Graph Few-Shot Learning via Adaptive Spectrum Experts and Cross-Set Distribution Calibration

Yonghao Liu[1*], **Yajun Wang**[1*], **Chunli Guo**[2*], **Wei Pang**[3], **Ximing Li**[1,4], **Fausto Giunchiglia**[5], **Xiaoyue Feng**[1†], **Renchu Guan**[1†]
[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University
[2]College of Software, Jilin University
[3]School of Mathematical and Computer Sciences, Heriot-Watt University
[4]RIKEN Center for Advanced Intelligence Project
[5]Department of Information Engineering and Computer Science, University of Trento
{yonghao20, yajun24, guocl24}@mails.jlu.edu.cn,
w.pang@hw.ac.uk, liximing86@gmail.com, fausto.giunchiglia@unitn.it,
{fengxy, guanrenchu}@jlu.edu.cn

Abstract

Graph few-shot learning has attracted increasing attention due to its ability to rapidly adapt models to new tasks with only limited labeled nodes. Despite the remarkable progress made by existing graph few-shot learning methods, several key limitations remain. First, most current approaches rely on predefined and unified graph filters (*e.g.*, low-pass or high-pass filters) to globally enhance or suppress node frequency signals. Such fixed spectral operations fail to account for the heterogeneity of local topological structures inherent in real-world graphs. Moreover, these methods often assume that the support and query sets are drawn from the same distribution. However, under few-shot conditions, the limited labeled data in the support set may not sufficiently capture the complex distribution of the query set, leading to suboptimal generalization. To address these challenges, we propose **GRACE**, a novel **G**raph few-shot lea**R**ning framework that integrates **A**daptive spectrum experts with **C**ross-s**E**t distribution calibration techniques. Theoretically, the proposed approach enhances model generalization by adapting to both local structural variations and cross-set distribution calibration. Empirically, GRACE consistently outperforms state-of-the-art baselines across a wide range of experimental settings. Our code can be found here.

1 Introduction

Graphs, as a fundamental and expressive data structure, are widely employed to model a variety of complex systems in the real world [1, 2], including social networks [3, 4], transportation networks [5, 6], and protein–protein interaction networks [7, 8]. Recently, graph neural networks (GNNs) have emerged as the *de facto* standard for learning on graph-structured data due to their powerful representation capabilities. However, the effectiveness of GNN-based models heavily relies on the availability of a large number of labeled nodes. A major challenge lies in the fact that annotating large-scale datasets is often impractical in real-world scenarios [9]. This process is not only time- and resource-intensive, but also demands extensive domain-specific expertise in certain specialized fields [10, 11]. For example, in the biomedical domain, accurately annotating unknown genes

[*]Equal Contribution
[†]Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

often requires substantial knowledge of molecular biology, which is difficult even for experienced researchers [12]. In such scenarios where labeled data are scarce, these models often suffer from severe overfitting issues [13]. Thus, graph few-shot learning (FSL) has attracted increasing attention as a promising paradigm that enables rapid adaptation to novel tasks using only a small number of labeled samples. Existing graph FSL models typically follow a two-stage paradigm [14–16]. These models first employ the graph encoder to learn low-dimensional embeddings of nodes, and then apply the few-shot learning algorithm to enable rapid generalization to new tasks. While several graph few-shot learning methods have achieved impressive results [17, 18], they still face several critical limitations that hinder their expressivity.

First, most existing graph FSL methods are grounded in either the homophily assumption (*i.e.*, nodes with the same label tend to be connected) or the heterophily assumption (*i.e.*, nodes with different labels tend to be connected) [19]. Based on these assumptions, they typically adopt predefined, uniform graph filters such as low-pass or high-pass filters [20]. This one-size-fits-all design implicitly applies global enhancement or suppression to node frequency signals. However, real-world graph data often exhibit significant local topological heterogeneity, where both homophilic and heterophilic connection patterns may coexist across different local regions of the graph [21, 22]. To substantiate our claim, we visualize the local link distribution of nodes in the Cora dataset [23]. As shown in Fig. 1, it is evident that different nodes exhibit diverse local connectivity patterns. Applying a single, globally designed filter—optimized for a specific connectivity assumption—to all nodes can lead to suboptimal performance and may adversely affect nodes whose local structures deviate from the assumed model. This naturally leads to a fundamental question: *Is it possible to develop a method that enables node-specific filtering strategies to better accommodate the diverse local structures present in real-world graphs?*

Second, these graph FSL methods implicitly assume that the support and query sets within each task are drawn from the same underlying distribution. However, this assumption is often challenged in real-world scenarios. On the one hand, the limited labeled data in the support set may fail to adequately capture the complex distribution of the query set [24]. On the other hand, the random sampling process during meta-task construction can introduce systematic biases—such as oversampling from dense subgraphs—which in turn leads to performance degradation under distribution shift conditions. The above claims are further supported by Fig. 2, where we visualize the node distributions of randomly sampled support and query sets on the Cora dataset. As shown in Fig. 2, there exists a clear distributional discrepancy between the two sets, highlighting the presence of distribution shift in practical task construction. Hence, effectively narrowing the distribution gap between the support and query sets is essential under distribution shift.

