FedSSP: Federated Graph Learning with Spectral Knowledge and Personalized Preference

Zihan Tan[1*] **Guancheng Wan**[1*] **Wenke Huang**[1*] **Mang Ye**[1,2†]
[1] National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Wuhan University, Wuhan, China.
[2] Taikang Center for Life and Medical Sciences, Wuhan University, Wuhan, China
{zihantan,guanchengwan,wenkehuang,yemang}@whu.edu.cn

Abstract

Personalized Federated Graph Learning (pFGL) facilitates the decentralized training of Graph Neural Networks (GNNs) without compromising privacy while accommodating personalized requirements for non-IID participants. In cross-domain scenarios, the presence of structural heterogeneity poses significant challenges for pFGL. Nevertheless, previous pFGL methods incorrectly share non-generic knowledge globally and fail to tailor personalized solutions locally under domain structural shift. We innovatively reveal that inherent domain structural shift can be well reflected by the spectral nature of graphs. Correspondingly, our method overcomes it by sharing generic spectral knowledge. Moreover, we indicate the biased message-passing schemes for graph structures and propose the personalized preference module. Combining both strategies for effective global collaboration and personalized local application, we propose our pFGL framework **FedSSP** which **S**hares generic **S**pectral knowledge while satisfying graph **P**references and further performs extensive experiments on cross-dataset and cross-domain settings to demonstrate its superiority.

1 Introduction

Graph Neural Networks (GNNs) [49, 67] have demonstrated their superiority in modeling graph data which frequently emerges in a variety of scenarios, as exemplified by graph clustering [40, 41, 39], graph contrastive learning [57, 65], anatomy detection [38, 59, 69], knowledge graph [64, 63, 32] and so on. However, large amounts of graph data are generated by edge devices in reality, which brings privacy concerns and the challenges of data silos [70, 22]. To address these difficulties, federated learning (FL) has recently been applied to graph learning [15, 36, 17]. It allows models on various clients to collaborate without sharing local data [24, 12, 21] and makes federated graph learning (FGL) a promising direction. Nonetheless, the non-IID problem remains a major challenge in FGL, as graph data from different clients usually vary significantly. In such scenarios, a single global model struggles to adapt well to the local data of each client with inconsistent data distributions [53, 51, 10]. To tackle these challenges, personalized federated graph learning (pFGL) has emerged, offering customized GNNs for each client to achieve satisfying local performance [1, 54].

However, pFGL still encounters substantial challenges from structural heterogeneity [25], especially in domain shift tasks, for instance, between social networks [75, 76] and molecular structures [52, 48]. There are two significant drawbacks to previous algorithms as Fig. 1 demonstrates. For global collaboration, the considerable domain structural shifts inevitably lead to non-generic knowledge,

[*] Equal contribution. † Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

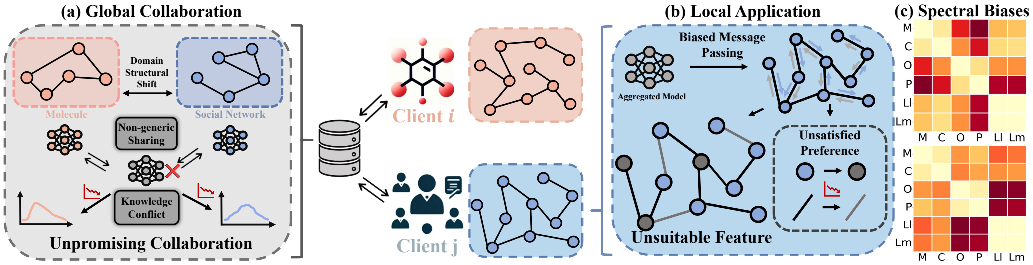


Figure 1: **Problem illustration.** We illustrate the challenges of the cross-domain scenario. (a) Considering the domain structural shifts, clients struggle with **knowledge conflict** caused by non-generic sharing which arises from the shifts, thus leading to unpromising global collaboration. (b) The aggregated message-passing scheme suffers from **unsatisfied preferences** of various unique datasets in this scenario, thus leading to unsuitable features of graphs in local applications. (c) The heat map of Jensen-Shannon divergence of algebraic connectivity [14] and eigenvalues distributions among six datasets from three different domains. Spectral characteristics exhibit significant **biases across** domains but are more **similar** within a **same** domain.

thus resulting in knowledge conflict. Both current methods suffer from conflict and are trapped in unpromising collaboration. Specifically, [54] share non-generic structural encoding and struggle with structural knowledge conflict, while strategy against the conflicts by [61] limits the potential for collaboration. The key to addressing knowledge conflict is pursuing a way to share generic knowledge that benefits all clients. Based on this observation, we raise the question: 1) *how to design a method to share generic knowledge to address* ***knowledge conflict*** *under domain structural shift?*

For local applications, each client owns its unique dataset with distinct structural characteristics in this cross-dataset scenario. Due to the GNN message-passing nature, distinct graph structures stored in different clients prefer different message-passing schemes. Consequently, the scheme provided by aggregated GNN exhibits biases from the optimum when applied locally, thus leading to unsuitable features. Both current methods neglect the unsatisfied preferences of unique graph structures from various clients. This deficiency leads us to consider: *2) how to design personalized plans to deal with* ***unsatisfied preferences*** *of unique graph datasets from various clients?*

To address problem 1), given that structural shifts make it hard to directly achieve generic sharing at the structural level, we propose to explore the structure shifts from another spectral perspective since previous works have demonstrated the strong correlation between graph structure and spectra [2, 66, 37, 28]. The major advantage of spectra is the detailed propagation and processing of graph signals on the graph structure, thus facilitating the discovery of generic knowledge in several certain processes unaffected by structural shifts. To validate our assumption, we first conducted experiments and explicitly revealed the domain spectral biases that directly reflect domain structural shifts on spectra as Fig. 1 demonstrates. To tackle these spectral biases directly to overcome structural shifts, we design **Generic Spectral Knowledge Sharing** (GSKS) to share generic spectral knowledge in spectral encoders, enabling clients to benefit from others through collaboration without knowledge conflict. Conversely, other components containing non-generic knowledge are designed to be retained locally. Correspondingly, clients can customize powerful graph convolutions for their local graph characteristics while benefiting from generic knowledge without knowledge conflict. Through this strategy, we promote the sharing of generic spectral knowledge and the personalizing of non-generic knowledge, thus achieving effective collaboration against knowledge conflict.

