Dynamic Spectral Federated Graph-Level Clustering

Wenxin Zhang[1*], **Xi Xuan** [2,3*], **Guangzhen Yao** [4*], **Feng Zhou** [5†], **Cuicui Luo** [1†], **Renxiang Guan**[4], **Zonghao Ying**[6], **Xiaojian Lin**[7], **Yangchen Zeng**[8], **Renda Han**[9]
[1]University of Chinese Academy of Sciences, [2]University of Eastern Finland,
[3]University of Illinois at Urbana-Champaign, [4]National University of Defense Technology,
[5]Center for Applied Statistics and School of Statistics, Renmin University of China,
[6]Beihang University, [7]Tsinghua University, [8]Southeast University, [9]Tianjin University

Abstract

Federated graph clustering (FGC) aims to partition distributed graphs into different groups while preserving data privacy. Existing FGC methods focus on spatial-domain information aggregation, while how to use expressive spectral-domain signals of graphs for FGC still remains unexplored. Main challenges are twofold: (1) Collecting and integrating semantic signals that fully express the critical patterns of graphs; (2) Obtaining spectral consensus while preserving local properties of clients. To address these challenges, we propose dynamic spectral federated graph-level clustering (DySFGC). DySFGC designs learnable polynomial filters to capture abundant graph signals and employs spectral contrastive learning and reconstruction to explore the latent semantic information. Besides, DySFGC develops dynamic spectral consensus, which updates the consensus via frequency-specific channels and conducts the dynamic updating process by evaluating the discrepancies between the server and clients. Extensive experiments on fifteen datasets demonstrate the superiority of DySFGC over eleven baselines.

1 Introduction

Graph-level clustering aims to partition graphs into different groups by modeling discriminative graph representations, which is a fundamental task in graph machine learning and has been widely applied to many areas [1, 2, 3, 4]. Existing graph clustering methods mainly follow a centralized learning setting in which all graph samples are unconditionally accessible [5, 6, 7, 8]. However, in real-world applications, such as E-commerce networks [9, 10, 11], cyber-security [12, 13], and intelligent transportation [14], graph data is often gathered from different edge devices with the limitation of privacy requirements, data ownership, and communication constraints, instead of following centralized distribution.

To conquer this issue, federated graph clustering (FGC) has emerged. The FGC learning paradigm is composed of several distributed clients and an independent server. Each client performs local representation encoding and clustering on its own graph dataset while keeping the private graph data in each client unshared. Typically, the server will derive a consensus parameter setting by integrating the parameters uploaded from each client, and then feed back the consensus to each client to update their local parameters. Recent research [15, 16, 17] exhibits productive progress in FGC, which usually leverages spatial-domain encoding integrated with distribution learning to group the graph samples into different clusters. Despite their promising achievements, these approaches overlook the extraction and propagation of graph spectral signals within the graph topology. Spectral signals indicate the diffusion process of graph signal under complex topologies across different frequency

*Equal Contribution.
†Corresponding Authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

spectra, which contains rich semantic information and improves the identification of discriminative behavioral patterns [18, 19, 20, 21].

To fulfill this gap, there are two key challenges to be solved. First, although spectral graph learning has been widely studied, it shows incompetence under the FGC scenario. Specifically, the inaccessibility of label information in the unsupervised clustering deteriorates the difficulties of modeling expressive spectral graph signals that contain distinct category characteristics. Therefore, a natural question is posed: **(1) How can we collect discriminative semantic signals from the spectral domain and establish critical spectral patterns of graphs?** Second, it is difficult to model the consensus from the spectral domain since the graph signals with different frequencies may exhibit inconsistent spectral characteristics across clients, which may lead to the cross-client and cross-frequency distribution drift phenomenon during the signal diffusion process. This limitation impedes effective collaborative interactions among clients. Thus, a critical issue exists: **(2) How can we model the spectral consensus and balance it with private properties in distributed clients?** For the first challenge, we develop learnable polynomial filters to extract multi-frequency signals and design exclusive spectral objectives to enhance the discrimination of signals. For the second challenge, we design a novel dynamic spectral consensus module based on multi-level frequencies to capture more generalized consensus across different clients while preserving the private properties.

Based on the above analysis, we propose dynamic spectral federated graph-level clustering (DySFGC). The core of DySFGC is to explore rich spectral signals and dynamically execute frequency-specific parameter updating. Specifically, within each client, we design learnable Chebyshev polynomial filters to capture multi-frequency signals. To obtain more spectral semantics, we employ band-pass reconstruction and spectral contrastive learning to explore discriminative latent representation spaces for clustering. To capture the spectral consensus, we introduce a multi-channel updating mechanism and a dynamic updating mechanism to achieve adaptive interactions between clients and the server.

- **New research task:** To the best of our knowledge, we make the first attempt to tackle the spectral FGC problem, which develops powerful graph filters to capture multi-frequency graph signals and a novel semantic-aware contrastive learning to improve spectral clustering.
- **Novel FGC framework:** A dynamic spectral federated graph-level clustering (DySFGC) framework is proposed to address federated settings, which introduces a multi-channel updating mechanism for spectral consensus generation and introduces a client updating process integrated with traditional server updating.
- **Advanced performance:** Extensive experiments on fifteen graph datasets with five domains demonstrate the superiority of the proposed FedNCN over eleven state-of-the-art baselines.

2 Related Work

2.1 Graph-Level Clustering

In recent years, graph-level clustering has emerged as a dominant paradigm for grouping graph-structured data by jointly learning global representations and clustering assignments in an unsupervised manner. Prevalent solutions prefer to utilize reconstruction to capture the semantics in the graph. For example, SGEC [22] enhances the clustering robustness by reconstructing topological similarity. DBGCN [8] solves the noise problem in graph-level clustering by evaluating the contribution of global representations and measuring the similarity between graphs. Besides, some studies introduce contrastive learning into graph-level clustering to amplify the perception of consistencies among graphs. For example, AMGC [23] introduces adaptive mask strategies to avoid the noise interference and improve clustering robustness. SC-CGC [24] captures the consistencies of graphs via changing the neighbors with different hop numbers. In addition, a couple of approaches focus on multi-view graph clustering [25, 26, 27, 28], which can handle complex graphs with multiple relations. However, most graph-level clustering methods follow centralized learning paradigms, where graph data are unconditionally accessible, and the learning process can be completed by training a single model. This inevitably requires a high degree of data sharing and poses a risk of data leakage. To solve this limitation, we study FGC, which improves the clustering performance of each client by facilitating integration and interaction between model parameters while maintaining high-level data privacy.

2.2 Federated Graph Learning

Federated graph learning (FGL) enables distributed training across diverse clients with graph-structured data, which has broad research and application prospects [29]. The critical step is to achieve communication between clients and the server to jointly model consensus among distributed graph data. For example, FedVN [30] introduces the concept of virtual nodes to moderate distribution shifts across clients. FedGM [31] leverages graph condensation to extract global consensus from clients with different distributions. EaDA [32] extracts latent environmental information by deploying prototype learning and introduces local-global alignment to alleviate distribution shifts. FedIIH [33] explores the influence of intrinsic heterogeneity within clients on global consensus by hierarchical variational inference. S2FGL [34] explores the spectral semantics knowledge based on Laplacian eigenvectors, but it studies traditional graph learning tasks and relies on supervised signals. Recently, traditional supervised FGL has been extended to unsupervised graph clustering [35, 15]. For example, FedGCN [16] introduces topology-oriented feature encoding and prototype consensus optimization for FGC. [17] studies attribute-missing FGC by introducing. However, they only focus on the spatial information propagation and overlook the critical spectral graph signals that contain redundant semantic information about graph data.

3 Preliminaries

3.1 Notations

Consider a client set $\mathcal{C} = \{C_1, C_2, \cdots C_M\}$, where M is the number of clients. In FGC setting, each client C_m holds a private graph dataset $\mathcal{G}^m = \{\mathcal{G}_1^m, \mathcal{G}_2^m, \cdots \mathcal{G}_{N_m}^m\}$, where $N_m = |\mathcal{G}^m|$ is the number of graph samples. Denote $\mathcal{G}_i^m = \{\mathbf{X}_i^m, \mathbf{A}_i^m\}$ as i-th graph in client C_m. $\mathbf{X}_i \in \mathbb{R}^{B_i \times F}$ and $\mathbf{A}_i \in \{0,1\}^{B_i \times B_i}$, respectively, represent the node feature matrix and adjacency matrix, where B_i and F are respectively denote the number of nodes in $\mathcal{G}_i^m$ and the original feature dimension. We denote $\mathbf{Z}_o^m \in \mathbb{R}^{N_m \times F} = \{\mathbf{z}_1, \cdots, \mathbf{z}_{N_m}\}$ as the global representation of graphs in $\mathcal{G}^m$, where $\mathbf{z}_i$ is the global representation of $\mathcal{G}_i^m$. Owing to privacy constraints and non-IID characteristics, clients are not allowed to share graph data [36]. For each client C_m, we define a model f_{θ_m} with learnable parameters θ_m to encode the graph representations. The aim of FGC is to group the $\mathcal{G}^m$ in the client C_m into different clusters $\mathcal{P}^m = \{\mathcal{P}_1^m, \cdots, \mathcal{P}_T^m\}$ and $\sum_i^T |\mathcal{P}_i^m| = N_m$, where T is the the number of graph class. Without loss of generality, we omit the subscripts i and m.

3.2 Spectral Graph Filter

For a graph $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$, given the normalized graph Laplacian matrix $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}}$ is the normalized adjacency matrix and $\mathbf{I}$ is the identity matrix. $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_0, \cdots, \lambda_{B-1})$ is the diagonal matrix of eigenvalues and $\mathbf{U}$ is eigenvector matrix. Graph filtering function can be defined using K-order polynomial approximation as $g(\mathbf{\Lambda}) = \sum_k^K \omega_k \mathbf{L}_k$ for efficiency. Chebyshev polynomials is one of the most popular schemes [37], which can be defined as: $\Phi_K(\mathbf{L}) = \sum_k^K \omega_k \Gamma_k(\mathbf{L})$, where $\omega_k = \frac{1}{K+1}\sum_{j=0}^K \gamma_j \Gamma_k(\cos(\frac{j+1/2}{K+1}\pi))$ and γ_j is learnable parameters.

4 Methodology

The framework of DySFGC is shown in Fig. 1. In each client, DySFGC employs learnable spectral filters based on Chebyshev polynomials to capture low- and high-pass signals, and implement contrastive learning and reconstruction to exploit rich semantic information. For graph clustering, DySFGC utilizes the KL divergence to evaluate the similarity between the target distribution and the soft clustering distribution. In the server, DySFGC employs a multi-channel updating mechanism, which individually updates the parameters of corresponding learnable modules for high-, low-, and band-pass signals. Besides, DySFGC introduces a dynamic updating mechanism, which captures the spectral consensus while maintaining client-specific properties.

