attributes and labels unchanged. For each

graph, undirected edges are randomly removed by a truncated Gaussian missing-rate protocol with rates clipped to $[20\%, 80\%]$, global mean 50%, client-level standard deviation 0.15, and graph-level standard deviation 0.08. All compared methods use the same client partitions, corrupted graphs, and evaluation protocol whenever applicable, so the comparison focuses on robustness to missing structures under identical missing patterns. More details are in Appendix E.

Baseline Methods. We compare FedHC with four groups of representative baselines: supervised federated graph learning methods adapted to clustering, federated graph anomaly detection methods adapted to clustering, centralized graph-level clustering methods adapted to federated learning, and recent federated graph-level clustering methods. For supervised baselines, ground-truth labels are removed during training and used only for post-hoc evaluation. This gives a broad comparison against model-aggregation, prototype-sharing, personalized, anomaly-oriented, and centralized clustering strategies under the same unsupervised protocol. Details are provided in Appendix F.

Implementation Details. FedHC uses a three-layer GIN encoder with hidden dimension 64. Unless otherwise specified, each experiment runs for 20 communication rounds, and each client performs 5 local epochs per round with mini-batch training. FedHC constructs local hypergraph recovery contexts on clients and server-side prototype hypergraphs from uploaded context prototypes, while raw graphs remain local throughout training. We report mean and standard deviation over two seeds, with more implementation details in Appendix G.

Evaluation Metrics. Following graph-level clustering and federated graph clustering protocols Cai et al. (2024a;b); Liu et al. (2025a), we report clustering accuracy (ACC), normalized mutual information (NMI) Strehl & Ghosh (2002), adjusted Rand index (ARI) Hubert & Arabie (1985), and macro-F1. ACC is computed after optimal cluster-label matching. Labels are used only for evaluation, and higher values indicate better clustering quality for all metrics. Metric definitions are in Appendix H.

For each federated setting, we evaluate each client locally and report the graph-count-weighted average across clients. The same random seeds determine the edge-missing masks and training runs for FedHC and applicable baselines. Hyperparameters are selected from the unsupervised training protocol, and labels are not used for model selection.

3.2 Comparison Experiments

We compare FedHC with representative baselines under the six structure-missing non-IID settings. As shown in Table 1, FedHC obtains the strongest overall results across the reported metrics. Supervised FGL baselines degrade after labels are removed, and anomaly detection methods are less suited to compact and separable clustering objectives. Centralized DGC methods adapted by local training and aggregation also struggle to align cluster semantics across private heterogeneous clients. Compared with existing FGC methods that mainly rely on isolated embeddings, single prototypes, or pairwise affinities, FedHC models higher-order hypergraph contexts on both clients and the server, providing more reliable cluster guidance when local graph structures are missing. The NMI and ARI gains suggest that FedHC improves cluster partition quality across clients.

Table 2: Ablation of federated collaboration strategies on representative settings.

Settings	Strategy	ACC	NMI	ARI	F1	avg.Δ
SM	FedAvg	66.8±1.6	26.9±2.4	30.4±2.0	64.2±1.9	-9.6
	FedProx	68.1±2.1	25.7±2.7	29.2±2.5	65.8±2.2	-9.5
	FedPer	69.4±1.8	28.3±2.1	32.1±1.8	66.4±1.7	-7.6
	FedNova	64.7±1.5	24.6±1.8	27.5±1.6	62.1±1.4	-11.9
	Ours	**75.5±0.4**	**36.5±1.1**	**41.3±0.9**	**73.4±0.4**	-
SM-BIO	FedAvg	64.2±2.0	18.5±2.7	20.8±2.4	61.3±2.1	-12.4
	FedProx	65.0±2.4	17.9±2.9	19.6±2.6	62.5±2.5	-12.4
	FedPer	67.1±1.8	20.4±2.2	23.5±2.1	63.8±1.9	-9.9
	FedNova	63.6±1.7	21.7±2.4	24.2±2.0	60.9±1.8	-11.0
	Ours	**75.0±0.3**	**30.8±1.0**	**35.2±0.1**	**73.5±0.1**	-
SM-BIO-SY	FedAvg	62.8±1.6	17.4±2.3	20.5±2.0	60.2±1.8	-12.6
	FedProx	64.1±2.5	16.8±2.8	19.1±2.6	62.0±2.4	-12.3
	FedPer	66.0±2.0	19.5±2.1	22.7±1.9	63.4±2.0	-9.9
	FedNova	63.5±2.2	20.6±2.6	23.6±2.5	61.1±2.3	-10.6
	Ours	**74.4±0.0**	**29.8±1.0**	**34.1±0.5**	**73.0±0.2**	-

3.3 Federated Collaboration Strategy Ablation

To isolate the effect of the proposed hypergraph-context collaboration, we keep the same local encoder, clustering objective, and structure-missing protocol, and replace only the server-side collaboration strategy with FedAvg McMahan et al. (2017), FedProx Li et al. (2020), FedPer Arivazhagan et al. (2019), and FedNova Wang et al. (2020). Due to space limitations, Table 2 reports three representative settings in the main text, while the remaining SN, SN-SY, and CV settings are reported in Appendix Table 6. These generic aggregation strategies underperform FedHC. FedPer is usually the strongest among them because it preserves part of the client-specific model, but its fixed personalization split still cannot model which local cluster contexts are reliable or mutually compatible under structure missing. FedHC instead aggregates higher-order local contexts on the server, yielding stable gains under both same-domain and cross-domain heterogeneity.