Moreover, we attempt to achieve target 2) and design suitable personalized plans for each client graph structure locally. Specifically, we explore the message-passing nature of GNN [5, 13, 55]. From the spectral perspective, spectral encoders strongly affect message transmission. Therefore, when aggregated spectral encoders are applied to distinct graph structures locally, they can deviate from the optimal message-passing scheme for the client [35]. Consequently, GNNs extract inappropriate frequency messages which lead to unsuitable features. To meet the unique graph preferences, we innovatively configure a learnable preference for each client and propose **Personalized Graph Preference Adjustment** (PGPA). These personalized preference modules apply adjustments to the feature extracted with the participation of global spectral encoders, allowing the feature to suit the unique graph structures of each client. Moreover, we introduce a regularization term to constrain the preference module to focus on local preferences to address the over-reliance. By combining these two strategies, we propose FedSSP, a framework that promises the effectiveness of collaboration by sharing generic spectral knowledge while satisfying the personalized graph preferences of clients.

In conclusion, our key contributions are:

- We are the first to reveal domain structural shifts through spectral biases, as well as consider the various unsatisfied preferences of distinct datasets from various clients.
- We proposed FedSSP, which innovatively overcomes knowledge conflicts from a spectral perspective and implements personalized graph preference adjustments for each client.
- We conduct experiments in various cross-dataset and cross-domain settings, proving that our approach outperforms several current state-of-the-art methods and achieves optimal results.

2 Related Work

2.1 Spectral GNNs

Spectral GNNs [4, 8] are based on spectral graph filters set in the spectral domain, providing powerful models for graph neural networks [60]. Spectral GNNs can generally be categorized into two types: those with fixed filters and those with learnable filters. Fixed filter spectral GNNs, such as APPNP [16], utilize personalized PageRank (PPR) [46] to construct their filtering functions. GNN-LF/HF [77] designs filter weights from the perspective of graph optimization functions. Learnable filter spectral GNNs include subclass that approximates arbitrary filters using various types of orthogonal polynomials, including Bernstein [20], Chebyshev [9, 19], and Jacobi [58]. Another subclass parameterizes the filters by neural networks, including LanczosNet [34] and Specformer[3]. The robust modeling capability of spectral graph neural networks on data inspires us to leverage this foundation to tackle the issue of structural heterogeneity across domains.

2.2 Personalized Federated Learning

Federated learning facilitates privacy-preserving collaborative learning on local data, introducing methods like FedAvg [44] to address this. Yet, it struggles with non-IID data across clients. Several techniques aim to address the challenge [29, 30, 26], but achieving a global model that suits all local data remains difficult [25]. Personalized Federated Learning (pFL) has attracted increasing attention for its ability to address the non-IID problem [11, 33, 53]. Research has approached improvements from various aspects. In personalized-aggregation-based methods, FedPHP aggregates the global model and old personalized models locally using moving averages to preserve historical information [31], FedALA achieves personalized aggregation through personalized masks [72], and APPLE improves by uploading core models and employing directed relationship vectors for downloading [42]. In model-splitting-based approaches, FedRoD [6] learns with a global feature extractor and two heads for both global and personalized tasks. FedCP decouples features suitable for global and local heads through a conditional policy scheme [73]. Moreover, methods based on regularization and knowledge distillation have also been utilized to enhance pFL. However, pFL methods lack targeted strategy designs for graph, making them not particularly suited for pFGL scenarios.

2.3 Federated Graph Learning

Recent studies have utilized the FL framework for distributed training of GNNs without sharing graph data [18, 36]. Current Federated Graph Learning (FGL) research can be categorized into two types: intra-graph and inter-graph FGL. In inter-graph FGL, each client has distinct graphs, and they jointly participate in federated learning to either improve GNN modeling of local data or achieve a model that can generalize across different datasets [62, 54]. Intra-graph FGL, on the other hand, aims to deal with challenges such as missing link prediction [7], subgraph community detection [74, 1], and node classification [23, 56]. However, most FGL methods lack specific design considerations for our scenario. More precisely, there is a general absence of consideration for the heterogeneity of graph-level structures and the personalized needs of different clients brought about by structural characteristics. In this paper, we focus on inter-graph FGL, taking into account spectral domain biases and the uniqueness of graph structures that result in client-specific preferences, to customize a personalized optimal model for each client specifically for graph classification tasks.

3 Preliminary

3.1 Graph Signal Filter

Assume that we have a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the node set with $|\mathcal{V}| = n$ and $\mathcal{E}$ is the edge set. The corresponding adjacency matrix is defined as $A \in \{0,1\}^{n \times n}$, where $A_{ab} = 1$ if there is an edge between nodes a and b, and $A_{ab} = 0$ otherwise. The normalized graph Laplacian matrix is defined as $\tilde{L} = I_n - D^{-1/2}AD^{-1/2}$, where I_n denotes the $n \times n$ identity matrix and D is the diagonal degree matrix. We assume $\mathcal{G}$ is undirected. $\tilde{L}$ is a real symmetric matrix, whose spectral decomposition can be written as $\tilde{L} = U\Lambda U^T$, where the columns of U are the eigenvectors and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ are the corresponding eigenvalues ranged in $[0, 2]$. The Graph Fourier transform of a signal $x \in \mathbb{R}^{n \times d}$ is defined as $\tilde{x} = U^T x \in \mathbb{R}^{n \times d}$. The inverse transform is defined as $x = U\tilde{x}$ [50]. The i-th column of U denotes a frequency component corresponding to the eigenvalue λ_i. Let $\tilde{x}_\lambda = U_\lambda^T x$, where U_λ represents the eigenvector corresponding to λ, be the frequency component of x at λ frequency. We can use a function $g : [0, 2] \to \mathbb{R}$ to filter each frequency component by multiplying $g(\lambda)$. By defining $\Lambda = diag(\lambda)$, g implements filtering on Λ , thus ultimately implementing filtering on the graph signal x. The whole process is defined as follows:

$$Ug(\Lambda)U^T x. \tag{1}$$

By defining $g(\tilde{L}) = \sum_{k=0}^{K} \alpha_k \tilde{L}^k$, where g is often set to be a polynomial of degree K for parameterizing the filter, the filtering process can be rewritten as follows:

$$Ug(\Lambda)U^T x = g(\tilde{L})x. \tag{2}$$

3.2 Federated Learning and Personalized Federated Learning

Traditional FL leverages isolated data of distributed clients and collaboratively learns models $\mathcal{M}$ for a generalizable global model without leaking privacy. Specifically, the goal is to minimize:

$$\min_\theta f_g(\theta) = \min_\theta \sum_{i=1}^{N} w_i \mathcal{M}_i(\theta), \tag{3}$$

where $f_g(\cdot)$ denotes the global objective. It is computed as the weighted sum of the N local objectives, with N being the number of clients and $w_i \geq 0$ being the weights. The local objective $\mathcal{M}_i(\cdot)$ is often defined as the expected error over all data under local data $\mathcal{D}_i$.

In the context of personalized federated learning, the global objective takes a more flexible form:

$$\min_{\mathbf{\Theta}} f_p(\mathbf{\Theta}) = \min_{\theta_i, i \in [N]} \sum_{i=1}^{N} w_i \mathcal{M}_i(\theta_i), \tag{4}$$

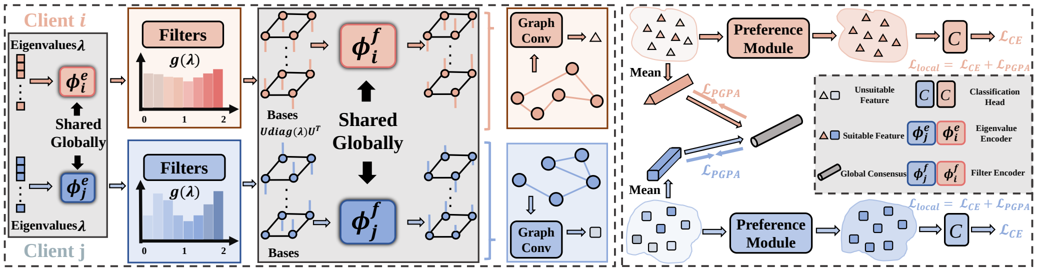
where $f_p(\cdot)$ is the global objective for the personalized algorithms, and $\mathbf{\Theta} = [\theta_1, \theta_2, \ldots, \theta_N]$ is the matrix with all personalized models. In this work, we aim to obtain the optimal $\mathbf{\Theta}^* = \arg\min_{\mathbf{\Theta}} f_p(\mathbf{\Theta})$, which equivalently represents the set of optimal personalized models $\theta_i, i \in [N]$.

4 Methodology

4.1 Generic Spectral Knowledge Sharing (GSKS)

Motivation. Current methods suffer from knowledge conflict arising from non-generic sharing under domain structural shifts. Since structural shifts impede the direct generic sharing at the structural level, we are the first to reveal and further innovatively resolve knowledge conflicts from the spectral perspective. To explicitly address the spectral biases that reflect structural shifts in Fig. 1, we base our pFGL strategy on spectral GNNs and further propose Generic Spectral Knowledge Sharing (GSKS) for effective collaboration that overcomes spectral bias and structural shift, thereby further overcoming knowledge conflict through sharing generic spectral knowledge.

Eigenvalue filtering. Aiming at more expressive representations of frequency information, the eigenvalues are firstly mapped from scalars to meaningful vectors for subsequent calculations of



(a) Generic Spectral Knowledge Sharing **(b) Personalized Graph Preference Adjustment**

Figure 2: **Architecture illustration** of FedSSP. The left box (a) refers to Generic Spectral Knowledge Sharing (GSKS), where we address knowledge conflict and promote effective global collaboration by **sharing generic** spectral knowledge in spectral encoders ϕ^e and ϕ^f while **retaining non-generic** in other components. The right box (b) represents Personalized Graph Preference Adjustment (PGPA), in which we leverage **preference module** guided by $\mathcal{L}_{PGPA}$ for satisfying various preferences of unique datasets and more suitable graph features in local applications. These two boxes correspondingly refer to the two core strategies of our framework FedSSP.

meaningful frequency extracting in the multi-head attention module as follows:

$$\phi^e(\theta^e;\lambda) = \begin{cases} \sin(\beta\lambda/c^{q/d}), & \text{if } q \text{ is even,} \\ \cos(\beta\lambda/c^{(q-1)/d}), & \text{if } q \text{ is odd,} \end{cases} \tag{5}$$

where c keeps eigenvalues within a suitable numerical range to distinguish different eigenvalues for trigonometric functions. $q \in \mathbb{Z} \cap [0, d-1]$ is the index for dimension d while β controlling the importance of λ with defaulted value 10000. Moreover, θ^e here denotes parameters of the eigenvalue encoder ϕ^e, by which eigenvalues are mapped from scalars to meaningful vectors for richer frequency information that is more expressive for further filtering by attention module and decoder in the procedure of $\mathbb{R}^1 \to \mathbb{R}^d$. The initial representations are the concatenation of eigenvalues and their encodings: $\lambda' = [concat[\lambda_1, \phi^e(\theta^e;\lambda_1)], \ldots, concat[\lambda_n, \phi^e(\theta^e;\lambda_n)]]^T \in \mathbb{R}^{n\times(d+1)}$. Then the multi-head attention module is leveraged. After stacking multiple transformer blocks, spectral decoder ψ^d for eigenvalue decoding can learn new eigenvalues from the expressive representations of the spectra:

$$\lambda_m = \psi^d(\text{Attention}(Q\theta_m^Q, K\theta_m^K, V\theta_m^V)), \tag{6}$$

Where the representations learned by the m-th head are fed into ψ^d, while ψ^d denotes a combination of liner and optional activation. $\lambda_m \in \mathbb{R}^{n\times 1}$ is the filtered eigenvalues by the m-th head. The whole process in Eq. (6) acts as a spectral filter g for the origin eigenvalues in Eq. (1).