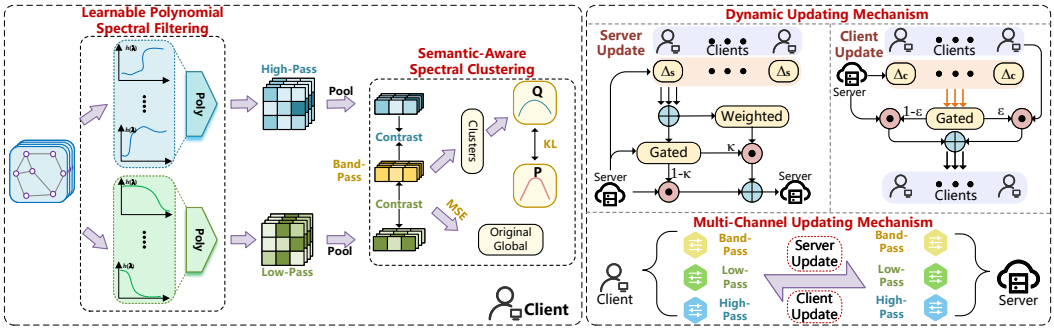


Figure 1: The framework of DySFGC is comprised of three components: learnable polynomial spectral filtering that designs spectral graph filters with learnable parameters based in Chebyshev polynomials to extract redundant high- and low-pass signals, semantic-aware spectral clustering that employs reconstruction and contrastive learning to learn discriminative spectral representations, and dynamic updating mechanism that designs multi-channel updating mechanism for frequency-aware parameters and utilizes the discrepancies between the server and clients to evaluate the weighted contributions of updating parameters. Three parts assemble into a unified framework.

4.1 Learnable Polynomial Spectral Filtering

The spectral domain contains redundant semantic signals that are highly related to the graph patterns. However, traditional graph spectral learning fails to exhibit competence in the unsupervised FGC scene since they lack semantic information about specific frequencies and is usually optimized towards convergence based on supervised label information. To obtain more discriminative spectral representations for clustering, we utilize Chebyshev polynomials [37] with learnable parameters as filter basis to capture low- and high-pass signals with significant semantic information about different frequencies. Wherein, low-pass signals mainly indicate the smoothing similarity between nodes, while high-pass signals help to reveal noise and edge information in the graph. Specifically, the low- and high-pass filters, Φ_K^L and Φ_K^H, can be represented as:

$$\Phi_K^L(\mathbf{L}) = \sum_k^K \omega_k^L \Gamma_k(\mathbf{L}), \Phi_K^H(\mathbf{L}) = \sum_k^K \omega_k^H \Gamma_k(\mathbf{L}). \tag{1}$$

As for the learnable parameter ω_k, DySFGC adopts a sliding Gaussian-parametric-based formulation to achieve the learnable filtering ability:

$$\gamma_j^L = 1 + \sum_{j=0}^K \exp(-\frac{\cos(j)}{2\sigma^2}), \gamma_j^H = 1 - \sum_{j=0}^K \exp(-\frac{\cos(j)}{2\sigma^2}). \tag{2}$$

where σ is a learnable parameter and are respectively initialized to 1. In the Gaussian-parametric model, DySFGC defaults the mean to 0 and leverages the cosine function to limit the sliding parameter j within [-1, 1] to control the filtering intensity. The exponential function ensures the monotonicity of the low- and high-pass filters. Due to $\exp(\cdot) \in (0, +\infty)$, $\gamma_0^L < \cdots < \gamma_K^L$ and $\gamma_0^H < \cdots < \gamma_K^H$, achieving low- and high-pass properties.

Formally, for each client, given a graph sample $\mathcal{G}$, DySFGC could obtain the encoded low- and high-pass signals:

$$\mathbf{H}_L = \Phi_K^L(\mathbf{L})\mathbf{X}, \quad \mathbf{H}_H = \Phi_K^H(\mathbf{L})\mathbf{X}. \tag{3}$$

where $\mathbf{H}_L, \mathbf{H}_H \in \mathbb{R}^{B \times F'}$ and F' is the latent dimension. DySFGC then uses the latent node representations to generate the global embeddings of the graph $\mathbf{h}_L, \mathbf{h}_H \in \mathbb{R}^{F'}$:

$$\mathbf{h}_L = \text{MeanPool}(\mathbf{H}_L), \quad \mathbf{h}_H = \text{MeanPool}(\mathbf{H}_H), \tag{4}$$

where $\text{MeanPool}(\cdot)$ is mean pooling operation. Next, DySFGC dynamically integrates high- and low-pass signals to generate band-pass signals and the corresponding global representations:

$$\begin{aligned} \mathbf{H}_B &= \text{MLP}_B(\alpha\mathbf{H}_L + (1-\alpha)\mathbf{H}_H), \\ \mathbf{h}_B &= \text{MeanPool}(\mathbf{H}_B). \end{aligned} \tag{5}$$

where $\alpha \in [0, 1]$ is a learnable parameter, $\mathbf{H}_B \in \mathbb{R}^{B \times F'}$ and $\mathbf{h}_B \in \mathbb{R}^{F'}$. Since the fixed parameters might be highly hypothetical and out of line with reality, a learnable α can not only cover signals in different frequency domains, but also fulfill the semantic information of the original features. Moreover, it is mainly optimized by reconstruction loss following the gradient flow of band-pass reconstruction, thereby preserving critical semantics in the original features. Finally, given the global representations of graphs, DySFGC can derive the global low-, high- and band-pass representations of graphs $\mathbf{Z}_L \in \mathbb{R}^{N \times F'} = [\mathbf{h}_L^1, \cdots, \mathbf{h}_L^N]$, $\mathbf{Z}_H \in \mathbb{R}^{N \times F'} = [\mathbf{h}_H^1, \cdots, \mathbf{h}_H^N]$ and $\mathbf{Z}_B \in \mathbb{R}^{N \times F'} = [\mathbf{h}_B^1, \cdots, \mathbf{h}_B^N]$ for each client.

In summary, the proposed learnable polynomial spectral filtering module ensures that the local filters in distributed clients capture more discriminative graph signals in different frequencies based on private datasets, fully extracting distinctive signals from every dataset.

4.2 Semantic-Aware Spectral Clustering

In each client, we hope to preserve the discriminative characteristics of signals in different frequencies while improving the semantics of global representations to improve the discriminative cluster-related expression. To enhance the distinct frequency semantics of signals, DySFGC first introduces the following spectral contrastive learning:

$$\begin{aligned} \mathcal{L}_L &= \mathbb{E}_{\mathcal{G}}[\log \mathcal{D}_L(\mathbf{Z}_L, \mathbf{Z}_B) + \log(1 - \mathcal{D}_L(\tilde{\mathbf{Z}}_L, \mathbf{Z}_B))], \\ \mathcal{L}_H &= \mathbb{E}_{\mathcal{G}}[\log \mathcal{D}_H(\mathbf{Z}_H, \mathbf{Z}_B) + \log(1 - \mathcal{D}_H(\tilde{\mathbf{Z}}_H, \mathbf{Z}_B))], \end{aligned} \tag{6}$$

where $\tilde{\mathbf{Z}}_H, \tilde{\mathbf{Z}}_L$ denote negative samples which is generated by random shuffling, and $\mathcal{D}_\phi(\mathbf{H}, \mathbf{h}) = \sigma(\mathbf{H}\mathbf{W}_\phi\mathbf{h}^\top), \phi \in \{L, H\}$ is a discriminator with learnable parameters $\mathbf{W} \in \mathbb{R}^{F' \times F'}$. Therefore, the spectral contrastive learning loss can be calculated as:

$$\mathcal{L}_{\text{con}} = (\mathcal{L}_L + \mathcal{L}_H) * 0.5. \tag{7}$$

We illustrate the theoretical intuition behind this loss. Although the fused band-pass representation is synthesized from these two components, using a non-linear mapping can create a distinct global view. By enforcing contrastive learning between the specific frequency views and the fused global view, we maximize the mutual information across different spectral channels because independent filters ensure the collaboration of frequency-specific signals instead of mutual conflict, thereby forcing the model to distill and retain the invariant structural semantics present in both the smooth topology and the localized structures, while discarding uninformative frequency-specific noise.

To obtain more expressive graph-level representations, DySFGC introduces a reconstruction with $\mathrm{MLP}_R(\cdot)$, which is widely applied in many fields [38, 39, 40, 41, 42, 43], to obtain global representations to preserve the critical semantics in the original graph and calculate the reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}||\mathrm{MLP}_R(\mathbf{Z}_B) - \mathbf{Z}_o||_2. \tag{8}$$

The reconstruction of band-pass signals is critical for clustering. It utilizes the band-pass representations that integrate frequency-specific signals to model the original semantic information of the original graphs, which avoids distortion of latent embeddings used for clustering.

To evaluate the clustering properties and boost the cluster-related expression of graph samples, based on deep learning, DySFGC employs a soft clustering matrix $\mathbf{Q} \in \mathbb{R}^{N \times t}$ to evaluate the similarity between global representations of $\mathcal{G}_i$ and the j-th clustering centers $\mathbf{c}_j$ based on Student's t-distribution:

$$q_{ij} = \frac{(1 + ||\mathbf{z}_B^i - \mathbf{c}_j||^2)^{-1}}{\sum_{t=1}^{T}(1 + ||\mathbf{z}_B^i - \mathbf{c}_t||^2)^{-1}}, \tag{9}$$

where $\mathbf{c}_j$ is initialized by K-Means clustering algorithm. Then, DySFGC obtains the target distribution $\mathbf{P} \in \mathbb{R}^{N \times t}$:

$$p_{ij} = \frac{(q_{ij})^2 / \sum_i q_{ij}}{\sum_{t=1}^{T}(q_{it})^2 / \sum_i q_{it}}. \tag{10}$$

Finally, DySFGC utilizes the KL-divergence $\mathcal{D}_{\mathrm{KL}}(\cdot||\cdot)$ to measure the clustering loss between the soft distribution and the target distribution:

$$\mathcal{L}_{\text{kl}} = \mathcal{D}_{\mathrm{KL}}(\mathbf{P}||\mathbf{Q}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{11}$$

The merits of the semantic-aware spectral clustering module lie in: (1) It effectively discriminates different semantic information about spectral signals using contrastive learning and reconstruction to precisely collect graph-specific characteristics; (2) It captures the category characteristics of graph samples to enhance the clustering via a KL divergence.

4.3 Dynamic Spectral Consensus

Different from traditional federated consensus generation [44, 16, 30], we introduce a dynamic spectral consensus generation strategy, which has two primary novel distinctions from existing methods. First, we introduce a multi-channel updating mechanism that separately updates the parameters of different frequencies. The motivation is that different frequency domains contain distinct semantic information about graphs, resulting in the discrepancy in potential knowledge learned by DySFGC. For example, high-pass singals is only captured by high-pass filters. This phenomenon encourages us to consider isolating the parameter update process through frequency-specific channels to obtain spectral consensus with more accurate semantics. Second, we introduce a client updating process integrated with the traditional server updating process. The intuition is that the existing updating mechanism directly downloads the consensus as the learnable parameters of distributed clients, which completely loses the private characteristics of local samples. However, we believe that the comprehensive global knowledge can improve the generalization of local models, but it cannot completely replace the local characteristics of distributed clients. Adaptive fusion of global and local knowledge can refine local generalization and alleviate cross-client heterogeneity.