3.4 Module Ablation

We further conduct module ablation to verify the contribution of the two core components in FedHC. LHC denotes the Local Hypergraph Context module, which builds reliability-aware graph-instance hyperedges and provides local recovery contexts before clustering. GPH denotes the Global Prototype Hypergraph module, which organizes uploaded recovery-context prototypes on the server and returns global hypergraph priors for the next refinement round. Following the same protocol as the collaboration-strategy ablation, we keep the backbone encoder, structure-missing protocol, client partitions, and clustering objective unchanged, and compare FedHC with three variants: removing both LHC and GPH, removing GPH, and removing LHC. As shown in Figure 3, removing both LHC and GPH leads to the weakest ACC because the model loses both local group-wise structural context and server-side global-prior feedback. The two single-removal variants show setting-dependent degradation, while the full FedHC consistently achieves the best ACC. Additional NMI, ARI, and F1 results are reported in Appendix Figure 7.

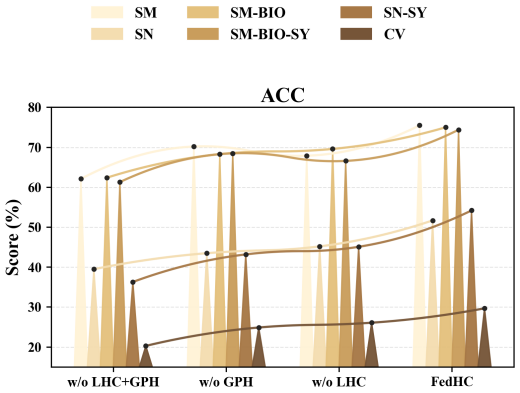


Figure 3: LHC/GPH ablation on ACC.

3.5 Convergence Analysis

We analyze the convergence of FedHC by tracking ACC, NMI, ARI, and F1 over communication rounds under the same structure-missing protocol. Figure 4 reports the representative SM setting in the main text, and the remaining settings are provided in Appendix Figure 8. The four metrics show a steady upward trend and then converge to stable values, indicating that the local hypergraph contexts and server-side prototype hypergraphs provide effective refinement guidance from the initial communication stage. The stable final trends across metrics indicate that FedHC improves both label assignment accuracy and cluster partition quality during federated structure-aware refinement.

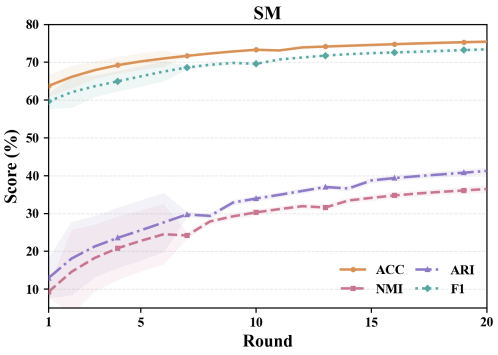


Figure 4: FedHC convergence on SM.

3.6 Hyperparameter Analysis

FedHC mainly introduces two types of additional hyperparameters: the local and server hyperedge sizes, and the two loss weights λ_{hc} and λ_{anc}. We analyze the loss weights by varying their common

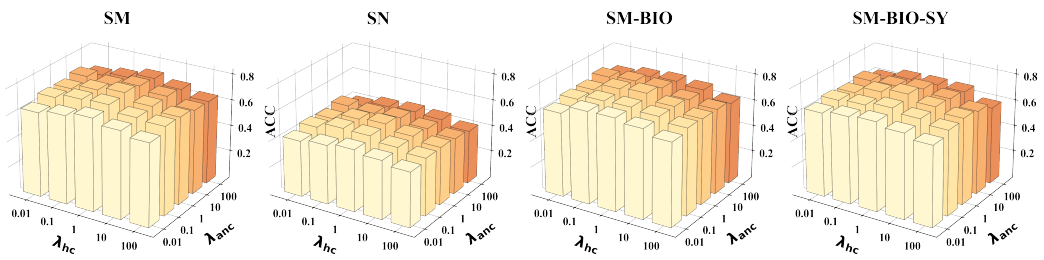


Figure 5: Hyperparameter sensitivity on ACC for four representative settings.

scale under the balanced ratio $\lambda_{\mathrm{hc}} : \lambda_{\mathrm{anc}} = 1 : 1$, while keeping the remaining settings unchanged. Figure 5 reports four representative settings, and Appendix Figure 10 provides SN-SY and CV results. The balanced 1:1 configuration achieves the best ACC in most cases, and nearby common-scale settings usually produce only small gaps. Moderate local hyperedge sizes provide the best balance between robust context construction and semantic purity; overly small hyperedges collapse to pairwise or prototype-like evidence, whereas overly large hyperedges may mix different latent clusters under severe structure missing.

3.7 Communication Overhead Analysis

We further evaluate communication-performance overhead under the same federated protocol. Figure 6 reports the representative SM setting, and the remaining settings are provided in Appendix Figure 9. FedHC uploads only compact context prototypes and reliability scores and downloads global hypergraph priors, so its payload per round is much smaller than aggregation strategies that exchange full or partial model parameters. Although the server needs an extra prototype-hypergraph construction step, this computation is performed over a small set of uploaded contexts and introduces only modest runtime and memory overhead. We report both payload and server-side resource usage because context-guided refinement should remain practical when the number of clients increases. This supports the compact communication analysis in Appendix C.