To address the challenge of knowledge conflict, we attempt to explore the functionality of each module. The eigenvalue encoder ϕ^e captures multi-scale representations of eigenvalues and provides meaningful vectors of distinct frequencies. Since the mapping from eigenvalues to vectors by ϕ^e is independent of the domain characteristics, θ^e of ϕ^e is considered to contain generic knowledge. In contrast, as the spectral biases we revealed in Figure 1 demonstrate, there exist biases in eigenvalues distribution across domains while spectral characteristics within the same domain are more similar. Therefore, the attention module learns the non-generic magnitudes and relative dependencies specific to the spectral characteristics of each client. Correspondingly, the eigenvalue decoder focuses on decoding the most suitable message-passing scheme and client-specific frequency components from the representation processed by the attention module. Attention module and decoder together formed g in Eq. (1), aiming at designing personalized filtered eigenvalue that guides message-passing at a personalized suitable frequency. Therefore, θ^e is shared in our strategy to achieve generic spectral knowledge sharing and effective collaboration unaffected by knowledge conflict.

Specifically, client i uploads its update of θ_i^e. At the t-th iteration ($t \geq 0$), the central server distributes global spectral encoder weights θ_g^t. Accordingly, client i updates local GNN weights including θ_i^e with their dataset $\mathcal{D}_i$ and send these updates as $\Delta\theta_i^t = \theta_i^t - \theta_g^t$ to the central server. Then the server aggregates the received local updates and modifies the global weight θ_g^{t+1} as follows:

$$\theta_g^{t+1} = \theta_g^t + \frac{\sum_{i=1}^{N} \Delta\theta_i^t}{N} (i \in [1..N]), \tag{7}$$

notably, aggregation based on sample size is unsuitable in this scenario for effective collaboration across various domains and datasets. Since clients here possess unique datasets with significant quantitative variance, those with larger datasets tend to dominate the collaboration. Thereby preventing them from benefiting from the spectral and frequency knowledge of clients with fewer samples. Correspondingly, clients with fewer graphs are almost entirely dominated by knowledge that does not originate from their local data. To address the problem, we leverage a direct average of spectral encoder weights from all clients to achieve fair collaboration and cross-dataset knowledge sharing.

Personalized graph convolution constructing. After getting M filtered eigenvalues, filter encoder: $\phi^f(\theta^f; B)\ \mathbb{R}^{M+1} \to \mathbb{R}^d$ is leveraged to construct the bases for personalized graph convolution. To avoid confusion and distinguish from the mentioned filter g on eigenvalues in Eq. (1), filter here in filter encoder refers to the filtering on feature message-passing through bases in graph convolution. New bases are first reconstructed and concatenated along the channel dimension. Specifically, by defining $\Lambda_m = diag(\lambda_m)$, they are fed into filter encoder ϕ^f as follows:

$$B_m = \mathbf{U}\Lambda_m\mathbf{U}^T, \quad \forall m \in \{1, \dots, M\}, \tag{8}$$

$$\hat{B} = \phi^f(\theta^f; B), \tag{9}$$

where $B = [B_1, B_2, \dots, B_M] \in \mathbb{R}^{n\times n\times M}$ while $B_m \in \mathbb{R}^{n\times n}$ is the m-th new basis and $\hat{B} \in \mathbb{R}^{n\times d}$ is for feature filtering in graph convolution ultimately. The original bases B_m are initially constructed from the eigenvectors U and the filtered eigenvalues λ_m, with $\phi^f(\theta^f; B)$ responsible for the learnable transformation of bases from original to new. This transformation facilitates learning more suitable schemes for graph message-passing and processing at various frequencies. Filter encoders $\phi^f(\theta^f; B)$ in clients encapsulate knowledge of various frequency components which affects how much graph signal varies from nodes to their neighbors for better graph convolution construction. Since distinct clients own different datasets here, they fail to adequately learn various message processing schemes during local training. Therefore, this knowledge is worth sharing globally, as collaboration on filter encoder can aid $\phi^f(\theta^f; B)$ of each client learn how to construct suitable graph convolution from various message-passing schemes. Therefore, we design client i to upload the weights θ_i^f of its ϕ_i^f the same way as ϕ_i^e Eq. (7), thereby achieving a comprehensive understanding of different frequency messages in graphs. Subsequently, it enables the construction of powerful personalized graph convolutions as follows:

$$x_v' = \sigma\left((\hat{B} \cdot x_v)\theta^{Conv} + x_v\right), \tag{10}$$

where x_v is the node v representation from the previous layer, while x_v' represents the output of the current layer. θ^{Conv} denotes weights of graph convolution, and σ refers to the optional activation. Ultimately, the representations of all nodes within a graph are aggregated by an average pooling layer to form the overall feature representation of graph $\mathcal{G}_l$ in dataset $\mathcal{D}_i$ of client i as follows:

$$h_l = \frac{1}{|\mathcal{V}|}\sum_{v=1}^{|\mathcal{V}|} x_v, \quad \forall l \in \{1, \dots, |\mathcal{D}_i|\}, \tag{11}$$

where h_l is defined as the average of all node features in graph $\mathcal{G}_l$, namely the graph feature, while $\mathcal{V}$ refers to the node quantities in graph l here. By sharing generic spectral knowledge and retaining client-specific knowledge we successfully achieve effective collaboration that overcomes spectral bias, thereby domain structural shift from the spectral perspective.

4.2 Personalized Graph Preference Adjustment(PGPA)

Motivation. Due to the GNN message-passing nature, distinct graph structures prefer different message-passing schemes, especially when meeting the uniqueness of datasets in cross-dataset and cross-domain scenarios. Consequently, The spectral encoders under collaboration fail to satisfy the preferences of graphs with distinctive structural characteristics. Correspondingly, graph convolutions tend to learn biased message-passing schemes, thereby extracting unsuitable graph features. Our approach provides personalized adjustments for client preference to address this challenge. Moreover, we solve the over-reliance issue that arises from the process of satisfying various preferences.