Multi-Channel Updating Mechanism To capture more discriminative consensus in the spectral domain, DySFGC first implements a multi-channel updating mechanism. Specifically, the low-pass filter Φ_K^L and the low-pass discriminator $\mathcal{D}_L$ are exclusive for low-frequency encoding; their parameters are designed to follow the low-pass updating channel. Similarly, the parameters in the high-pass filter Φ_K^H and the high-pass discriminator $\mathcal{D}_H$ follow the high-pass updating channel. MLP_B and MLP_R follow a band-pass updating channel since their input is band-pass signals. Formally, for client C_m, we denote the learnable parameters in low-, high- and band-pass channel updating channel at t-th iteration ($t \geq 0$), respectively, as $\theta_L^m(t)$, $\theta_H^m(t)$, and $\theta_B^m(t)$. The corresponding global parameters in the server is $\theta_L^g(t)$, $\theta_H^g(t)$ and $\theta_B^g(t)$.

Dynamic Updating Mechanism We design a novel dynamic updating mechanism for different channels, including the server updating process and the client updating process, to address the second limitation.

(1) Server Updating Process. Take the low-pass channel as an example. To better balance the contribution of the local client to the server, we consider the difference of the parameters between client and server, which is denoted as $\Delta_s\theta_L^m(t) = \theta_L^m(t) - \theta_L^g(t)$. Meanwhile, to better evaluate the model learning ability, we consider preserving the original parameters in the server dynamically. The updating process can be defined as:

$$\theta_L^g(t+1) = (1-\kappa_L)\theta_L^g(t) + \kappa_L \sum_{C_m \in \mathcal{C}} \eta_L^m \Delta_s \theta_L^m(t). \tag{12}$$

Wherein, $\eta_L^m = \Phi(|\frac{\Delta_s\theta_L^m}{\sum_{C_i\in\mathcal{C}}\Delta_s\theta_L^i}|)$ is the aggregation coefficient, which aims to balance the contributions of different clients. $\Phi(\cdot)$ is mean function along all dimension. $\kappa_L = \frac{1}{1+\exp(-\Delta_L^o)}$ is a gated coefficient, where $\Delta_L^o = \Phi(\sum_{C_m\in\mathcal{C}}\Delta_s\theta_L^m(t) - \theta_L^g(t))$. It dynamically evaluates the contribution of the server's original global consensus in the updating process. Similarly, we can obtain the updating process for high- and band-pass channels:

$$\theta_H^g(t+1) = (1-\kappa_H)\theta_H^g(t) + \kappa_H \sum_{C_m \in \mathcal{C}} \eta_H^m \Delta_s \theta_H^m, \tag{13}$$

$$\theta_B^g(t+1) = (1-\kappa_B)\theta_B^g(t) + \kappa_B \sum_{C_m \in \mathcal{C}} \eta_B^m \Delta_s \theta_B^m, \tag{14}$$

where $\eta_H^m = \Phi(|\frac{\Delta_s\theta_H^m}{\sum_{C_i\in\mathcal{C}}\Delta_s\theta_H^i}|)$, $\eta_B^m = \Phi(|\frac{\Delta_s\theta_B^m}{\sum_{C_i\in\mathcal{C}}\Delta_s\theta_B^i}|)$ are aggregation coefficients respectively for high- and band-pass channels, $\kappa_H = \frac{1}{1+\exp(-\Delta_H^o)}$, $\kappa_B = \frac{1}{1+\exp(-\Delta_B^o)}$ are the gated coefficients

respectively for high- and band-pass channels, and $\Delta_H^o = \Phi(\frac{1}{M}\sum_{C_m \in \mathcal{C}} \Delta_s \theta_H^m - \theta_H^g(t))$ and $\Delta_B^o = \Phi(\frac{1}{M}\sum_{C_m \in \mathcal{C}} \Delta_s \theta_B^m - \theta_B^g(t))$.

(2) Client Updating Process. To better preserve the semantics of local clients, we set a gate coefficient for the local client when downloading the global parameters for the server. Take the low-pass channel as an example. Specifically, we denote the updated global parameters at t-th iteration as $\theta_L^g(t)$, and the parameters of local client C_m at $(t+1)$-th iteration as $\theta_L^m(t+1)$. We denote the difference between the updated global parameters and the local as $\Delta_c \theta_L^m = \theta_L^g(t+1) - \theta_L^m(t)$. The client updating process can be formulated as:

$$\theta_L^m(t+1) = (1-\epsilon_L^m)\theta_L^g(t+1) + \epsilon_L^m \theta_L^m(t), \tag{15}$$

where $\epsilon_L^m = \Phi(\frac{\Delta_c \theta_L^m}{\sum_{C_i \in \mathcal{C}} \Delta_c \theta_L^i})$ is the gate coefficient for clients. If the difference between the server and client C_m is relatively small, ϵ_L^m has a relatively small value, indicating that the updating process of client C_m focuses more on global consensus. However, if the global consensus shows less consistency with the local client, the update process will focus more on the client itself. In this way, it can maintain private properties while obtaining global consensus, effectively avoiding performance degradation caused by distribution shifts in the frequency domains. Similar to the low-pass channel, we can obtain the client updating process for high- and band-pass channels:

$$\theta_H^m(t+1) = (1-\epsilon_H^m)\theta_H^g(t) + \epsilon_H^m \theta_H^m(t), \tag{16}$$

$$\theta_B^m(t+1) = (1-\epsilon_B^m)\theta_B^g(t) + \epsilon_B^m \theta_B^m(t), \tag{17}$$

where $\epsilon_H^m = \Phi(\frac{\Delta_c \theta_H^m}{\sum_{C_i \in \mathcal{C}} \Delta_c \theta_H^i})$ and $\epsilon_B^m = \Phi(\frac{\Delta_c \theta_B^m}{\sum_{C_i \in \mathcal{C}} \Delta_c \theta_B^i})$.

The proposed dynamic spectral consensus module offers two benefits: (1) It introduces a frequency-specific updating mechanism to refine the learning process of model parameters according to different frequency characteristics; (2) The update of the server and clients can dynamically balance the mutual contributions and ensure the adaptive preservation of original semantics, which guarantees the effective synergy between private characteristics and global consensus.

4.4 Optimization Objective and Training

The overall loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{kl}} + \zeta_1 \mathcal{L}_{\text{con}} + \zeta_2 \mathcal{L}_{\text{rec}}, \tag{18}$$

where ζ_1, ζ_2 are hyperparameters. Section L shows a detailed complexity analysis and a detailed pseudocode of DySFGC.

5 Experiments

5.1 Experimental setup

Datasets and Baselines To verify the effectiveness of DySFGC, following [16], we employ 15 graph datasets across different domains [45], including: (1) SM (MU-TAG, BZR, COX2, DHFR, PTC_MP, AIDS, BZR_MD), SY (SYNTHETIC), SN (COLLAB, IMDB-MULTI), BIO (DD, PROTEINS), and CV (Letter-high, Letter-low, Letter-med). In the FGC problem, five types of non-IID settings are generated: (1) 2 clusters within the same domain (SM); (2) 3 clusters within the same domain (SN); (3) 15 clusters within the same domain (CV); (4) 2 clusters across two domains (SM-BIO); (5) 2 clusters across three domains (SM-BIO-SY). Details are in Section A. We select eleven state-of-the-art baselines, including three types: (1) Traditional graph clustering methods, DBGCN [8] and S2CAG [46]; (2) FGL methods, FedTAD [47], FedIIH [33], FedSPA [48], S2FGL [34], GuardFGL [49], FedGM [31], FedC4 [50], and FedHERO [51]; (3) A FGC method, FedGCN [16].

Implementation and Metrics The Adam optimizer is utilized, and the learning rate is 1e-3. The hidden size F' is set to 128. The training and communication epochs, T_c and T_s, are respectively set to 100 and 30. Hyperparameters ζ_1, ζ_2 are respectively set to 1 and 0.8. The filter order is set to 4. We evaluate DySFGC with four classical clustering metrics compared with all baselines [52, 53, 54]: Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and F1 Score (F1). More details about baseline and fairness are shown in Section B.

Table 1: The performance on five datasets under unsupervised settings. The results are in %.

Domain	Metric	DBGCN	S2CAG	FedTAD	FedIIH	FedSPA	S2FGL	GuardFGL	FedGM	FedC4	FedHERO	FedGCN	DySFGC
SM	ACC	58.9±2.2	61.4±3.1	45.3±2.6	57.6±1.8	62.1±2.4	52.1±2.0	68.5±1.5	58.3±2.3	49.1±2.7	60.8±1.9	75.9±2.1	**80.1**±0.9
	NMI	12.0±1.4	11.6±2.8	9.02±2.1	11.8±1.6	12.8±2.3	10.5±1.9	19.2±1.3	12.6±2.5	9.5±2.2	17.1±1.5	24.9±2.0	**30.9**±0.8
	ARI	0.1±1.9	4.9±2.4	2.5±1.6	5.2±2.1	13.8±2.5	8.9±1.7	30.3±2.2	15.9±1.8	2.7±2.3	26.2±1.4	31.1±3.2	**45.5**±0.7
	F1	49.7±2.5	55.2±1.8	38.9±3.1	48.8±2.2	57.1±1.6	44.5±2.4	61.2±1.4	47.5±2.7	39.5±2.0	57.9±1.9	67.1±2.3	**76.2**±1.1
SM-BIO	ACC	59.5±2.1	65.2±3.8	50.6±2.5	55.2±1.9	62.6±2.3	56.8±2.0	70.8±1.7	61.2±2.4	53.5±2.2	70.9±1.8	69.2±1.5	**78.5**±0.8
	NMI	5.3±1.9	6.8±2.5	4.9±2.1	5.1±1.6	7.8±2.4	6.6±1.8	13.5±2.2	10.5±1.5	6.4±1.7	10.5±2.0	14.0±2.9	**35.2**±0.7
	ARI	3.8±2.6	7.2±2.1	2.6±1.8	3.7±2.3	9.2±1.9	3.4±1.5	13.8±2.4	10.9±1.7	5.6±2.0	11.8±1.4	17.5±3.2	**38.7**±0.6
	F1	51.7±1.8	55.9±2.7	46.9±2.2	51.7±1.5	48.6±2.1	57.3±2.5	60.3±1.9	50.3±1.6	40.4±2.4	52.4±1.3	59.1±1.7	**66.1**±0.8
SM-BIO-SY	ACC	57.9±2.8	54.2±1.9	61.6±2.3	66.5±1.7	69.0±2.1	57.8±2.5	60.2±1.8	67.1±2.0	62.5±1.6	70.2±2.2	68.6±1.9	**77.6**±0.9
	NMI	15.7±2.6	13.9±3.1	16.5±2.4	16.8±1.9	19.9±2.2	12.6±2.0	15.4±1.7	16.3±2.3	17.1±1.5	25.3±1.8	13.5±2.5	**28.6**±0.8
	ARI	16.1±3.2	14.9±2.7	13.6±2.1	15.0±1.8	16.3±2.5	13.2±1.6	17.6±2.3	14.4±1.9	11.3±2.0	17.9±1.5	17.2±3.6	**22.7**±0.7
	F1	52.3±2.4	51.0±3.5	54.2±2.0	50.8±2.6	60.0±1.9	50.8±2.3	54.6±1.7	58.9±2.1	54.2±1.8	60.5±2.2	59.4±3.8	**66.8**±1.0
SN	ACC	51.7±3.1	55.5±2.4	53.9±2.7	55.8±1.9	58.1±2.3	58.9±2.0	62.8±2.6	60.5±1.8	56.2±2.2	61.5±2.1	66.6±2.3	**75.2**±1.1
	NMI	13.7±3.5	15.5±2.9	15.7±2.6	17.9±2.2	18.6±2.4	18.9±2.1	25.6±2.7	21.6±1.9	16.6±1.8	22.1±2.3	30.4±6.6	**46.9**±0.9
	ARI	12.4±2.2	14.9±3.2	13.4±2.0	16.5±2.4	18.0±1.8	21.1±2.3	28.5±1.6	21.1±2.1	20.8±1.7	24.7±1.9	34.1±5.3	**45.4**±1.0
	F1	50.7±2.5	54.3±1.8	52.6±2.1	46.5±2.3	51.3±2.6	53.7±2.0	58.8±1.7	56.7±1.4	46.3±1.9	52.7±1.6	50.7±2.4	**67.8**±0.8
CV	ACC	12.4±3.1	15.9±2.4	11.2±2.8	13.1±1.9	15.0±2.5	20.6±2.2	16.3±2.6	22.4±2.0	21.9±2.3	24.8±1.8	34.6±3.2	**41.6**±0.9
	NMI	20.4±2.7	18.5±2.3	18.6±2.5	20.4±1.9	20.6±2.4	22.1±1.6	15.6±3.4	20.6±2.1	21.3±2.2	23.5±1.7	34.2±1.5	**40.9**±0.8
	ARI	15.3±2.3	16.7±3.0	10.9±2.1	11.4±1.8	13.1±2.2	12.1±1.5	13.9±2.0	11.6±1.7	13.5±1.9	13.7±1.4	19.3±2.1	**28.9**±0.7
	F1	11.6±1.8	14.7±3.3	10.7±2.4	12.1±1.6	13.9±2.1	15.8±1.9	13.7±1.5	17.9±2.2	16.8±1.7	20.2±1.4	31.6±2.9	**40.6**±0.8