SM

	Payload/Round	Total Payload	Server Time	Peak RSS
FedAvg	1.84 MB	36.9 MB	6.1 ms	159 MiB
FedProx	1.89 MB	38.0 MB	6.6 ms	169 MiB
FedPer	1.31 MB	26.1 MB	5.5 ms	152 MiB
FedNova	1.98 MB	39.3 MB	7.4 ms	175 MiB
FedHC	12.1 KB	0.2 MB	9.1 ms	145 MiB

Figure 6: Communication overhead on SM.

4 Conclusion

In this paper, we study federated graph-level clustering with structure missing, where incomplete graph structures can lead to unreliable local representations and biased client prototypes. We propose FedHC, a federated graph clustering method that uses hypergraph contexts to reduce structure-missing bias. On each client, FedHC first constructs a graph-instance hypergraph from locally observed graphs and uses the resulting local hypergraph contexts to refine structure-aware graph representations. Instead of uploading private graphs or graph neighborhoods, clients only send compact context prototypes to the server. The server then builds a prototype hypergraph to integrate related contexts across clients and returns global hypergraph priors to further guide local refinement and clustering. Through this local-global design, FedHC exploits both client-specific structural patterns and cross-client shared structural information while preserving graph privacy. Experiments under different structure missing settings show that FedHC improves clustering performance over representative federated graph clustering baselines. These results support hypergraph-context refinement as a useful direction for federated graph clustering when the observed graph structures are incomplete.

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

Nimesh Agrawal, Anuj Kumar Sirohi, Sandeep Kumar, et al. No prejudice! fair federated graph neural networks for personalized recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 10775–10783, 2024.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya K Singh, and Sunpreet S Arora. 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 2020*, 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, Zhoumin Lu, Wenzhong Guo, and See-kiong Ng. FGad: Self-boosted Knowledge Distillation for An Effective Federated Graph Anomaly Detection Framework. *ArXiv:2402.12761*, 2024d.

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, 2025.

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.

Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W Tsang. Learning on Attribute-missing Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (2):740–757, 2020.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022.

Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. S3gc: Scalable self-supervised graph clustering. *Advances in Neural Information Processing Systems*, 35:3248–3261, 2022.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.

Xinyu Fu and Irwin King. Fedhgn: a federated framework for heterogeneous graph neural networks. *arXiv preprint arXiv:2305.09729*, 2023.

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.

Mengling Hu, Chaochao Chen, Weiming Liu, Xinyi Zhang, Xinting Liao, and Xiaolin Zheng. Learning Uniform Clusters on Hypersphere for Deep Graph-level Clustering. *ArXiv E-prints*, pp. arXiv–2311, 2023.

Tianyu Hu, Renda Han, Liu Mao, Jing Chen, and Xia Xie. Structure-Missing Graph-Level Clustering Network. *Neural Networks*, pp. 108682, 2026.

Lawrence Hubert and Phipps Arabie. Comparing Partitions. *Journal of Classification*, 2:193–218, 1985.

Thomas N Kipf and Max Welling. Variational Graph Auto-encoders. *ArXiv:1611.07308*, 1050, 2016.

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.

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.

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

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, 2025c.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of Benchmark Datasets for Learning with Graphs. *ArXiv Preprint ArXiv:2007.08663*, 2020.

Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12: 2539–2561, 2011.

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*, pp. 9953–9961, 2023.

Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep Fusion Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9978–9987, 2021.

Wenxuan Tu, Renxiang Guan, Sihang Zhou, Chuan Ma, Xin Peng, Zhiping Cai, Zhe Liu, Jieren Cheng, and Xinwang Liu. Attribute-Missing Graph Clustering Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 15392–15401, 2024.

Guancheng Wan, Wenke Huang, and Mang Ye. Federated graph learning under domain shift with generalizable prototypes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15429–15437, 2024.

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.

Xiaobao Wang, Meng Ge, Lingshan Li, Di Jin, Kai He, and Erik Cambria. Global Entity Relationship Enhancement Network for Multimodal Sarcasm Detection. *IEEE Transactions on Affective Computing*, pp. 1–14, 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 ACM Web Conference*, pp. 925–934, 2026.

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.

Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling Missing Data with Graph Representation Learning. *Advances in Neural Information Processing Systems*, 33:19075–19087, 2020.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, 2006.

Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. A survey on graph structure learning: Progress and opportunities. *arXiv preprint arXiv:2103.03036*, 2021.

Shuman Zhuang, Zhihao Wu, Wei Huang, Luojun Lin, Jia-Li Yin, Lele Fu, and Hong-Ning Dai. Prior refinement is better: Diffusion-driven graph harmonization for federated graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29241–29250, 2026.

A Additional Related Work

A.1 Federated Graph-Level Clustering and Related Baselines

Deep graph clustering is the starting point for graph-level clustering in both centralized and federated settings. SDCN Bo et al. (2020) combines graph structure reconstruction with deep clustering refinement. DFCN Tu et al. (2021) studies feature-structure fusion for graph clustering. S3GC Devvrit et al. (2022) considers scalable self-supervised graph clustering. For graph-level data, DGLC Cai et al. (2024a) learns graph representations through pseudo-label-guided mutual information maximization, DCGLC Cai et al. (2024b) aligns graph clusters from several contrastive views, and GPC Chen et al. (2025) studies prompt-based graph clustering. These methods give competitive centralized clustering objectives. Yet they usually assume that all graph instances are available on one machine, so they do not fit cases where graph collections are split across clients and raw graphs cannot be exchanged.