To satisfy the various preferences and make the graph features more suitable for graphs, we propose a learnable preference module that adjusts to features extracted by client i to satisfy its special requirements and unique graph structure preference explicitly. The module includes learnable parameters matched in size with the feature space, thus acting as a refined tool to flexibly satisfy the preferences of each client when updating during local training. Considering local model $\mathcal{M}$ including feature extractor $\mathcal{F}(\theta^F;\mathcal{G})$, classification head $\mathcal{C}(\theta^C, h)$, and preference module $\mathcal{P}(\delta)$, where $\mathcal{G}$ represents graphs contained in dataset $\mathcal{D}$ of a client. The whole feature-extracting process can be defined as follows in our strategy:

$$h = \mathcal{F}(\theta^F;\mathcal{G}), \quad h' := h + \delta, \tag{12}$$

by integrating the original feature h with preference adjustments δ, h' becomes the ultimately suitable feature that satisfies client preference. Now we leverage adjusted feature h' for $z' = \mathcal{C}(\theta^C, h')$ instead of the original unsuitable representation h . Specifically, the local loss for client i is:

$$\mathcal{L}_i = \mathbb{E}_{(z'_i, y_i)\sim\mathcal{D}_i}[\mathcal{L}(z'_i, y_i)], \tag{13}$$

however, the preference module inevitably learns not only the client-specific preference but also aspects that should be handled by the feature extractor $\mathcal{F}$ without a guide for preference. As a result, the local feature extractor $\mathcal{F}$ might overly rely on adjustments provided by the preference module during training, thereby hindering its capability. Correspondingly, this over-reliance can negatively impact federated collaboration. When the capability of $\mathcal{F}$ is degraded, the shared spectral encoders fail to convey beneficial knowledge to others, leading to unpromising collaboration.

Therefore, it is essential to guide the preference module to focus solely on the aspects of client preferences, rather than interfering with the feature extraction guided by collaboration. We achieve this by guiding the output of the feature extractor to align more closely with global graph features. Consequently, the PGPA module is directed to concentrate on client preferences. To implement this, we first calculate the mean of local graph features in each iteration:

$$\bar{h}_i = (1-\mu)\cdot \bar{h}_i^{\text{pre}} + \mu \cdot \bar{h}_i^{\text{cur}}, \tag{14}$$

where μ denotes the momentum we introduce for bringing graph modeling patterns from previous batches to the current batch in the same local epoch. h_i^{pre} and h_i^{cur} represent the local mean graph feature of the previous batches and the current batch. It is necessary to distinguish between mean and prototype. In this scenario, clients own various datasets, thus the class information is client-specific. Correspondingly, the mean h_i which represents the average modeling for graphs in client i is unrelated to class information. After local training, client i uploads its mean to the server for global consensus aggregation. Based on our exploration of Eq. (7), a direct average is leveraged here:

$$\bar{h}_g = \frac{\sum_{i=1}^{N} \bar{h}_i}{N}, \tag{15}$$

where $\bar{h}_g$ refers to the global graph modeling consensus calculated from all samples across all clients. Accordingly, we employ the Mean Squared Error (MSE) to measure the distance between the local graph feature mean and the global graph mean obtained from the previous round. This measurement serves as a regularization term to encourage the local graph feature to align closer to the global modeling consensus, thus guiding the preference module to focus on preference and correspondingly address the over-reliance issue. Specifically, local feature extractors are encouraged to extract certain frequency messages in graphs that reflect the global modeling consensus, making the PGPA only responsible for client-specific preference. The local loss of client i is now defined as:

$$\mathcal{L}_i = \mathbb{E}_{(z'_i, y_i)\sim\mathcal{D}_i}[\mathcal{L}(z'_i, y_i) + \tau \cdot \text{MSE}(\bar{h}_i, \bar{h}_g)]. \tag{16}$$

By implementing $\text{MSE}(\bar{h}_i, \bar{h}_g)$, we explicitly align local graph modelings with the global consensus, thus guiding the preference module $\mathcal{P}$ to focus on preference and addressing the considered issue of over-reliance by forcing the preference module to focus on client-specific graph preference.

5 Experiments

5.1 Experimental Setup

We perform experiments on graph classification tasks in various cross-dataset and cross-domain scenarios to validate the superiority of our framework FedSSP.

Datasets. Follow the settings in [54], we use 15 public graph classification datasets from four different domains, including Small Molecules (MUTAG, BZR, COX2, DHFR, PTC_MR, AIDS, NCI1), Bioinformatics (PROTEIN, OHSU, Peking_1), Social Networks(IMDB-BINARY, IMDB-MULTI), and Computer Vision (Letter-low, Letter-high, Letter-med) [45]. Node features are available in some datasets, and graph labels are either binary or multi-class. We create six different settings: (1) cross-dataset setting based on seven small molecules datasets (SM); (2)-(6) both cross-dataset and cross-domain settings based on datasets from two different domains(BIO-SM, SM-CV) and three different domains(BIO-SM-SN, BIO-SN-CV, CHEM-SN-CV)

Baselines. We compare ours with several SOTA federated approaches. (1)**Local** as the first baseline; (2)**FedAvg** [44]; (3)**FedProx** [30] which address heterogeneity issues in FL; (4)**APPLE** [43] and (5)**FedCP** [73], two state-of-the-art pFL method;(6)**FedSage** [74], (7)**GCFL** [61] ,(8)**FGSSL** [23], and (9)**FedStar** [54], four state-of-the-art FGL methods.

Implementation Details. The experiments are conducted using NVIDIA GeForce RTX 3090 GPUs as the hardware platform, coupled with Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz. For each setting, every client holds its unique graph dataset, among which 10% are held out for testing, 10% for validation, and 80% for training. We leverage the AdamW optimizer [27] for local GNNs with learning rate 0.001, the default parameter of $\epsilon = 1e - 8$, and $(\beta_1, \beta_2) = (0.99, 0.999)$, as suggested by [47, 68]. The number of communication rounds is 200 for all FL methods. We report the results with an average of over 5 runs of different random seeds.

5.2 Experimental Results

Performance Comparison We show the federated graph classification results of all methods under six different non-IID settings, including one cross-dataset setting (SM), two cross-double-domain settings (BIO-SM, SM-CV) cross-multi-domain settings (BIO-SM-SN, BIO-SN-CV, SM-SN-CV). We summarize the final average test accuracy in Tab. 1. These results indicate that FedSSP outperforms all other baselines in five out of the six settings. Notably, traditional FL algorithms such as FedAvg and FedProx failed to outperform self-training due to the strong cross-datasets and cross-domain non-IID challenge of this scenario. Correspondingly, algorithms such as FedStar and FedCP which are designed specifically for pFGL or pFL scenarios perform better here.