5.2 Experimental Results and Analysis

Performance in Unsupervised Settings To evaluate the advancement of DySFGC, we implement unsupervised clustering experiments. Since most baselines are designed for supervised learning, we extend them to an unsupervised paradigm to keep fairness. The results are shown in Table 1, where the best observations are highlighted in bold, and the runner-up is underlined. From the observations, DySFGC outperforms baselines consistently on five domains. The reason can be attributed to: (1) DySFGC captures the graph signals with different frequencies and introduces spectral contrastive learning and reconstruction loss, which extract more spectral semantic information compared with baselines that focus on the spatial domain; (2) DySFGC implements multi-channel and dynamic parameter interactions between distributed clients and the server, which help local models capture frequency-specific spectral consensus while dynamically integrating parameters based on the contributions to generate comprehensive global consensus.

Table 2: The performance on five datasets under supervised settings with respect to ACC and NMI. The results are in %.

Domain	Metric	DBGCN	S2CAG	FedTAD	FedIIH	FedSPA	S2FGL	GuardFGL	FedGM	FedC4	FedHERO	FedGCN	DySFGC
SM	ACC	61.8±2.1	64.2±3.6	48.9±2.8	60.9±1.9	64.8±2.5	55.1±2.3	71.2±1.7	61.0±2.2	52.4±3.1	63.5±1.8	78.2±2.7	**82.3**±1.0
	NMI	14.8±1.5	15.2±3.2	12.4±2.1	14.9±1.7	15.8±2.3	13.2±1.9	22.3±1.4	15.7±2.6	12.6±2.0	20.2±1.6	27.4±2.2	**33.6**±0.9
SM-BIO	ACC	62.1±2.3	67.8±4.2	54.2±2.1	58.6±1.9	65.1±2.4	59.4±2.7	73.1±1.5	64.0±1.8	56.8±2.5	73.5±2.0	71.4±1.3	**80.4**±1.1
	NMI	6.6±1.9	8.1±2.4	8.1±2.1	8.7±1.8	11.2±1.6	9.8±2.2	16.4±1.4	13.7±1.7	9.5±1.5	13.4±2.0	16.3±2.9	**37.5**±0.7
SM-BIO-SY	ACC	60.8±2.9	57.4±2.2	64.1±2.5	68.8±1.8	71.0±2.1	60.3±2.4	63.0±2.7	69.2±1.6	65.0±2.0	72.4±1.9	70.5±1.5	**79.2**±0.8
	NMI	18.9±2.6	17.1±3.1	19.4±2.3	19.6±1.7	22.0±2.0	15.4±2.5	18.1±2.2	18.9±1.5	19.6±1.9	27.4±1.4	15.7±2.8	**30.5**±0.6
SN	ACC	54.6±3.1	58.2±2.4	56.7±2.8	58.5±1.9	60.6±2.2	61.2±1.7	65.1±2.5	63.0±2.0	58.8±2.3	64.0±1.8	68.7±2.9	**77.1**±0.9
	NMI	16.9±3.4	18.6±2.7	18.3±3.0	20.3±2.2	20.8±2.5	21.1±2.1	28.0±2.4	24.0±1.9	19.3±1.7	24.4±2.0	32.7±6.6	**49.1**±1.1
CV	ACC	15.2±2.9	18.4±2.5	14.0±3.2	15.6±1.8	17.8±2.4	22.9±2.1	19.1±2.7	24.9±2.2	24.2±2.0	27.1±1.9	36.8±3.1	**43.4**±1.0
	NMI	23.0±3.1	21.0±2.8	21.2±2.5	22.6±2.1	22.8±2.6	24.2±1.9	18.1±3.4	22.9±2.0	23.5±2.3	25.6±2.2	36.1±1.6	**42.6**±0.8

Performance in Supervised Settings To better verify the superiority of DySFGC, we implement supervised federated learning for traditional FGL baselines. Specifically, following [16], for the CV domain datasets, one true label is assigned to each class, while for the other domain datasets, five true labels are assigned to each class. As shown in Table 2, DySFGC still achieves superior observations over traditional FGL baselines with supervised information across diverse domains, further demonstrating the advancement of DySFGC. We can also discover that with supervised signals, the models generally achieve better performance compared with unsupervised settings. More results are shown in Section C.

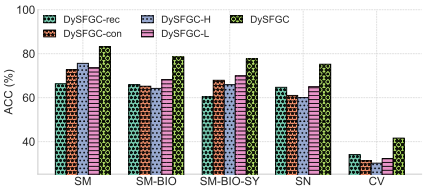


Figure 2: The performance of loss function rationality experiments.

Loss Function Rationality To validate the rationality and effectiveness of different modules in DySFGC, we introduce four variants to implement an ablation study to validate the effectiveness of contrastive learning and reconstruction. From the observations in Fig. 2, all variants show performance

degradation with respect to two metrics, suggesting the effectiveness of different modules in the loss function. DySFGC-rec shows relatively significant performance degradation compared with DySFGC-con, indicating that reconstruction contributes more to obtaining discriminative representations for clustering. The observations of DySFGC-H and DySFGC-L indicate that the spectral contrastive learning of high- and low-pass signals plays important roles in capturing discriminative spectral information to model the semantics of graphs. The details of variants and more experimental results are shown in Section D.

Sensitivity We explore the sensitivity of hyperparameters, including loss function hyperparameters ζ_1, ζ_2. Fig. 3 exhibit the sensitivity experiments of ζ_1 and ζ_2. The observations show slight performance degradation when the values of hyperparameters are relatively small. This is because small values compromise the contribution of corresponding modules. As the values increase, the performance tends to plateau. More experiments on model hyperparameters (hidden size F', and the number of polynomial orders K), interactive sensitivity, and communication epoch T_s are shown in Section E.

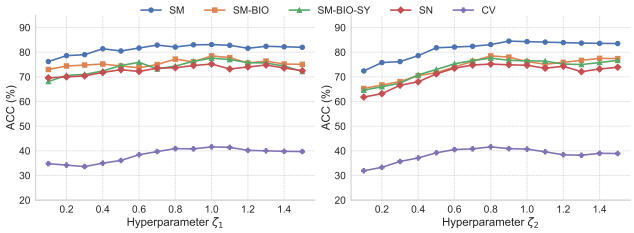


Figure 3: The sensitivity experiments for ζ_1 and ζ_2.

Dynamic Spectral Consensus Rationality We introduce three variants to verify the effectiveness of the dynamic spectral consensus module. From the observations in Fig. 4, all variants exhibit instability and show performance degradation, indicating that the multi-channel updating mechanism and the dynamic server and client updating processes can preserve distinct characteristics for distributed clients to achieve better clustering. Section F contains details of variants, additional results, and ablation studies compared with traditional federated algorithms.

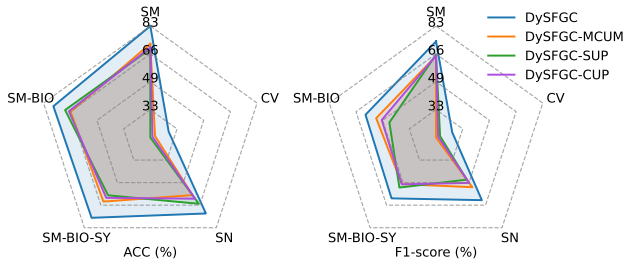


Figure 4: The dynamic spectral consensus rationality experiments.

Other Experiments and Discussion We conduct rich experiments to validate the effectiveness of DySFGC. Section G introduces efficiency experiments to demonstrate the advancement of DySFGC in computational complexity. Section H introduces robustness by adding Gaussian noise to the nodes to demonstrate the stability of DySFGC. Section I conduct graph clustering on large-scale datasets to demonstrate the scalability of DySFGC. Section J explores the number of channels with respect to graph filtering and parameter updating. We discuss the privacy concerns and limitations in Section K.

6 Conclusion

In this paper, we explore spectral federated graph-level clustering, which is the first attempt to exploit the influences of graph signals at different frequencies on the clustering. To address this problem, we develop a novel dynamic spectral federated graph-level clustering (DySFGC). DySFGC extracts discriminative cluster-related semantic information about spectral signals using learnable graph filters. DySFGC also deploys different channels to update spectral consensus with dynamic weights, which helps the private clients enhance the clustering. Extensive experiments on five groups of datasets demonstrate the superiority of DySFGC over eleven baselines. In the future, we plan to extend the generalization capacity of DySFGC to more complex and challenging issues in FGC, such as fairness [55] and heterophily [56].

Acknowledgments

This work was supported by Major Program of National Fund of Philosophy and Social Science of China (Grant No. 25&ZD161), the National Natural Science Foundation of China (Grant No. 72210107001, No. 62576346), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), and the CAS Pilot Project for Basic and Interdisciplinary Scientific Research (Grant No. XDB1380303).

References

[1] Renda Han, Xiaobao Wang, Longbiao Wang, Wenxin Zhang, Ronghao Fu, Kaiming Wang, Zeyu Zhang, and Kuntharrgyal Khysru. A unified graph clustering network. In *Proceedings of the ACM Web Conference 2026*, pages 845–856. ACM, 2026.