Federated graph learning studies this decentralized case. GCFL Xie et al. (2021) groups clients with similar training behavior for federated graph classification. FedStar Tan et al. (2023) shares structural knowledge across graph clients while leaving local feature patterns on each client. Prototype-based federated graph learning has also been used to handle graph domain shift Wan et al. (2024). These studies show that graph clients need graph-aware collaboration, because their topology, features, and semantics may differ. However, most of them are supervised methods, while federated graph-level clustering must infer cluster assignments without labels.

Recent FGC methods address this unsupervised federated setting from different angles. FedGCN Liu et al. (2025a) shares local cluster prototypes and estimates global consensus prototypes at the server. FedPKA Wu et al. (2025) uses personalized knowledge aggregation so that each client can receive knowledge from more related clients. PERFECT Liu et al. (2026a) further studies personalized client identification and aggregation under heterogeneous graph distributions. FGCN-AKC Han et al. (2025) uses adaptive knowledge compensation to reduce the mismatch among local clustering patterns. FedAI Han et al. (2026) focuses on missing attributes and uses attribute inference to support graph-level clustering. FedCND Wu et al. (2026) studies the case where different clients have different cluster numbers. These methods cover non-IID data, personalized aggregation, knowledge drift, missing attributes, and cluster-number mismatch.

Some methods outside FGC can also be adapted as baselines. Federated graph anomaly detection methods such as LG-FGAD Cai et al. (2024c), FGAD Cai et al. (2024d), and AGDiff Cai et al. (2025) learn graph representations for abnormal-pattern detection; after removing the anomaly-specific decision layer, their representations can be clustered for comparison. Supervised federated graph methods can be adapted in a similar way by removing labels during training and clustering their learned graph representations after training. Compared with these baselines, FedHC studies a different failure source: the graph structure observed by each client may be damaged. Existing FGC methods mainly exchange graph embeddings, model parameters, cluster prototypes, pairwise affinities, or personalized aggregation weights. FedHC instead exchanges hypergraph context prototypes, where each prototype is built from a group of related structure-missing graphs and is later used as evidence for structure-aware refinement.

A.2 Structure Missing and Hypergraph Context Modeling

Missing information in graph learning is often studied as missing node attributes. Attribute-missing graph learning methods estimate hidden features or learn representations from observed topology and partial attributes Chen et al. (2020); You et al. (2020); Tu et al. (2024). In federated graph clustering, FedAI Han et al. (2026) handles missing attributes through attribute inference, and IfedGC Liu et al. (2026b) uses causal-aware attribute completion. These methods are related because incomplete information can distort clustering. Still, attribute missing and structure missing are not the same problem. Attribute missing changes feature observations, while structure missing changes edges, neighborhoods, connectivity, density, walks, and graph-level structural statistics.

Recent centralized graph-level clustering has begun to study structure missing Hu et al. (2026). In the centralized setting, all damaged graphs are available together, so the method can estimate common structural patterns from the full graph collection. In federated graph-level clustering, each client only observes local graphs, and different clients may have different missing rates because of collection

policies, observation noise, or limited coverage. A prototype uploaded by one client can therefore mix cluster semantics with client-specific structure-missing bias. Directly clustering damaged graphs, or directly averaging such prototypes, can then pass local structure errors to the server and back to clients.

Hypergraphs are well suited to represent group relations. Hypergraph neural networks Feng et al. (2019) and multiset-function-based hypergraph models Chien et al. (2022) model relations among a set of nodes instead of only two nodes. Federated hypergraph and heterogeneous graph learning methods also study higher-order or schema-aware collaboration in decentralized graph systems Fu & King (2023). These works motivate using higher-order structures when pairwise relations are too weak to describe the data. FedHC uses this idea in a graph-level clustering setting: the hypergraph is not the raw input graph, but an auxiliary context structure built over graph instances and uploaded prototypes.

FedHC follows the path described in the main paper: structure missing, hypergraph contexts, structure-aware refinement, and graph clustering. On each client, the graph-instance hypergraph groups graphs with similar representations and observed structural patterns. Each hyperedge summarizes a local recovery context, which gives a reference for refining structure-aware graph representations before clustering. The client does not upload private graphs or graph neighborhoods; it uploads compact hypergraph context prototypes. The server then builds a prototype hypergraph over these uploaded contexts. Each server hyperedge connects related recovery contexts from clients with different missing patterns and forms a global hypergraph prior. This prior is returned to clients to guide the next round of local refinement and clustering. In this way, FedHC differs from prototype averaging and pairwise affinity sharing: it treats a group of graphs, and later a group of local recovery contexts, as the unit of structural evidence.

B NOTATIONS

Table 3 summarizes the main symbols used in FedHC. The notation distinguishes private client-side graph observations, structure-missing inputs, local recovery contexts, refined structures, uploaded context prototypes, global structural priors, and the loss terms that couple local recovery with global feedback.

Table 3: Summary of key notations in FedHC.