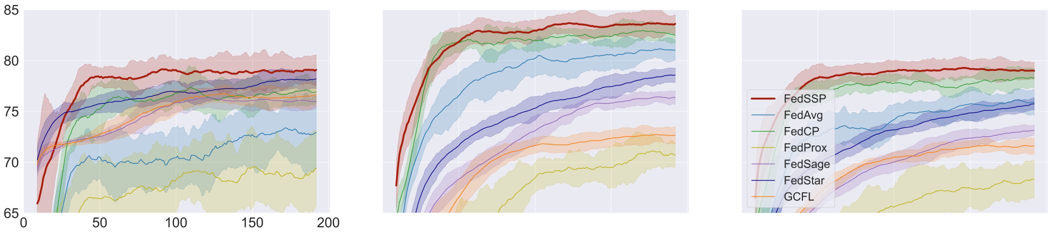
Table 1: Comparison with the state-of-the-art methods on one cross-dataset and five cross-domain settings. Best in bold and second with underline. In each setting, each client owns a unique dataset.

Methods	single-domain	double-domain		Multi-Domain		
	SM	SM-BIO	SM-CV	SM-BIO-SN	BIO-SN-CV	SM-SN-CV
Local	77.33 ± 1.15	72.52 ± 1.86	82.24 ± 1.73	71.13 ± 1.32	72.59 ± 2.70	77.83 ± 0.54
FedAvg [ASTAT17]	74.12 ± 2.10	67.82 ± 1.63	81.21 ± 1.00	67.31 ± 2.56	70.93 ± 2.91	75.33 ± 1.06
FedProx [arXiv18]	69.35 ± 3.36	67.27 ± 4.17	70.02 ± 2.27	63.89 ± 4.33	69.32 ± 1.75	67.15 ± 2.25
FedSage [NeurIPS21]	75.61 ± 1.16	72.60 ± 3.18	76.23 ± 0.49	70.84 ± 0.88	69.69 ± 1.11	73.36 ± 0.86
GCFL [NeurIPS21]	77.71 ± 1.53	72.05 ± 2.20	72.64 ± 0.71	70.43 ± 1.39	67.91 ± 2.15	71.79 ± 0.21
APPLE [IJCAI22]	74.29 ± 1.89	70.40 ± 2.13	76.07 ± 2.55	71.07 ± 1.64	72.52 ± 1.03	72.33 ± 0.42
FedCP [KDD23]	77.58 ± 2.00	71.15 ± 1.77	81.59 ± 0.40	71.32 ± 1.23	$\underline{73.74 \pm 2.53}$	$\underline{78.17 \pm 1.78}$
FGSSL [IJCAI23]	77.90 ± 0.85	72.47 ± 2.15	$\underline{82.60 \pm 0.48}$	68.13 ± 1.71	73.44 ± 1.33	77.90 ± 0.62
FedStar [AAAI23]	$\underline{78.63 \pm 2.11}$	$\underline{72.71 \pm 1.22}$	78.84 ± 1.07	$\mathbf{72.60 \pm 2.45}$	69.51 ± 3.24	75.94 ± 0.40
FedSSP (ours)	$\mathbf{79.62 \pm 2.23}$	$\mathbf{73.66 \pm 2.34}$	$\mathbf{84.29 \pm 0.68}$	$\underline{72.37 \pm 2.18}$	$\mathbf{75.07 \pm 2.70}$	$\mathbf{79.12 \pm 1.23}$

Convergence Analysis Fig. 3 shows the curves of the average test accuracy with standard deviation during the training process across five random runs of three settings (SM, SM-CV, SM-SN-CV) representing single-domain, double-domain, and multi-domain scenarios, including the results of various baselines. As is shown, traditional FL methods such as FedAvg or FedProx own higher standard deviations and are more unstable while methods designed specifically for pFGL scenarios such as GCFL and FedStar are more stable with lower standard deviations.

5.3 Ablation Study

Effects of Key Components Mechanism of FedSSP To better understand the impact of specific design components on the overall performance of FedSSP, we conducted an ablation study in which



(a) single-domain (SM) (b) double-domain(SM-CV) (c) Multi-domain (SM-SN-CV)

Figure 3: Test accuracy curves of FedSSP and six other methods along the communication rounds on our three different settings(SM, SM-CV, SM-SN-CV). The y-axis range is from 65 to 85 for all settings.

we varied these components of single, double, and multi-domain settings(SM, SM-CV, SM-SN-CV). As shown in Tab. 2, compared to FedAvg, GSKS significantly enhances accuracy when applied independently. Correspondingly, as a further exploration of the nature of GNN message passing, PGPA achieves noticeable success in adjusting client preferences.

Effects of Key Component Mechanism of GSKS Tab. 3 discuss the key component of GSKS. We demonstrated the impact of different sharing strategies. Specifically, sharing only non-generic spectral GNN components or all components often fails to outperform self-training, while GSKS successfully dominates all the strategies. Accordingly, the results fully validate the effectiveness of GSKS in sharing universal knowledge and promoting effective collaboration in this scenario.

Table 2: **Ablation study** of key components of FedSSP on single-domain, double-domain, and multi-domain settings (SM, SM-CV, SM-SN-CV).

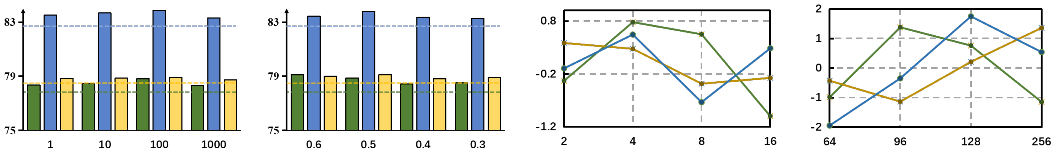
GSKS	PGPA	Setting		
		SM	SM-CV	SM-SN-CV
✗	✗	74.12	81.21	75.33
✓	✗	77.83	82.78	78.54
✗	✓	74.59	81.33	76.12
✓	✓	**79.62**	**84.29**	**79.12**

Table 3: **Ablation study** of key components of GSKS on a single-domain, double-domain, and multi-domain settings (SM, SM-CV, SM-SN-CV).