[2] Yuanchi Ma, Kaize Shi, Xueping Peng, Hui He, Peng Zhang, Jinyan Liu, Zhongxiang Lei, and Zhendong Niu. Deep graph clustering with triple fusion mechanism for community detection. *IEEE Trans. Comput. Soc. Syst.*, 12(4):1743–1758, 2025.

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

[4] Wenxin Zhang, Xi Xuan, Renda Han, Desheng Wu, Cuicui Luo, and Ljupco Kocarev. Graph clustering with scalable graph filters and view-specific semantic fusion. In *Database Systems for Advanced Applications - 31st International Conference, DASFAA 2026, Jeju, South Korea, April 27-30, 2026, Proceedings, Part II*, volume 16536, pages 83–97, 2026.

[5] Dongyuan Li, Satoshi Kosugi, Ying Zhang, Manabu Okumura, Feng Xia, and Renhe Jiang. Revisiting dynamic graph clustering via matrix factorization. In *Proceedings of the ACM on Web Conference 2025*, pages 1342–1352, 2025.

[6] Renda Han, Guangzhen Yao, Mingyuan Li, Wenxin Zhang, Yu Li, Wen Xin, Huajie Lei, Mengfei Li, Zeyu Zhang, Chengze Du, and Yahe Tian. Multi-relation graph-kernel strengthen network for graph-level clustering. In *International Joint Conference on Neural Networks*, pages 1–8, 2025.

[7] Yazhou Ren, Junlong Ke, Zichen Wen, Tianyi Wu, Yang Yang, Xiaorong Pu, and Lifang He. Multi-view graph clustering via node-guided contrastive encoding. In *Forty-second International Conference on Machine Learning*, 2025.

[8] Renda Han, Wenxuan Tu, Junlong Wu, Wenxin Zhang, Jingxin Liu, Haotian Wang, Jieren Cheng, Huajie Lei, Guangzhen Yao, Lingren Wang, Mengfei Li, and Yu Li. Dual boost-driven graph-level clustering network. In *International Joint Conference on Neural Networks*, pages 1–8, 2025.

[9] Zhihui Xu, Bing Li, and Wenming Cao. Enhancing federated learning-based social recommendations with graph attention networks. *Neurocomputing*, 617:129045, 2025.

[10] Wenxin Zhang, Xi Xuan, Guangzhen Yao, Renda Han, Chuang Zhao, Xiangxiang Lang, Feng Zhou, and Cuicui Luo. Multi-scale asymmetric graph contrastive anomaly detection. *Inf. Process. Manag.*, 64(1):105026, 2027.

[11] Wenxin Zhang and Cuicui Luo. GE-GNN: gated edge-augmented graph neural network for fraud detection. *IEEE Trans. Big Data*, 11(4):1664–1676, 2025.

[12] Qinghua Mao, Xi Lin, Wenchao Xu, Yuxin Qi, Xiu Su, Gaolei Li, and Jianhua Li. Fecograph: Label-aware federated graph contrastive learning for few-shot network intrusion detection. *IEEE Trans. Inf. Forensics Secur.*, 20:2266–2280, 2025.

[13] Junjie Mu, Zonghao Ying, Zhekui Fan, Zonglei Jing, Yaoyuan Zhang, Zhengmin Yu, Wenxin Zhang, Quanchen Zou, and Xiangzheng Zhang. Mask-gcg: Are all tokens in adversarial suffixes necessary for jailbreak attacks? In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13887–13891, 2026.

[14] Tianlong Zhang, Xiaoxi He, Yuxiang Wang, Yi Xu, Rendi Wu, Zhifei Wang, and Yongxin Tong. Fedmetro: Efficient metro passenger flow prediction via federated graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5215–5224, 2025.

[15] 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 *Advanced Intelligent Computing Technology and Applications - 21st International Conference*, volume 15857, pages 27–38, 2025.

[16] Jingxin Liu, Jieren Cheng, Renda Han, Wenxuan Tu, Jiaxin Wang, and Xin Peng. Federated graph-level clustering network. In *the Association for the Advancement of Artificial Intelligence*, pages 18870–18878, 2025.

[17] Wei Feng, Zeyu Bi, Qianqian Wang, and Bo Dong. Federated multi-view graph clustering with incomplete attribute imputation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5118–5126, 2025.

[18] Weihua Ou, Yi Yao, Jiahao Xiong, Yunshun Wu, Xianjun Deng, Jianping Gou, and Jiamin Chen. Spectral adversarial attack on graph via node injection. *Neural Networks*, 184:107046, 2025.

[19] Zekang Li, Ruonan Liu, Dongyue Chen, and Qinghua Hu. Or-gate mixup multiscale spectral graph neural network for node anomaly detection. *IEEE Trans. Neural Networks Learn. Syst.*, 36(9):16692–16705, 2025.

[20] Wenxin Zhang, Jingxing Zhong, Guangzhen Yao, Renda Han, Xiaojian Lin, Lei Jiang, Zeyu Zhang, and Cuicui Luo. Dual-channel heterophilic message passing for graph fraud detection. In *International Joint Conference on Neural Networks*, pages 1–8, 2025.

[21] Liang Yang, Xin Chen, Jiaming Zhuo, Di Jin, Chuan Wang, Xiaochun Cao, Zhen Wang, and Yuanfang Guo. Disentangled graph spectral domain adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

[22] Xuan Zheng, Yihang Lu, Rong Wang, Feiping Nie, and Xuelong Li. Structured graph-based ensemble clustering. *IEEE Trans. Knowl. Data Eng.*, 37(6):3728–3738, 2025.

[23] Jinbin Yang, Jinyu Cai, Yunhe Zhang, Sujia Huang, and Shiping Wang. Towards adaptive masked structural learning for graph-level clustering. *IEEE Trans. Netw. Sci. Eng.*, 12(3):2021–2032, 2025.

[24] Xiaoqiang Yan, Kun Deng, Quan Zou, Zhen Tian, and Hui Yu. Self-cumulative contrastive graph clustering. *IEEE CAA J. Autom. Sinica*, 12(6):1194–1208, 2025.

[25] Yazhou Ren, Junlong Ke, Zichen Wen, Tianyi Wu, Yang Yang, Xiaorong Pu, and Lifang He. Multi-view graph clustering via node-guided contrastive encoding. In *Forty-second International Conference on Machine Learning*, 2025.

[26] Xinyue Deng, Dongliang Wang, Xinfeng Li, Yuying Zhu, Yanan Li, and Zhenwen Ren. Multi-view clustering with tensor-aligned dynamic anchor graphs. *Neurocomputing*, 664:132087, 2026.

[27] Zheming Xu, Congyan Lang, Lili Wei, Liqian Liang, Tao Wang, Yidong Li, and Michael C. Kampffmeyer. UNAGI: unified neighbor-aware graph neural network for multi-view clustering. *Neural Networks*, 185:107193, 2025.

[28] Yaowen Hu, Wenxuan Tu, Yue Liu, Xinhang Wan, Junyi Yan, Taichun Zhou, and Xinwang Liu. Scalable attribute-missing graph clustering via neighborhood differentiation. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Xunkai Li, Yinlin Zhu, Boyang Pang, Guochen Yan, Yeyu Yan, Zening Li, Zhengyu Wu, Wentao Zhang, Ronghua Li, and Guoren Wang. Openfgl: A comprehensive benchmark for federated graph learning. *Proc. VLDB Endow.*, 18(5):1305–1320, 2025.

[30] Xingbo Fu, Zihan Chen, Yinhan He, Song Wang, Binchi Zhang, Chen Chen, and Jundong Li. Virtual nodes can help: Tackling distribution shifts in federated graph learning. In *the Association for the Advancement of Artificial Intelligence*, pages 16657–16665, 2025.

[31] Hao Zhang, Xunkai Li, Yinlin Zhu, and Lianglin Hu. Rethinking federated graph learning: A data condensation perspective. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 6967–6975, 2025.

[32] Linan Yue, Qi Liu, Yawen Li, Fangzhou Yao, Weibo Gao, and Junping Du. Empowering federated graph rationale learning with latent environments. In *Proceedings of the ACM on Web Conference 2025*, pages 3863–3873, 2025.

[33] Wentao Yu, Shuo Chen, Yongxin Tong, Tianlong Gu, and Chen Gong. Modeling inter-intra heterogeneity for graph federated learning. In *the Association for the Advancement of Artificial Intelligence*, pages 22236–22244, 2025.

[34] Zihan Tan, Suyuan Huang, Guancheng Wan, Wenke Huang, He Li, and Mang Ye. S2FGL: spatial spectral federated graph learning. In *Forty-second International Conference on Machine Learning*, 2025.

[35] Guoqing Chao, Zhenghao Zhang, Lei Meng, Jie Wen, and Dianhui Chu. Federated incomplete multi-view clustering with globally fused graph guidance. In *Forty-second International Conference on Machine Learning*, 2025.

[36] Zeli Guan, Yawen Li, Junping Du, Runqing Tang, and Xiaolong Meng. Adpfedgnn: Adaptive decoupling personalized federated graph neural network. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5253–5261, 2025.

[37] Jingyu Chen, Runlin Lei, and Zhewei Wei. Polygcl: GRAPH CONTRASTIVE LEARNING via learnable spectral polynomial filters. In *The Twelfth International Conference on Learning Representations*, 2024.

[38] 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*, 2026.

[39] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1185–1196, 2025.

[40] Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. Dbloss: Decomposition-based loss function for time series forecasting. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025*, 2025.

[41] Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, pages 102339–102355, 2026.

[42] Xiaojian Lin, Wenxin Zhang, Yuchu Jiang, Wangyu Wu, Yiran Guo, Kangxu Wang, Zongzheng Zhang, Guijin Wang, Lei Jin, and Hao Zhao. Butter: Frequency consistency and hierarchical fusion for autonomous driving object detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 111–120, 2025.

[43] Xiaojian Lin, Yang Li, Yaomin Shen, Wenxin Zhang, Yu Liu, Ping Zhang, and Bo Chen. From twelve leads to single: Cross-lead knowledge distillation for domain-generalized atrial fibrillation detection. *Biomedical Signal Processing and Control*, 129:111520, 2027.

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

[45] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *CoRR*, abs/2007.08663, 2020.

[46] Xiaoyang Lin, Renchi Yang, Haoran Zheng, and Xiangyu Ke. Spectral subspace clustering for attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 789–799, 2025.

[47] Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 5716–5724, 2024.

[48] Zihan Tan, Guancheng Wan, Wenke Huang, He Li, Guibin Zhang, Carl Yang, and Mang Ye. Fedspa: Generalizable federated graph learning under homophily heterogeneity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15464–15475, 2025.

[49] Luying Zhong, Xuan Lai, Junjie Zhang, Zhiqin Huang, and Zheyi Chen. Guardfgl: Similarity-driven federated graph learning with adversarial robustness and membership privacy. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4062–4073, 2025.

[50] Zekai Chen, Xunkai Li, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Rethinking client-oriented federated graph learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 393–402, 2025.