Symbol	Description	Symbol	Description
M	Number of clients	N_m	Number of graphs on client m
G_{mi}	Private graph instance	$\widetilde{G}_{mi}$	Structure-missing graph observation
$\mathbf{X}_{mi}$	Node feature matrix	$\widetilde{\mathbf{A}}_{mi}$	Observed adjacency matrix
$\widehat{\mathbf{A}}_{mi}$	Effective refined structure	$\mathrm{Refine}(\cdot)$	Structure-aware refinement operator
f_{θ_m}	Local graph encoder	$\widetilde{\mathbf{z}}_{mi}$	Auxiliary graph descriptor
$\mathbf{z}_{mi}$	Refined graph representation	$\mathbf{s}_{mi}$	Structural summary
ρ_{mi}	Structure reliability score	$\mathbf{h}_{mi}$	Context feature for hyperedges
$\mathcal{H}_m$	Client graph-instance hypergraph	$\mathcal{E}_m$	Local hyperedge set
k	Local hyperedge size	k_s	Server hyperedge size
$\mathbf{c}_{me}$	Local recovery context	$\bar{\mathbf{c}}_{mi}$	Graph-specific recovery prior
$\mathcal{P}_m$	Uploaded context/cluster prototypes	$\mathbf{p}_{mr}$	Local context prototype
ω_{mr}	Prototype reliability score	$\mathcal{A}$	Global structural prior set
$\mathbf{a}_g$	Global structural prior	$\hat{y}_{mi}$	Predicted cluster assignment
$\mathcal{L}_{\mathrm{hc}}$	Hypergraph consistency loss	$\mathcal{L}_{\mathrm{anc}}$	Global-prior feedback loss
λ_{hc}	Local context loss weight	λ_{anc}	Global-prior feedback weight

C ADDITIONAL ANALYSIS OF FEDHC

FedHC is trained with neural graph encoders and clustering objectives, so the full optimization problem is non-convex. We do not claim global optimality. This section explains the parts that are

specific to FedHC: reliability-aware local recovery contexts, server-side global structural priors, and compact communication.

C.1 Reliability-Aware Local Recovery Contexts

Consider a local hyperedge e on client m. Let the auxiliary descriptor of graph i be a noisy observation

$$\widetilde{\mathbf{z}}_{mi} = \boldsymbol{\mu}_e + \boldsymbol{\epsilon}_{mi}, \tag{14}$$

where $\boldsymbol{\mu}_e$ is the latent recovery-context center and $\boldsymbol{\epsilon}_{mi}$ is a zero-mean perturbation caused by stochastic training and structure missing. FedHC estimates the local recovery context by

$$\mathbf{c}_{me} = \frac{\sum_{i \in e} \rho_{mi} \widetilde{\mathbf{z}}_{mi}}{\sum_{i \in e} \rho_{mi} + \epsilon}. \tag{15}$$

Ignoring the numerical constant ϵ for clarity, this estimator is unbiased when the perturbations are zero-mean:

$$\mathbb{E}[\mathbf{c}_{me}] = \boldsymbol{\mu}_e. \tag{16}$$

If the perturbations are independent with covariance bounded by $\sigma_i^2 I$, then

$$\mathrm{Var}(\mathbf{c}_{me}) \preceq \frac{\sum_{i \in e} \rho_{mi}^2 \sigma_i^2}{\left(\sum_{i \in e} \rho_{mi}\right)^2} I. \tag{17}$$

The expression shows why FedHC builds recovery contexts from groups of related graphs rather than from a single damaged graph. When lower reliability indicates heavier edge loss and larger perturbation, the factor ρ_{mi} also reduces the effect of highly corrupted graphs on the structural-context prior.

C.2 Global Structural Priors

The server receives recovery-context prototypes $\{(\mathbf{p}_u, \omega_u)\}_{u=1}^{R}$ and constructs prototype hyperedges. For a server hyperedge E_g, FedHC computes

$$\mathbf{a}_g = \frac{\sum_{u \in E_g} \omega_u \mathbf{p}_u}{\sum_{u \in E_g} \omega_u + \epsilon}. \tag{18}$$

This step plays the cross-client role of the local recovery-context estimator. Each uploaded prototype is a compact estimate of a client-side recovery context, and the prototype hypergraph groups contexts that point to similar structural patterns. Reliability-aware aggregation prevents global priors from being controlled by clients with severe structure missing. The returned prior then enters the client-side refinement step, either through an explicit structure-refinement operator or through context-guided regularization of graph representations. Thus, local training uses both client-specific context and cross-client structural information.

C.3 Communication and Privacy Boundary

FedHC does not transmit raw graphs, observed graph neighborhoods, graph-level adjacency matrices, candidate refined edges, or all sample embeddings. Client m uploads only R_m recovery-context prototypes and reliability scores. If the representation dimension is d_z, the additional upload cost is

$$\mathcal{O}(R_m d_z), \tag{19}$$

and the server-side hypergraph construction cost is

$$\mathcal{O}(R^2 d_z), \qquad R = \sum_{m=1}^{M} R_m. \tag{20}$$

Since R is the number of uploaded context prototypes rather than the number of private graphs, the server computation is performed on compact summaries. The privacy boundary is the same as in prototype-sharing federated clustering: private graph structures and structure-level information remain local, while only aggregated context-level representations are exchanged.

D Details of Benchmark Datasets

We use 15 graph-level datasets from TUDataset Morris et al. (2020). Each dataset is treated as one federated client, and the raw graphs are never exchanged among clients. The datasets cover five domains: small molecules (SM), bioinformatics (BIO), synthetic graphs (SY), social networks (SN), and computer vision (CV). Table 4 reports the number of graph labels, graphs, average nodes, and average undirected edges per graph.

Table 4: Statistics of graph-level benchmark datasets.