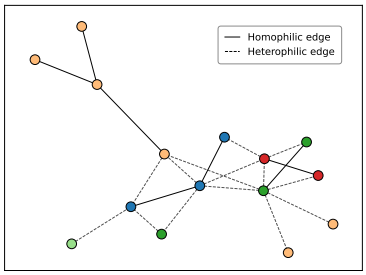


Figure 1: Diversity of local connectivity patterns in the Cora. Node colors indicate their class labels.

To address the aforementioned challenges, we propose a novel framework named **GRACE**, which integrates both adaptive spectrum experts and cross-set distribution calibration to facilitate effective graph FSL. Specifically, inspired by the mixture-of-experts (MoE) paradigm, we develop a node-specific filtering mechanism that leverages multiple experts to model diverse local connectivity patterns. Each expert is responsible for capturing a distinct graph filtering behavior, while a gating mechanism adaptively assigns expert weights based on the structural characteristics of each node. Next, to alleviate the distributional mismatch between the support and query sets, we initially derive class prototypes from the support set, which are subsequently refined through an explicit calibration process guided by the query set. Theoretically, GRACE enhances the model's generalization lower bound by incorporating adaptive spectrum experts that align with local graph structures. Empirically, it

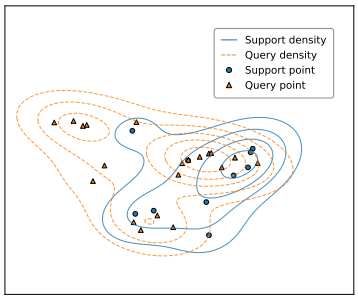


Figure 2: Visualization of distributional discrepancy between support and query sets in the Cora.

achieves substantial performance gains over competitive baselines on several standard benchmarks. In summary, our contributions are as follows.

(I) We propose a novel framework, GRACE, which integrates adaptive spectrum experts and cross-set distribution calibration to address the challenges of graph FSL.

(II) We provide theoretical analysis showing that GRACE offers improved generalization guarantees by adapting to local structural heterogeneity and mitigating distribution shift.

(III) We conduct extensive experiments on multiple benchmark datasets, demonstrating that GRACE consistently outperforms existing state-of-the-art methods.

2 Related Work

Graph Neural Networks. GNNs have become the cornerstone in the field of graph-structured data analysis, providing a powerful solution for graph representation learning [1, 25]. Typically, most GNNs follow the message passing mechanism [26], where the nodes continuously aggregate information from their neighboring nodes, gradually extracting local information. This characteristic enables GNNs based on this mechanism to perform excellently when dealing with homophilic graphs. However, when faced with heterophilic graphs, traditional GNNs are clearly inadequate. To this end, researchers have developed a series of specialized models [27, 28]. Recent studies have found that graphs in the real world often exhibit mixed structural patterns [21, 22]. However, traditional GNNs generally adopt a "one-size-fits-all" approach, applying the same global filter to all nodes. This practice cannot fully exploit the characteristics of each node when dealing with graphs with mixed patterns, and it is difficult to achieve the optimal effect. Therefore, our model introduces a node-specific adaptive filtering method, which selects an appropriate filter for each node according to its characteristics.

Few-Shot Learning. FSL aims to solve new tasks using a limited number of samples and the knowledge accumulated from previous experiences. This approach has received great attention due to its effectiveness in handling data with rare labels [29–31]. Generally speaking, the existing FSL models can be divided mainly into two categories: (i) optimization-based methods [15, 18, 10] and (ii) metric-based methods [17, 32, 11]. The former focuses on designing different mechanisms to utilize the gradients of samples. For example, the Model-Agnostic Meta-Learning algorithm (MAML) [33] proposes an inner-outer loop mechanism for gradient updates to learn good initial parameters, allowing the model to quickly adapt to new tasks with a small amount of training data. The latter aims to learn a transferable distance metric to evaluate the similarity or degree of association between given samples and query samples. For instance, Prototypical Network [34] calculates the prototype of each category by taking the mean vector of the support examples and classifies query instances by measuring the Euclidean distances between these query instances and the prototypes.