Ours	Other	Setting		
		SM	SM-CV	SM-SN-CV
✗	✗	77.33	82.24	77.83
✓	✓	74.12	81.21	75.33
✗	✓	77.21	81.64	78.17
✓	✗	**77.83**	**82.78**	**78.54**

5.4 Hyper-parameter Study

We compare the graph classification performance under different values of PGPA parameter τ, momentum μ, number of attention heads, and hidden dimension. Where Fig. 4 shows the results when these hyper-parameters are fixed at different scales and values. It indicates that the choosing of τ can affect the strength of PGPA while performance is not influenced much unless they are set to extreme values. All studies of τ and μ here outperform the baseline. We also test the performances under different values of attention heads and hidden dimensions. For results in Tab. 1, we set up 4 heads for the multi-head attention mechanism while 128 for the hidden dimension.



(a) PGPA Parameter τ (b) Momentum μ (c) Attention Heads (d) Hidden Dimension

Figure 4: **Analysis on hyper-parameter in FedSSP**. Graph classification results under different τ, μ, attention heads, and hidden dimensions. Colors green, blue, and yellow refer to performance on single, double, and multi-domain settings (SM, SM-CV, SM-SN-CV). The dashed lines of corresponding colors represent the baseline test accuracy for each setting, which includes only the GSKS strategy.

6 Discussion

Even though FedSSP has achieved significant success in cross-domain federated graph learning collaborations, it still faces certain limitations as a spectral GNN-based approach. Compared to spatial GNNs, while spectral GNNs have the advantage of overcoming structural heterogeneity from the spectral domain, many spectral GNNs require eigenvectors and eigenvalues, which adds to the computational overhead of data preprocessing and subsequent storage burden.

Furthermore, we notice a similar approach in DBE [71] which employs static global consensus in FL to separate personalized and global information. Nevertheless, it inevitably struggles to handle scenarios where the message-passing of multiple GNNs is continuously updated. It merely provides a static anchor point, making it difficult to establish a global graph modeling consensus that could guide the local GNNs in capturing graph signals. Instead, we align the GNN backbone with dynamic global graph modeling to avoid the preference module from overly extracting features that should be captured by the GNN itself, which could lead to decreased GNN performance and hinder global collaboration. This approach allows for real-time adjustment of message-passing across different client GNNs, focusing the preference module solely on personalization. Additionally, to address issues such as sample size disparity between domains and dominance of large domains in model parameter aggregation, we adopt a direct averaging strategy in dynamic global aggregation instead of conventional weighted averaging to mitigate these concerns.

7 Conclusion

In this paper, we pioneer an innovative exploration of cross-domain personalized Federated Graph Learning. To achieve this goal, we made improvements in two aspects: seeking effective global collaboration and suitable local application, thus proposing a novel framework FedSSP. For global collaboration, we employ GSKS to facilitate the sharing of generic spectral knowledge, overcoming knowledge conflict by domain structural shift from a spectral perspective. For local applications, we design PGPA to satisfy various preferences of unique datasets contained in clients. By integrating these two strategies, FedSSP outperforms various state-of-the-art methods on various cross-dataset and cross-domain pFGL scenarios in graph classification.

8 Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant (62361166629, 62176188, 623B2080). The numerical calculations in this paper had been supported by the supercomputing system in the Supercomputing Center of Wuhan University.

References

[1] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *ICML*.

[2] Deyu Bo, Yuan Fang, Yang Liu, and Chuan Shi. Graph contrastive learning with stable and scalable spectral encoding. In *NeurIPS*, 2023.

[3] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. *arXiv preprint arXiv:2303.01028*, 2023.

[4] Deyu Bo, Xiao Wang, Yang Liu, Yuan Fang, Yawen Li, and Chuan Shi. A survey on spectral graph neural networks. *arXiv preprint arXiv:2302.05631*, 2023.

[5] Shaofei Cai, Liang Li, Jincan Deng, Beichen Zhang, Zheng-Jun Zha, Li Su, and Qingming Huang. Rethinking graph neural architecture search from message-passing. In *CVPR*, 2021.

[6] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*, 2022.

[7] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia, and Huajun Chen. Fede: Embedding knowledge graphs in federated setting. In *IJCKG*, 2021.

[8] Zhiqian Chen, Fanglan Chen, Lei Zhang, Taoran Ji, Kaiqun Fu, Liang Zhao, Feng Chen, Lingfei Wu, Charu Aggarwal, and Chang-Tien Lu. Bridging the gap between spatial and spectral domains: A unified framework for graph neural networks. *ACM Computing Surveys*, 2023.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016.

[10] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. 2020.

[11] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *NeurIPS*, 2020.

[12] Xiuwen Fang, Mang Ye, and Xiyuan Yang. Robust heterogeneous federated learning under data corruption. In *ICCV*, 2023.

[13] Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. How powerful are k-hop message passing graph neural networks. 2022.

[14] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 1973.

[15] Xingbo Fu, Binchi Zhang, Yushun Dong, Chen Chen, and Jundong Li. Federated graph machine learning: A survey of concepts, techniques, and applications. *arXiv preprint arXiv:2207.11812*, 2022.

[16] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

[17] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, 2021.

[18] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. In *ICLR*, 2021.

[19] Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. In *NeurIPS*, 2022.

[20] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*, 2021.

[21] Ming Hu, Yue Cao, Anran Li, Zhiming Li, Chengwei Liu, Tianlin Li, Mingsong Chen, and Yang Liu. Fedmut: Generalized federated learning via stochastic mutation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12528–12537, 2024.

[22] Ming Hu, Peiheng Zhou, Zhihao Yue, Zhiwei Ling, Yihao Huang, Anran Li, Yang Liu, Xiang Lian, and Mingsong Chen. Fedcross: Towards accurate federated learning via multi-model cross-aggregation. In *IEEE International Conference on Data Engineering (ICDE)*, pages 2137–2150. IEEE, 2024.

[23] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. Federated graph semantic and structural learning. In *IJCAI*, 2023.

[24] Wenke Huang, Mang Ye, Bo Du, and Xiang Gao. Few-shot model agnostic federated learning. In *ACM MM*, pages 7309–7316, 2022.