Dataset	Domain	Classes	Graphs	A.Nodes	A.Edges
MUTAG	SM	2	188	17.93	19.79
BZR	SM	2	405	35.75	38.36
COX2	SM	2	467	41.22	43.45
DHFR	SM	2	756	42.43	44.54
PTC_MR	SM	2	344	14.29	14.69
AIDS	SM	2	2000	15.69	16.20
BZR_MD	SM	2	306	21.30	225.05
DD	BIO	2	1178	284.32	715.66
PROTEINS	BIO	2	1113	39.06	72.82
SYNTHETIC	SY	2	300	100.00	196.00
COLLAB	SN	3	5000	74.49	2457.22
IMDB-MULTI	SN	3	1500	13.00	65.94
Letter-low	CV	15	2250	4.68	3.13
Letter-med	CV	15	2250	4.67	3.21
Letter-high	CV	15	2250	4.67	4.50

The original setting name SN-SY^4(3) is inherited from prior graph-level clustering protocols. In our local TUDataset files, SYNTHIE does not provide graph labels and therefore cannot be evaluated with ACC, NMI, ARI, or F1. We use the label-evaluable SYNTHETIC dataset for the synthetic client in SN-SY^4(3) and SM-BIO-SY^2(10).

E Details of Federated Settings

Each federated setting assigns one graph-level dataset to one client. The six settings in Table 5 cover same-domain and cross-domain federations with different numbers of classes and clients. Structure missing is applied locally after client construction, so no raw graph structure is shared across clients.

For all six settings, we evaluate client-dependent structure missing with a truncated Gaussian protocol. The missing-rate range is $[20\%, 80\%]$, the global mean is 50%, the client-level standard deviation is 0.15, and the graph-level standard deviation is 0.08. This protocol creates different structure-missing patterns across clients while keeping the evaluation protocol comparable across methods.

F Details of Baseline Methods

We compare FedHC with four groups of graph-level clustering baselines under the same client partitions and structure-missing protocol.

Supervised federated graph learning methods. FedSage, GCFL Xie et al. (2021), and FedStar Tan et al. (2023) are originally designed for supervised federated graph learning. To adapt them to clustering, we remove label supervision during training and use the learned graph representations for post-hoc clustering.

Federated graph anomaly detection methods. LG-FGAD Cai et al. (2024c), FGAD Cai et al. (2024d), and AGDiff Cai et al. (2025) are adapted by using their graph representation modules and converting the final anomaly-oriented output into clustering assignments.

Table 5: Federated graph-level settings and corresponding client datasets.

Dataset	SM^2(7)	SN^3(2)	SM-BIO^2(9)	SM-BIO-SY^2(10)	SN-SY^4(3)	CV^{15}(3)
MUTAG	✓		✓	✓		
BZR	✓		✓	✓		
COX2	✓		✓	✓		
DHFR	✓		✓	✓		
PTC_MR	✓		✓	✓		
AIDS	✓		✓	✓		
BZR_MD	✓		✓	✓		
DD			✓	✓		
PROTEINS			✓	✓		
SYNTHETIC				✓	✓	
COLLAB		✓			✓	
IMDB-MULTI		✓			✓	
Letter-low						✓
Letter-med						✓
Letter-high						✓

Centralized deep graph-level clustering methods. UDGC Hu et al. (2023), DGLC Cai et al. (2024a), DCGLC Cai et al. (2024b), and GPC Chen et al. (2025) are adapted to the federated setting by training locally on each client and aggregating the exchangeable representation or prototype information when applicable.

Federated graph-level clustering methods. FedGCN Liu et al. (2025a), FedPKA Wu et al. (2025), FGCN-AKC Han et al. (2025), FedAI Han et al. (2026), PERFECT Liu et al. (2026a), and FedCND Wu et al. (2026) are the closest baselines. They exchange prototypes, personalized knowledge, attribute-recovery information, or cluster-number-aware signals. FedHC differs by exchanging recovery-context prototypes and constructing a server-side prototype hypergraph that returns global structural priors.

All compared methods use the same available graph observations after local edge dropping. Ground-truth labels are used only for final ACC, NMI, ARI, and F1 evaluation.

F.1 Additional Federated Collaboration Strategy Ablation

We further report the collaboration-strategy ablation on the remaining SN, SN-SY, and CV settings. The protocol is the same as in the main text: only the server-side collaboration strategy is replaced, while the local encoder, clustering objective, client partitions, and structure-missing process remain unchanged. As shown in Table 6, FedHC gives higher scores than generic aggregation strategies, indicating that global structural priors from prototype hyperedges help beyond molecule and bioinformatics clients.

Table 6: Additional ablation results of various federated collaboration strategies under structure missing.