[51] Zihan Chen, Xingbo Fu, Yushun Dong, Jundong Li, and Cong Shen. Fedhero: A federated learning approach for node classification task on heterophilic graphs. *Trans. Mach. Learn. Res.*, 2025, 2025.

[52] Wenxin Zhang, Yuhao Zhu, Cuicui Luo, Desheng Wu, Xi Xuan, Ljupco Kocarev, Renda Han, and Xiangxiang Lang. Feature and structural uncertainty modeling for risk identification in supply chain networks. *Chaos, Solitons & Fractals*, 201:117238, 2025.

[53] Wenxin Zhang, Xi Xuan, Renda Han, Zonghao Ying, Cuicui Luo, Desheng Wu, and Ljupco Kocarev. Robust rumor detection against noise. *Neurocomputing*, 671:132741, 2026.

[54] 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. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025*, 2025.

[55] Xiaojian Lin and Michael Losavio. A comprehensive survey on bias and fairness in generative ai: Legal, ethical, and technical responses. In *Advances in Intelligent Data Analysis and Applications*, pages 249–259, 2026.

[56] Wenxin Zhang, Rong Fu, Xi Xuan, Renda Han, Zonghao Ying, Guangzhen Yao, Simon Fong, Lingfei Ren, and Cuicui Luo. A gnn-based fraud detection approach against heterophily inconsistencies. *Appl. Soft Comput.*, 203:116010, 2026.

[57] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the Third Conference on Machine Learning and Systems*, 2020.

[58] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR*, abs/1912.00818, 2019.

[59] Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In Jason Tsong-Li Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 433–444. ACM, 2008.

[60] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate club: An API oriented open-source python framework for unsupervised learning on graphs. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*, pages 3125–3132, 2020.

A Details of Datasets

The details of datasets are shown in Table 3. We split fifteen datasets into five groups, including small molecules (SM), Bioinformatics (BIO), Synthetic (SY), social network (SN), and Computer vision (CV). Wherein, the SM domain has 2 clusters, the BIO domain has 2 clusters, the SY domain has 2 clusters, the SN domain has 3 clusters, and the CV domain has 15 clusters.

Table 3: The statistical description of datasets.

Datasetts	Domain	Classes	#Graphs	A.# Node	A.#Edges
MUTAG	SM	2	188	17.93	19.79
BZR			405	35.75	38.36
COX2			467	41.22	43.45
DHFR			756	42.43	44.54
PTC_MR			344	14.29	14.69
AIDS			2000	15.69	16.20
BZR_MD			306	21.30	225.06
DD	BIO	2	1178	284.32	715.66
PROTEINS			1113	39.06	72.82
SYNTHETIC	SY	2	300	100.00	196.00
COLLAB	SN	3	5000	74.49	2457.78
IMDB-MULTI			1500	13.00	65.94
Letter-high	CV	15	2250	4.67	4.50
Letter-low			2250	4.68	3.13
Letter-med			2250	4.67	3.21

B Evaluation Metrics

We implement the experiments of DySFGC and all baselines on the same device and configurations. Some methods are supervised learning methods, which are not directly compatible with the unsupervised learning framework. Therefore, we leverage the embeddings of the last encoding layer as the outputs and implement reconstruction and contrastive learning to train the model. Besides, we use random shuffling to generate negative samples. All methods are deployed on a Python 3.9.12 environment, utilizing a single NVIDIA A40 GPU, 40GB of RAM, and a 2.60GHz Xeon(R) Gold 6240 CPU.

We introduce the metric details:

- Accuracy (ACC) measures the percentage of correctly clustered samples relative to the ground truth labels after optimal label alignment. Since clustering algorithms assign arbitrary labels that do not necessarily correspond to the true class labels, the Hungarian algorithm is employed to find the optimal one-to-one mapping between predicted clusters and true classes. ACC is computed as:

$$\text{ACC} = \frac{\sum_{i=1}^{n} \delta(y_i, \text{map}(c_i))}{n}, \tag{19}$$

  where n is the total number of samples, y_i is the true label, c_i is the predicted cluster label, $\text{map}(\cdot)$ represents the optimal mapping function, and $\delta(\cdot, \cdot)$ is the Kronecker delta function that equals 1 when arguments are identical and 0 otherwise. ACC ranges from 0 to 1, with higher values indicating better clustering performance.

- Normalized Mutual Information (NMI) quantifies the mutual dependence between the predicted clustering assignments and the ground truth labels, normalized by their respective entropies. This symmetric measure evaluates how much information about the true labels is captured by the clustering results. NMI is defined as:

$$\text{NMI}(U, V) = \frac{2 \cdot I(U, V)}{H(U) + H(V)}, \tag{20}$$

  where U represents the true class labels, V represents the predicted cluster assignments, $I(U, V)$ denotes the mutual information between U and V, and $H(U)$ and $H(V)$ are the entropies of U and V respectively. NMI values range from 0 (no mutual information) to 1 (perfect correlation), with higher values indicating better clustering quality that preserves the information content of the true class structure.

- Adjusted Rand Index (ARI) evaluates the similarity between two data partitions while correcting for chance agreement. Unlike the standard Rand Index, ARI accounts for the expected similarity of random cluster assignments, making it more interpretable and robust. It compares pairwise relationships between samples:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}, \tag{21}$$

  where RI is the Rand Index calculated as $\frac{a+b}{C_2^n}$, with a being the number of sample pairs in the same cluster in both partitions, b the number of pairs in different clusters in both partitions, and C_2^n the total number of sample pairs. ARI ranges from -1 to 1, where 1 indicates perfect agreement, 0 represents random labeling, and negative values suggest worse-than-random performance.
- In clustering evaluation, the F1 score computes the harmonic mean of precision and recall for each cluster when matched to its corresponding true class. The macro-averaged F1 score is commonly used for overall assessment:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{22}$$

  where Precision is the proportion of correctly assigned samples within a cluster, and Recall is the proportion of samples from a true class correctly captured by the cluster. After establishing optimal cluster-class correspondence, the F1 scores for all classes are averaged to obtain the final metric. F1 ranges from 0 to 1, with higher values indicating better balance between cluster purity and completeness.

These metrics comprehensively evaluate the performance of the clustering algorithm from multiple perspectives.

For baseline fairness, we illustrate the imperative modification to baselines.

- FedTAD: In the client side, we imeplement reconstruction on $\mathbf{h}_i^{HYB}$ with an MLP. On the server side, we keep unsupervised knowledge distillation loss $\mathcal{L}_{diverg}$, $\mathcal{L}_{div}$, and replace supervised loss $\mathcal{L}_s$ with the node-subgraph contrastive learning for server-side knowledge generation.
- FedSPA: It uses labels to split the message passing process. We leverage node-subgraph contrastive learning and reconstruction to calculate the scores and then generate pseudo-labels according to a threshold to guide the process.
- S2FGL: It is the only federated graph spectral baseline; thus, we don't add contrastive loss. We keep the distillation loss $\mathcal{L}_{FKD}$ and reconstruction $\mathcal{L}_{FGMA}$, and remove cross-entropy $\mathcal{L}_{CE}$.
- GuardFGL: It employs variational inference with supervised cross-entropy. We employ standard variational loss with reconstruction on $\hat{\mathbf{Y}}$.
- FedGM: The main idea is to measure the distance between two cross-entropy losses. We replace the supervised loss with the node-subgraph contrastive learning and reconstruction losses.
- FedHERO: We keep the regularization term $\mathcal{L}_{smooth}$ and use $\mathbf{Z}$ to implement node-subgraph contrastive learning and reconstruction.

C Supervised Learning

Table 4 shows the ARI and F1 metrics on supervised learning experiments. From the observations, we can see that DySFGC achieves the best performance compared with baselines, which demonstrates that the proposed method remains highly functional under supervised settings.

D Loss Function Rationality

First, we introduce the details of variants. DySFGC-rec denotes DySFGC training without reconstruction, DySFGC-con denotes DySFGC training without contrastive learning, DySFGC-H denotes DySFGC training without $\mathcal{L}_H$, and DySFGC-L denotes DySFGC training without $\mathcal{L}_L$.

Table 4: The performance on five datasets under supervised settings with respect to ARI and F1. The results are in %.

Domain	Metric	DBGCN	S2CAG	FedTAD	FedIIH	FedSPA	S2FGL	GuardFGL	FedGM	FedC4	FedHERO	FedGCN	DySFGC
SM	ARI	0.8±2.2	5.6±3.1	4.4±1.7	6.8±2.5	15.9±2.8	10.2±1.9	32.5±2.3	18.2±2.1	3.3±2.6	28.5±1.8	33.2±3.4	**47.6**±0.9
	F1	53.9±2.6	58.1±1.9	42.6±3.2	52.1±2.4	59.6±2.1	47.3±2.5	64.0±1.6	50.6±3.3	43.2±2.7	60.4±2.0	69.0±2.8	**78.1**±1.1
SM-BIO	ARI	3.7±2.8	6.1±2.3	5.4±1.9	6.3±2.6	11.8±1.7	6.1±2.1	16.4±2.4	13.5±1.8	8.4±2.5	14.2±2.2	19.6±3.1	**40.5**±0.7
	F1	55.4±1.8	58.1±2.5	50.2±2.7	55.0±1.6	52.1±2.2	60.0±2.9	62.8±2.0	53.1±2.3	43.6±1.9	55.0±2.4	61.0±1.7	**68.0**±0.8
SM-BIO-SY	ARI	18.8±3.2	17.6±2.7	16.4±2.4	17.6±2.1	18.7±1.9	15.6±2.5	20.1±2.8	16.8±1.6	14.1±2.3	20.2±2.0	19.3±3.6	**25.1**±1.0
	F1	55.0±2.4	53.8±3.1	57.0±2.6	53.6±1.8	62.1±2.3	53.5±2.0	57.1±1.5	61.0±2.5	56.8±2.2	62.7±1.9	61.2±3.8	**68.6**±1.1
SN	ARI	15.1±2.2	17.6±3.1	15.8±1.8	18.6±2.5	20.1±1.9	23.0±2.1	30.6±1.6	23.4±2.4	23.1±1.7	26.9±1.4	36.3±5.3	**47.2**±0.8
	F1	53.4±2.6	56.9±1.9	55.0±2.2	49.3±2.8	53.9±1.5	56.0±2.0	61.2±1.7	59.1±2.3	49.1±2.1	55.1±1.4	52.6±2.4	**69.5**±0.9
CV	ARI	18.1±2.4	19.2±3.2	13.4±2.1	13.9±1.8	15.6±2.5	14.5±1.6	16.2±2.3	14.2±1.9	15.9±2.2	16.1±1.5	21.0±2.8	**30.6**±1.1
	F1	14.3±1.7	17.2±3.3	13.6±2.4	14.9±1.9	16.5±2.8	18.3±2.1	16.4±1.5	20.1±2.6	19.3±2.0	22.4±1.4	33.5±3.1	**42.1**±1.2

Second, Fig. 5 shows the ARI and F1 metrics results. From the observations, we can see that DySFGC remains the optimal performance. This phenomenon indicates that contrastive learning and reconstruction on spectral signals are helpful to capture high-quality representations for local models.