[25] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[26] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *ICML*, 2020.

[27] D Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *ICLR*, 2015.

[28] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *NeruIPS*, 2021.

[29] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021.

[30] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2020.

[31] Xin-Chun Li, De-Chuan Zhan, Yunfeng Shao, Bingshuai Li, and Shaoming Song. Fedphp: Federated personalization with inherited private models. In *ECML*, 2021.

[32] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, Fuchun Sun, and Kunlun He. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[33] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. 2020.

[34] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel. Lanczosnet: Multi-scale deep graph convolutional networks. In *ICLR*, 2019.

[35] Clement Liu, ZewVignac, Guancheng Wan, B Aditya Prakash, Max SY Lau, and Wei Jin. Building powerful and equivariant graph neural networks with structural message-passing. *arXiv preprint arXiv:2403.19852*.

[36] Rui Liu and Han Yu. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256*, 2022.

[37] Yang Liu, Deyu Bo, and Chuan Shi. Graph condensation via eigenbasis matching. *arXiv preprint arXiv:2310.09202*, 2023.

[38] Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, and Shirui Pan. Arc: A generalist graph anomaly detector with in-context learning. In *Advances in Neural Information Processing Systems*, 2024.

[39] Yue Liu, Ke Liang, Jun Xia, Xihong Yang, Sihang Zhou, Meng Liu, Xinwang Liu, and Stan Z Li. Reinforcement graph clustering with unknown cluster number. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3528–3537, 2023.

[40] Yue Liu, Xihong Yang, Sihang Zhou, and Xinwang Liu. Simple contrastive graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[41] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. In *Proc. of AAAI*, 2023.

[42] Jun Luo and Shandong Wu. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI*, 2022.

[43] Jun Luo and Shandong Wu. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI*, pages 2166–2173, 2022.

[44] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 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. *arXiv preprint arXiv:2007.08663*, 2020.

[46] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bring order to the web. Technical report, 1998.

[47] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *NeurIPS*, 2022.

[48] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *NeurIPS*, 2024.

[49] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[50] David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing. Mag.*, 2013.

[51] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *NeurIPS*, 2017.

[52] Ruoxi Sun, Hanjun Dai, and Adams Wei Yu. Does gnn pretraining help molecular representation? In *NeurIPS*, 2022.

[53] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *NeurIPS*, 2020.

[54] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *AAAI*, 2023.

[55] Clement Vignac, Andreas Loukas, and Pascal Frossard. Building powerful and equivariant graph neural networks with structural message-passing. In *NeurIPS*, 2020.

[56] Guancheng Wan, Wenke Huang, and Mang Ye. Federated graph learning under domain shift with generalizable prototypes. In *AAAI*, 2024.

[57] Guancheng Wan, Yijun Tian, Wenke Huang, Nitesh V Chawla, and Mang Ye. S3GCL: Spectral, swift, spatial graph contrastive learning. In *ICML*, 2024.

[58] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *ICML*, 2023.

[59] Yili Wang, Yixin Liu, Xu Shen, Chenyu Li, Kaize Ding, Rui Miao, Ying Wang, Shirui Pan, and Xin Wang. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. *arXiv preprint arXiv:2406.15523*, 2024.

[60] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE TNNLS*, pages 4–24, 2020.

[61] Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. In *NeruIPS*, 2021.

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

[63] Siheng Xiong, Yuan Yang, Faramarz Fekri, and James Clayton Kerce. Tilp: Differentiable learning of temporal logical rules on knowledge graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

[64] Siheng Xiong, Yuan Yang, Ali Payani, James C Kerce, and Faramarz Fekri. Teilp: Time prediction over knowledge graphs via logical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16112–16119, 2024.

[65] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. In *NeurIPS*, pages 30414–30425, 2021.

[66] Beining Yang, Kai Wang, Qingyun Sun, Cheng Ji, Xingcheng Fu, Hao Tang, Yang You, and Jianxin Li. Does graph distillation see like vision dataset counterpart? In *NeurIPS*, 2023.

[67] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *AAAI*, 2019.

[68] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *NeurIPS*, 2021.

[69] Guixian Zhang, Shichao Zhang, and Guan Yuan. Bayesian graph local extrema convolution with long-tail strategy for misinformation detection. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–21, 2024.

[70] Huanding Zhang, Tao Shen, Fei Wu, Mingyang Yin, Hongxia Yang, and Chao Wu. Federated graph learning – a position paper. 2021.

[71] Jianqing Zhang, Yang Hua, Jian Cao, Hao Wang, Tao Song, Zhengui XUE, Ruhui Ma, and Haibing Guan. Eliminating domain bias for federated learning in representation space. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[72] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *AAAI*, 2023.

[73] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedcp: Separating feature information for personalized federated learning via conditional policy. In *ACM SIGKDD*, 2023.

[74] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. In *NeurIPS*, 2021.

[75] Yanfu Zhang, Hongchang Gao, Jian Pei, and Heng Huang. Robust self-supervised structural graph neural network for social network prediction. In *WWW*, 2022.

[76] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. Minimizing polarization and disagreement in social networks via link recommendation. In *NeurIPS*, 2021.

[77] Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. Interpreting and unifying graph neural networks with an optimization framework. In *WWW*, 2021.

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 NA 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 NA 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: [NA]

   Justification: The paper has no limitation.

   Guidelines:

   - The answer NA 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: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA 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: We fully disclose all the information needed to reproduce the main experimental results in this paper and our code. We are convinced that the obtained results can be reproduced.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: Code is accessible in this paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details are included in Sec. 5.1.

   Guidelines:

   - The answer NA 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: Statistical significance of the experiments is considered and included in Sec. 5.1.

   Guidelines:

   - The answer NA 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 sufficient information on the computer resources.

   Guidelines:

   - The answer NA 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: The research conducted in the paper strictly adheres to the NeurIPS Code of Ethics, ensuring that all aspects of the work are in compliance with the guidelines provided.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: The research presented in this paper is foundational. It is not directly tied to any specific applications or deployments.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks

    Guidelines:

    - The answer NA 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: [NA]

    Justification: This paper does not use existing assets.

    - The answer NA 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: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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.