Settings	Strategy	ACC	NMI	ARI	F1	avg.Δ
SN	FedAvg	45.1±2.0	12.8±1.8	7.1±1.2	40.8±2.5	-5.9
	FedProx	44.3±2.3	11.6±2.0	6.4±1.5	39.2±2.7	-7.0
	FedPer	46.8±1.7	13.7±1.6	7.9±1.3	42.1±2.1	-4.7
	FedNova	43.5±1.8	10.9±1.7	6.1±1.2	40.0±2.0	-7.2
	Ours	**51.7±0.4**	**17.4±0.5**	**11.3±0.5**	**49.0±3.3**	-
SN-SY	FedAvg	43.0±2.4	11.6±1.9	6.9±1.5	38.4±2.8	-9.1
	FedProx	42.4±2.6	10.2±2.1	5.8±1.7	37.1±3.0	-10.2
	FedPer	45.6±2.2	12.8±1.6	7.6±1.4	41.2±2.5	-7.2
	FedNova	41.7±2.0	9.8±1.8	5.5±1.5	38.9±2.4	-10.1
	Ours	**54.2±2.6**	**17.5±0.8**	**12.4±1.6**	**52.0±3.4**	-
CV	FedAvg	24.0±1.8	25.8±2.1	11.7±1.4	24.2±1.7	-5.0
	FedProx	23.5±1.6	26.4±2.4	11.4±1.5	24.8±2.0	-4.9
	FedPer	25.1±1.9	27.2±1.8	12.3±1.3	25.7±1.9	-3.9
	FedNova	23.8±1.5	24.9±1.7	12.0±1.2	23.6±1.6	-5.4
	Ours	**29.7±0.4**	**30.7±0.6**	**15.5±0.7**	**29.9±0.3**	-

F.2 Additional Module Ablation Metrics

Figure 7 reports the remaining three metrics for the module ablation study in the

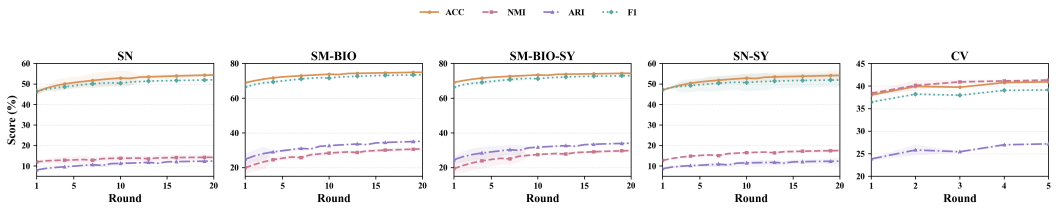


Figure 8: Additional FedHC convergence results.

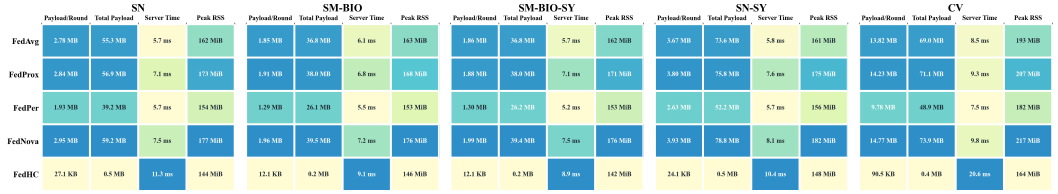


Figure 9: Additional communication overhead.

main text. The trends are consistent with the ACC results: removing both the Local Hypergraph Context module (LHC) and the Global Prototype Hypergraph module (GPH) produces the weakest performance, while removing only LHC or GPH leads to setting-dependent degradation. The complete FedHC model obtains the best NMI, ARI, and F1, showing that local recovery contexts and global structural priors both matter.

F.3 Additional Convergence Results

Figure 8 reports the FedHC convergence trends on the remaining SN, SM-BIO, SM-BIO-SY, SN-SY, and CV settings. Together with Figure 4 in the main text, the results cover all six structure-missing settings. ACC, NMI, ARI, and F1 generally rise during communication and converge to stable values, confirming that the proposed context-guided refinement mechanism remains stable across different federated graph domains.

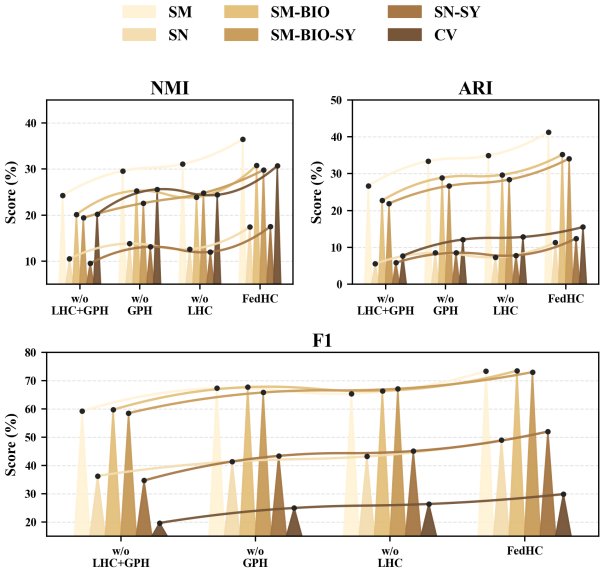


Figure 7: Additional LHC/GPH ablation metrics.

F.4 Additional Communication Overhead

Figure 9 reports the communication-performance overhead on the remaining five settings. Together with Figure 6, the results show that FedHC keeps the payload compact across different client scales because the exchanged objects are recovery-context prototypes, reliability scores, and global priors. The additional server-side cost increases mildly when more clients participate, but it remains concentrated on the compact prototype set.