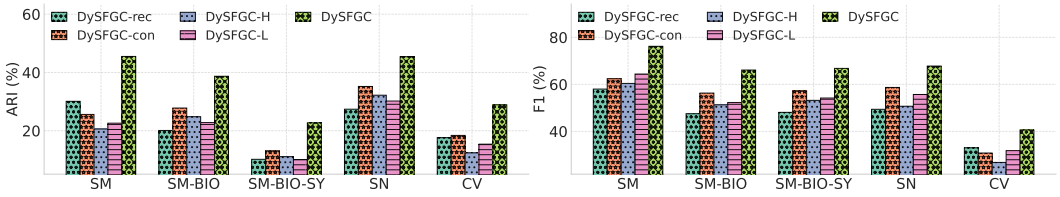


Figure 5: The additional performance of loss function rationality experiments.

E Sensitivity

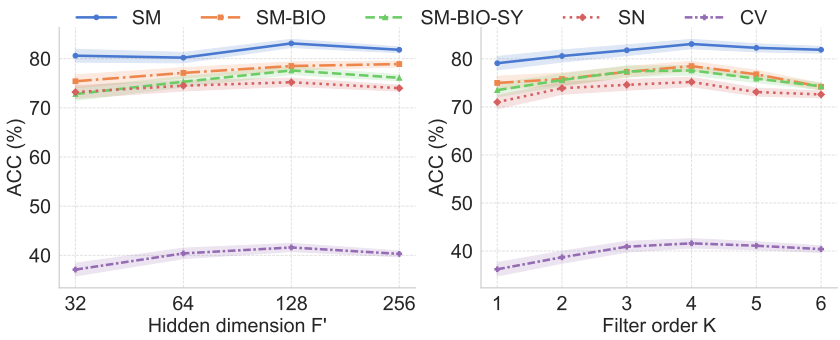


Figure 6: The sensitivity experiments with respect to F' and K.

Model Hyperparameters Fig. 6 shows the sensitivity of F' and K, and the performance also shows a slight fluctuation because they influence the representation ability, while maintaining stable in total.

Interaction Sensitivity We implement the interactive sensitivity experiments of ζ_1 and ζ_2 on five domains, as shown in Fig. 8. From the observations, we can see that the ACC has a little fluctuation with the hyperparameters varying, while remaining robust and stable on the whole. Specifically, when both hyperparameter values are small, the performance of the model is relatively low. This is because the small values prevent the loss function of the model from achieving the expected learning effect.

Communication Epoch We explore the influence of different communication epochs on the performance with respect to ACC and F1 metrics. As shown in Fig. 7, it can be observed that increasing the number of communication epochs helps the model capture more robust and discriminative global

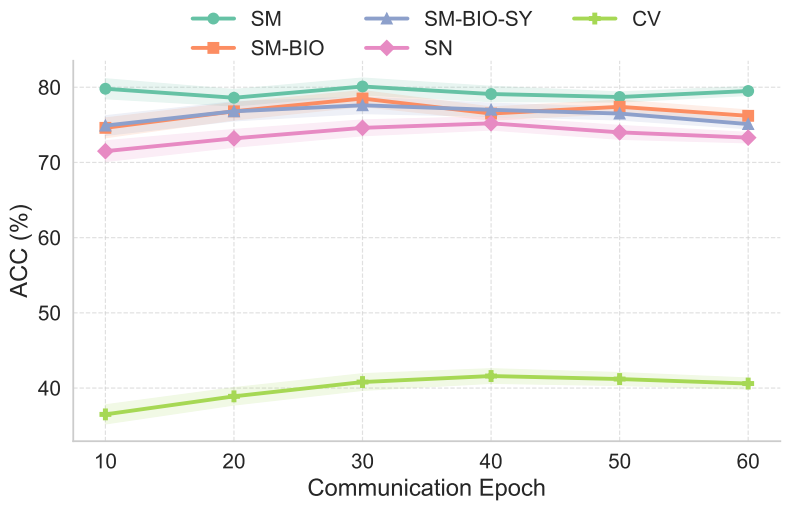


Figure 7: The sensitivity experiments with respect to the communication epoch.

consensus, thereby enhancing the clustering capability of local models. Specifically, most domains achieve optimal performance when the number of communication epochs is set to 30 or 40. This also validates that overly frequent communication may cause local models to lose their specificity.

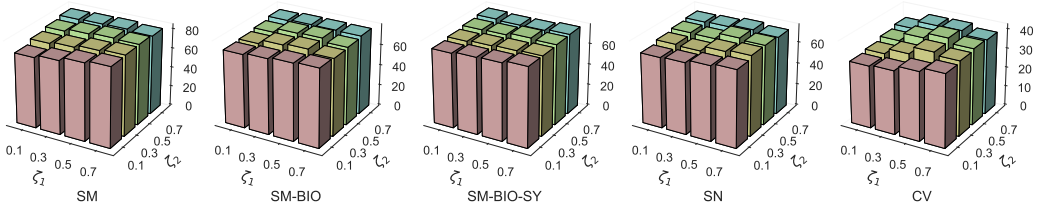


Figure 8: The interactive sensitivity experimental results with respect to ζ_1 and ζ_2.

F The Utility of Parameter Updating Mechanism

F.1 Details of variants

DySFGC-MCUM denotes DySFGC without a multi-channel updating mechanism. DySFGC-SUP denotes DySFGC without a server updating mechanism while utilizing the proposed client updating mechanism. DySFGC-CUP denotes DySFGC without a client updating mechanism while using the proposed server updating mechanism.

F.2 MNI performance in ablation experiments

Fig. 9 shows the NMI metric of DySFGC with traditional parameter updating mechanisms. DySFGC demonstrates a distinct advantage in this NMI metric. The observations further validate the effectiveness of using frequency-specific channels and the dynamic updating mechanisms.

F.3 Comparison with baselines

Furthermore, we then introduce some traditional federated updating mechanisms for comparison to further validate the advancement of the proposed dynamic updating mechanism, including Local, FedAvg [44], FedProx [57], and FedPer [58], where Local means DySFGC without parameter updating. From the observations in Fig. 10, we can conclude that the proposed dynamic updating

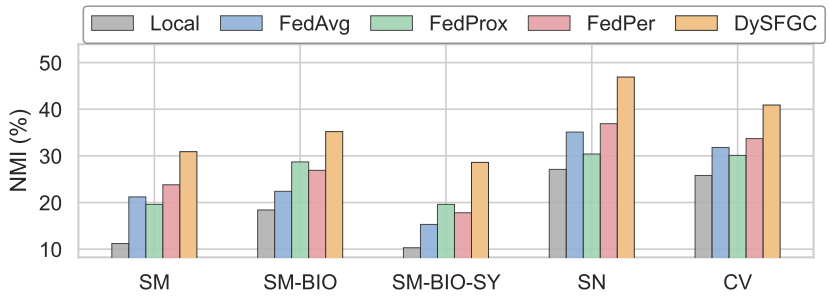


Figure 9: The utility of the parameter updating mechanism with respect to NMI.

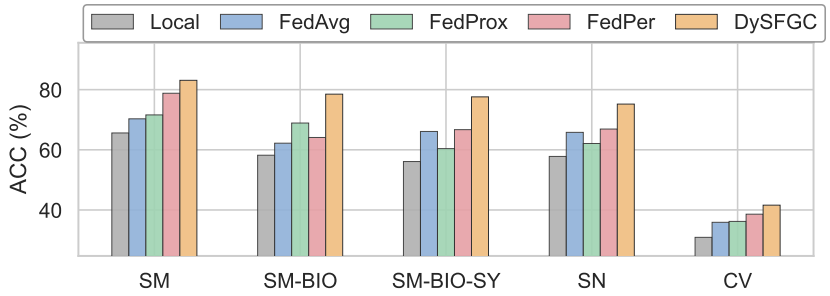


Figure 10: The comparison experiment of DySFGC with three traditional federated updating mechanisms and Local.

mechanism achieves the best performance with respect to AUROC. This phenomenon indicates that the multi-channel updating mechanism and the dynamic updating mechanism enhance the ability of models to capture discriminative features.

G Efficiency

We validate the efficiency of DySFGC compared with all baselines on SM and SM-BIO domains, as shown in Fig. 11. We can see that DySFGC achieves superior efficiency over baselines. This is because DySFGC introduces polynomial filters, which reduce the computational costs, and also employs a lightweight contrastive strategy, avoiding a complicated negative sampling process. Some baselines show competitiveness, for example, FedGCN employs structure-oriented features to obtain the latent representations, which encode critical topology characteristics without complex information propagation. FedIIH utilizes variational inference that requires multiple sampling to model the distributions, leading to extra computational cost.

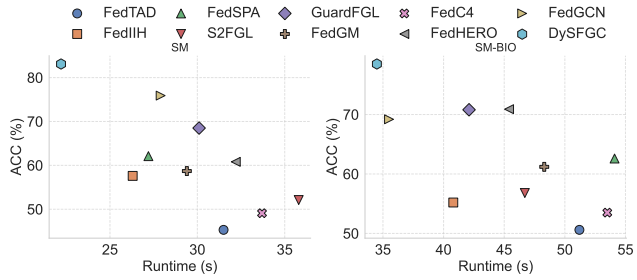


Figure 11: The efficiency experiments of DySFGC compared with nine federated baselines on SM and SM-BIO domains.

The efficiency experiments on the rest domains are shown in Fig. 12. From the results on SM-BIO-SY, SN, and CV domains, we can see that DySFGC still achieves the best efficiency, which is consistent with the observations on SM and SM-BIO domains. In conclusion, the proposed method has advanced efficiency against state-of-the-art baselines.

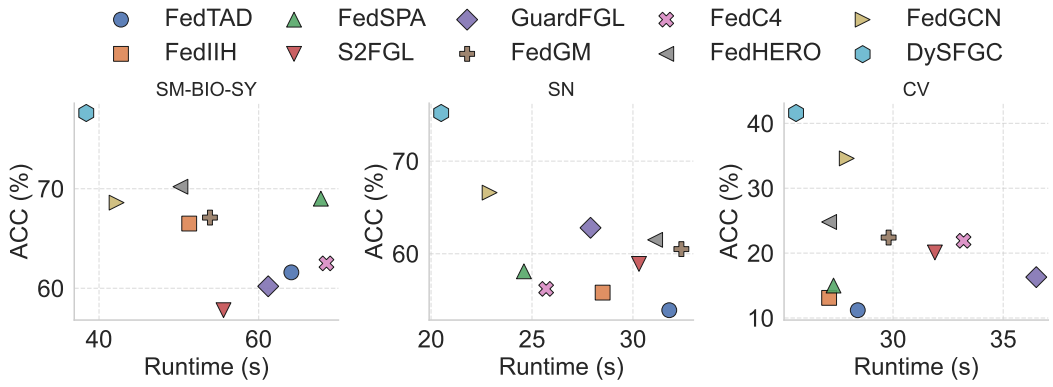


Figure 12: The efficiency experiments on the other three domains.

H Robustness

We validate the robustness of DySFGC by adding standard Gaussian noise to nodes in each client at different rates. From Table 5, the average degradations of the ACC metric with increasing noise level for the five domains are, respectively, 0.95, 0.45, 0.70, 0.55, and 0.80. This phenomenon sufficiently illustrates that the proposed DySFGC is robust against noise. Specifically, the degradation on SM and CV domains is relatively significant compared to other domains because the noise may destroy the semantic information in the graph.