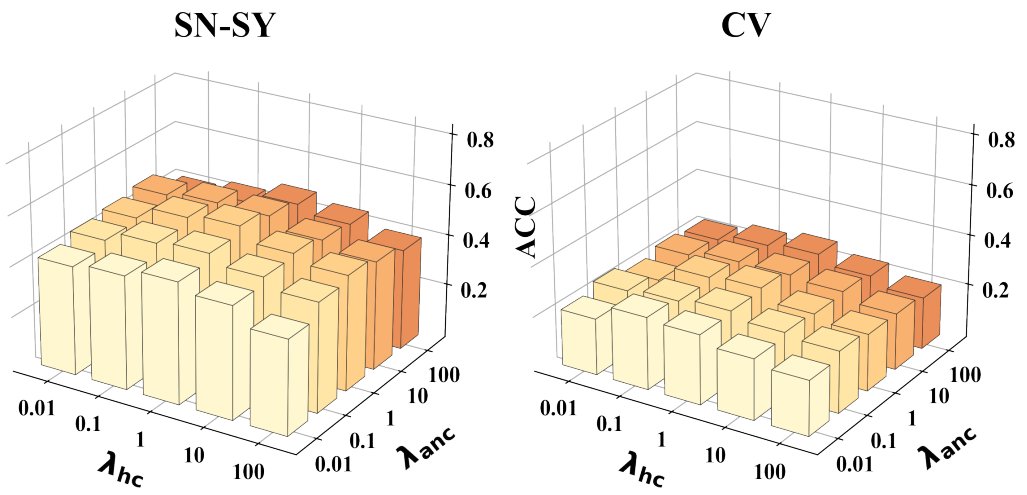


Figure 10: Additional hyperparameter sensitivity on ACC.

G More Implementation Details

Backbone and training. FedHC uses a three-layer GIN graph encoder with hidden dimension 64. Unless otherwise specified, the main experiments run for 20 communication rounds, and each client performs 5 local epochs per round. The mini-batch size is 128 for most graph-level datasets and 256 for large social-network settings. The optimizer is Adam, with learning rate selected from the same tuning protocol used for the main FedHC variants.

Structure missing protocol. For each local graph, undirected edges are randomly removed before training. We use the Gaussian structure-missing mode with missing rates clipped to $[0.2, 0.8]$. The global missing mean is 0.5, the client-level standard deviation is 0.15, and the graph-level standard deviation is 0.08. The realized edge-retention ratio is used as the structure reliability score when constructing local recovery contexts.

Hypergraph context parameters. FedHC uses local graph-instance hyperedges as recovery contexts and server-side prototype hyperedges as global structural priors. The main configurations tune the local hyperedge size from $\{5, 8\}$ and the server hyperedge size from $\{3, 4, 5\}$ depending on the setting. The number of uploaded local context prototypes is 8 by default and 12 for the tuned $SN^3(2)$ setting. The number of global priors is 16 by default and 24 for the tuned $SN^3(2)$ setting.

Loss weights. FedHC uses two loss weights: λ_{hc} for the local hypergraph consistency term and λ_{anc} for the returned global-prior feedback. Unless otherwise specified, we use a balanced 1:1 setting, i.e., $\lambda_{\mathrm{hc}} = \lambda_{\mathrm{anc}}$, and tune only their common scale. The feedback term is disabled before the first server update.

Setting-specific preprocessing. Large social-network settings use fast structural preprocessing, disable expensive NetworkX structural features, and set random-walk and spectral structural dimensions to zero. Large BIO-combined settings use fast preprocessing with reduced random-walk and spectral dimensions. For $SN^3(2)$, the reported stable setting uses structure-derived node features from the observed graph, including degree and related local statistics. The $CV^{15}(3)$ setting concatenates learned graph embeddings with standardized raw geometric descriptors from the original Letter coordinates, while excluding missing-edge structure summaries that add noise for these graphs.

Reporting. All main results are reported as mean and standard deviation over two seeds. The same corrupted graph observations and evaluation metrics are used across compared methods whenever applicable.

G.1 Additional Hyperparameter Sensitivity

Figure 10 reports the hyperparameter sensitivity results on SN-SY and CV. Together with Figure 5 in the main text, the results cover all six structure-missing settings. The remaining settings show the same pattern: moderate local context and global-prior weights work better than settings that make either term too weak or too large.

H 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:

$$\mathrm{ACC} = \max_{\pi} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(y_i = \pi(\hat{y}_i)\right), \tag{21}$$

  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:

$$\mathrm{NMI} = \frac{I(Y, \hat{Y})}{\sqrt{H(Y)H(\hat{Y})}}. \tag{22}$$

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

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]}. \tag{23}$$

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

$$\mathrm{F1} = \frac{1}{C} \sum_{c=1}^{C} \frac{2\,\mathrm{Precision}_c\,\mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}. \tag{24}$$

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

I Limitations

FedHC focuses on federated graph-level clustering with structure missing. Several limitations remain. First, FedHC relies on the quality of the local graph encoder and structural summaries; on social graph datasets with weak or absent node attributes, the representation backbone can still limit the final clustering quality. Second, the current implementation constructs hyperedges from embedding and structural-similarity neighborhoods, which may require approximate nearest-neighbor search for much larger clients. Third, our experiments simulate structure missing with a controlled truncated Gaussian protocol. In practical deployments, estimating structure reliability from observable data-quality signals needs further study. Finally, FedHC addresses missing edges through context-guided structural refinement rather than missing node attributes; combining this direction with attribute completion is left for future work.

J Broader Impacts

This work studies general graph representation learning and federated clustering without targeting a specific sensitive application. The method can help decentralized organizations analyze graph collections without sharing raw graphs or graph neighborhoods. As with other clustering methods, practitioners should avoid interpreting discovered clusters as ground-truth categories without domain validation, especially in high-stakes applications.