Table 5: ACC(%) performance under different noise rates.

Noise Rate	SM	SM-BIO	SM-BIO-SY	SN	CV
5%	79.5±1.8	78.0±2.4	77.1±1.5	74.8±2.7	41.2±2.1
10%	79.0±1.6	77.6±2.0	76.9±1.5	74.1±2.3	40.5±1.8
15%	77.6±1.9	77.1±2.3	76.2±1.3	73.2±2.8	39.7±2.0
Avg.	0.95	0.45	0.70	0.55	0.80

We conduct extensive robustness experiments compared with baselines, as shown in Table 6. From the results, we can see that DySFGC maintains the superiority over baselines in the noise environment, while some baselines exhibit significant performance degradation.

Table 6: The robustness experiments of DySFGC compared with baselines at 10% noise rate with respect to ACC. The results are in %.

Domain	DBGCN	S2CAG	FedTAD	FedIIH	FedSPA	S2FGL	GuardFGL	FedGM	FedC4	FedHERO	FedGCN	DySFGC
SM	57.2±1.8	60.1±2.9	44.2±2.1	56.4±1.7	61.3±2.3	50.8±2.6	67.9±1.9	57.5±2.5	47.9±2.2	59.7±2.0	73.8±2.8	**79.0**±1.6
SM-BIO	58.6±2.2	63.4±3.1	50.5±1.4	55.1±2.5	61.2±1.8	54.7±2.3	68.6±1.3	59.2±2.7	52.4±1.9	68.8±2.1	68.1±1.2	**77.6**±2.0
SM-BIO-SY	56.5±1.9	53.8±2.4	59.9±1.6	64.3±2.8	66.2±1.3	56.1±2.2	59.5±2.5	65.4±1.7	60.3±2.1	68.2±1.9	66.4±2.6	**76.9**±1.5
SN	50.1±2.3	53.9±1.8	52.4±2.7	54.1±1.5	56.8±2.4	57.4±1.9	61.3±2.1	58.7±1.3	54.5±2.6	59.2±2.2	64.9±1.7	**74.1**±2.3
CV	11.4±1.5	14.2±2.8	9.8±1.9	12.1±2.3	13.9±1.4	18.2±2.5	15.6±1.7	20.4±2.1	20.6±2.6	22.8±1.3	32.5±2.2	**40.5**±1.8

I Scalability

In this section, we validate the scalability of DySFGC on large-scale datasets. We introduce a new group, SML-SOCL, based on the TUDataset. This cross-domain group consists of five small-molecule (SM) datasets: NCI-H23 [59], OVCAR-8 [59], P388 [59], MOLT-4 [59], and MCF-7 [59], as well

as two social datasets (SCO): reddi_threads [60] and twitch_egos [60]. The details of datasets are shown in Table 7.

Table 7: The statistical description of datasets.

Datasetts	**Domain**	**Classes**	**#Graphs**	**A.# Node**	**A.#Edges**
NCI-H23	SM	2	40353	26.07	28.10
OVCAR-8			40516	26.08	28.11
P388			41472	22.11	23.56
MOLT-4			39765	26.10	28.14
MCF-7			27770	26.40	28.53
reddit_threads	SCO	2	203088	23.93	24.99
twitch_egos			127094	29.67	86.59

Following default settings, we set 7 clients, and each client holds a private graph dataset. We select FedGCN [16], the first federated graph clustering method, as the baseline for comparison. From the observations in Table 8, DySFGC outperforms FedGCN on all four metrics, which indicates that the performance advantage is maintained on large-scale datasets. We will include additional baselines in the next version.

Table 8: Scalability performance of the proposed DySFGC model compared with FedGCN on the large-scale domain (SM-SOC). All results are in %.

Methods	ACC	NMI	ARI	F1
FedGCN	$69.43_{\pm 1.2}$	$28.06_{\pm 1.1}$	$33.62_{\pm 0.8}$	$61.73_{\pm 1.3}$
DySFGC	$\mathbf{75.26}_{\pm 0.9}$	$\mathbf{33.64}_{\pm 0.5}$	$\mathbf{35.06}_{\pm 0.5}$	$\mathbf{68.40}_{\pm 0.8}$

J Number of channels

There is a fact that spectral graph learning studies (for example, graph contrastive learning, graph clustering, and so on) often focus on extracting graph signals with different frequencies based on different graph filters to capture discriminative representations. Specifically, high-pass filters can capture high-frequency signals, low-pass filters capture low-frequency signals, and band-pass filters capture band-pass signals. This classification criterion is based on the principle of the filters because it is clearly noticed that the signals in these three frequency domains contain distinct semantic information about graphs. Based on this, we realize some components of DySFGC are exclusively for a specific frequency domain, for example, high-pass signals are only captured by high-pass filters, while unrelated to low-pass filters. So, the parameters should be discriminated based on the frequency domain. Therefore, we adopt multi-scale filters to increase the channel count, then fuse the multi-scale features within each frequency domain, ultimately forming three frequency-domain signals. This approach is used to validate the effect of multi-scale filtering.

To validate the rationality of using three channels, we conduct experiments with respect to the number of channels. Specifically, we change the number of channels in each frequency domain. By default, the channel is divided into three channels: high-pass, low-pass, and band-pass. For variants, we will increase the number of channels in specific frequency domains to 2 and 3, respectively. The corresponding total number of channels is 6 and 9. For these variants, we first leverage intra-channel signals concatenation to obtain comprehensive frequency-specific signals, and then employ a loss function to train the parameters.

The ACC and F1 results are shown in Table 9. From the observations, we can see that although the F1 performance of using two channels surpasses DySFGC on the SM-BIO domain, increasing the number of channels can lead to a comprehensive performance improvement on four groups of datasets with respect to ACC and F1. Therefore, using three channels is a relatively suitable strategy for DySFGC.

Table 9: The experiments on the number of channels with respect to ACC and F1. All results are in %.

#. Channels	SM		SM-BIO		SN		CV	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
9	$78.1_{\pm 1.0}$	$74.6_{\pm 1.5}$	$76.7_{\pm 1.2}$	$63.2_{\pm 1.1}$	$74.3_{\pm 1.1}$	$66.2_{\pm 1.6}$	$40.4_{\pm 1.5}$	$35.2_{\pm 1.3}$
6	$75.4_{\pm 1.2}$	$72.8_{\pm 1.8}$	$77.0_{\pm 1.1}$	$\mathbf{67.3}_{\pm 0.9}$	$72.6_{\pm 0.8}$	$64.5_{\pm 1.1}$	$40.3_{\pm 1.3}$	$38.3_{\pm 1.5}$
3(DySFGC)	$\mathbf{80.1}_{\pm 0.9}$	$\mathbf{76.2}_{\pm 1.1}$	$\mathbf{78.5}_{\pm 0.8}$	$66.1_{\pm 0.8}$	$\mathbf{77.6}_{\pm 0.9}$	$\mathbf{66.8}_{\pm 1.0}$	$\mathbf{41.6}_{\pm 0.9}$	$\mathbf{40.6}_{\pm 0.8}$

K Discussion

We first discuss the privacy concerns and then illustrate the limitations.

Privacy DySFGC has the basic ability to protect privacy. In DySFGC, each client performs local encoding and clustering on its private graph dataset while keeping the raw graph elements, including nodes, edges, and features, unshared. The communication information between clients and the server consists solely of model parameters. Therefore, it will not cause significant privacy leakage and bad implications for individual privacy. We only upload the global feature obtained by average pooling of each graph sample. This global feature is a comprehensive representation of all node features in the graph, which helps protect node-level privacy within each graph sample.

In addition, the core objective of this paper is to explore spectral solutions for FGC, extending traditional spatial-domain federated graph clustering methods. Although parameter or gradient sharing can still leak sensitive information, the privacy risks from parameter sharing are a known issue across FL methods in general, which is not the core issue discussed in this paper.

Limitations This paper represents a preliminary exploration of FGC in the spectral domain. However, it does not address more complex scenarios, such as cross-dataset clustering. Cross-dataset clustering aims to perform clustering on unseen datasets after training on a source dataset, which is highly challenging as it requires strong model generalization capabilities. One feasible approach is to enhance generalization through prompt learning. However, prompt learning relies on certain label guidance, making it highly challenging to be in a fully unsupervised setting.

L Complexity Analysis and Algorithm Details

In this section, we first analyze the computational complexity from the client side. For a client C_m, we employ Chebyshev polynomials to approximate the spectral filter, and the complexity of the filtering operation is constrained to $\mathcal{O}(K \cdot |E| \cdot |F|)$, where E is the number of edges. The band-pass signal is derived via an MLP through a learnable dynamic combination of the low- and high-pass representations. The complexity is $\mathcal{O}(N_m \cdot F'^2)$. This ensures the complexity scales linearly with the graph size.

Then, we provide the pseudocode of the algorithm, as is shown in Algorithm 1.

Algorithm 1: The pseudocode of DySFGC.

Input: Client set $\mathcal{C} = \{C_1, \cdots, C_M\}$; Graphs datasets $\mathcal{G} = \{\mathcal{G}^1, \cdots, \mathcal{G}^M\}$, Client number M; Client epochs T_c; Communication epochs T_s
Output: Cluster results;
Initialize parameters $\mathbf{W}_s$ of the server.
for $t_s = 1$ **to** T_s **do**
  for C_m **in** $\mathcal{C}$ **do**
    Obtain the client parameters $\mathbf{W}_m$ from the server using the multi-channel updating mechanism by Eq. (15)-Eq. (17).
    for $t_c = 1$ **to** T_c **do**
      Obtain local and corresponding global representations of graphs with different frequencies by Eq. (1)- Eq. (5).
      Calculate spectral contrastive loss $\mathcal{L}_{\text{con}}$ by Eq. (7).
      Calculate reconstruction loss $\mathcal{L}_{\text{rec}}$ by Eq. (8).
      Calculate KL divergence $\mathcal{L}_{\text{kl}}$ by Eq. (11).
      Update the local parameters $\mathbf{W}_s$.
    end for
    Update the local parameters using the multi-channel updating mechanism by Eq. (12)-Eq. (14)
  end for
end for

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the contributions and scope of this paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not claim theoretical contributions. However, we provide theoretical analyses for several key results in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper provides open access to the data, with sufficient instructions to faithfully reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see the Experiment Setup section and the Appendix for more details.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the average performance over multiple runs with different random seeds, together with ±1 standard deviation as error bars to quantify the stochastic variability of the training process.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide detailed descriptions of the experimental environment, including hardware and software configurations. Specifically, experiments are conducted on GPU-based platforms, and all implementation details are reported to facilitate reproducibility. Although we do not explicitly report memory consumption or exact wall-clock training time, we provide a thorough computational complexity analysis of the proposed method to characterize its efficiency.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work complies with the NeurIPS Code of Ethics. The research does not involve human subjects, personal data, or sensitive information, and no ethical concerns are raised by the datasets or experimental procedures used in this paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide the discussion of broader impacts in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of the assets used in this paper are properly credited, and the corresponding licenses and terms of use are explicitly stated and fully respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.