Mixture-of-Experts. The MoE architecture [35] is mainly based on the principle of "divide and conquer" [36], that is, first dividing the problem space, and then having specialized sub-models or experts handle their respective parts of the tasks. It has been widely applied in the fields of natural language processing [37, 38] and computer vision [39, 40] to improve the efficiency and performance of large-scale models. Recently, several studies [20, 41–44] in the graph domain have also explored the integration of MoE architectures to enhance graph representation learning. For example, GMoE [42] uses the MoE architecture to adaptively select the propagation hops for different nodes. According to the features of nodes and the information of neighboring nodes, it selects the most suitable propagation hops for each node through a gating mechanism. GraphMETRO [44] utilizes the MoE architecture to address the problem of graph distribution shift. Despite recent progress in applying MoE architectures to general graph learning tasks, their potential remains unexplored in graph FSL scenarios.

3 Preliminary Study

In this section, we formally define the studied problem in this work. We focus on few-shot node classification (FSNC), one of the most representative tasks in graph FSL, to evaluate the performance of our proposed model. Formally, we consider an input graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}\}$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively; $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix; and

$\mathbf{A} \in \{0,1\}^{n\times n}$ is the adjacency matrix, where $\mathbf{A}_{ij} = 1$ if there is an edge between node i and node j, and $\mathbf{A}_{ij} = 0$ otherwise. Typically, FSNC consists of two stages: meta-training and meta-testing. The label space of meta-training is denoted as $\mathcal{Y}_{\text{base}}$, and that of meta-testing as $\mathcal{Y}_{\text{new}}$, where $\mathcal{Y}_{\text{base}} \cup \mathcal{Y}_{\text{new}} = \mathcal{Y}$ and $\mathcal{Y}_{\text{base}} \cap \mathcal{Y}_{\text{new}} = \emptyset$. Moreover, we adopt the episodic training paradigm widely used in FSL by constructing a series of meta-tasks. In both in the meta-training and meta-testing phases, the construction of each meta-task follows a consistent procedure. Specifically, each meta-task consists of a support set and a query set, *i.e.*, $\mathcal{T}_t = \{\mathcal{S}_t, \mathcal{Q}_t\}$. The support set is formed by randomly sampling N classes from a particular label space $\mathcal{Y}_*$, and selecting K labeled nodes per class—yielding an N-way K-shot classification problem, *i.e.*, $\mathcal{S}_t = \{(\mathbf{X}^s_{t,i}, \mathbf{Y}^s_{t,i})\}^{N\times K}_{i=1}$. The query set is then constructed by sampling M additional nodes per class from the remaining labeled data of those same N classes, *i.e.*, $\mathcal{Q}_t = \{(\mathbf{X}^q_{t,i}, \mathbf{Y}^q_{t,i})\}^{N\times M}_{i=1}$. Note that the only difference between meta-training and meta-testing tasks lies in the label space from which classes are sampled: the former samples classes from $\mathcal{Y}_{\text{base}}$, while the latter samples from $\mathcal{Y}_{\text{new}}$. The goal of FSNC is to extract generalizable knowledge from a collection of meta-training tasks $\mathcal{T}_{\text{train}} = \{\mathcal{T}_t\}^T_{t=1}$, such that the model can swiftly adapt to a meta-testing task $\mathcal{T}_{\text{test}} = \{\mathcal{S}_{\text{test}}, \mathcal{Q}_{\text{test}}\}$ by leveraging a small support set $\mathcal{S}_{\text{test}} = \{\mathbf{X}^s_{\text{test},i}, \mathbf{Y}^s_{\text{test},i}\}^{N\times K}_{i=1}$ containing only a few labeled instances per class, and accurately predict labels for unseen nodes in the corresponding query set $\mathcal{Q}_{\text{test}} = \{\mathbf{X}^q_{\text{test},i}, \mathbf{Y}^q_{\text{test},i}\}^{N\times M}_{i=1}$.

4 Method

In this section, we provide detailed descriptions of our proposed model, GRACE, which consists of two key components: *adaptive spectrum experts* and *cross-set distribution calibration*. The former dynamically assigns expert weights for each node based on its local connectivity patterns, enabling the model to learn more discriminative node embeddings. The latter leverages class prototypes to explicitly calibrate the distribution shift between the support and query sets, thereby enhancing the model's generalization across tasks. To facilitate the better understanding of our model, we illustrate the overall framework of GRACE in Fig. 3.

4.1 Adaptive Spectrum Expert

Generally, the first step of FSNC is to learn expressive node embeddings. As previously discussed, existing graph FSL models adopt a fixed graph filter and fail to consider the diverse local connectivity patterns of individual nodes. To this end, we introduce an MoE-based architecture designed to adaptively capture different structural patterns across nodes. Given that real-world graph-structured data often exhibit either homophily or heterophily, we instantiate two experts to model these typical connectivity types: one with low-pass filtering characteristics to smooth node features under homophilic settings, and the other with high-pass filtering behavior to emphasize feature differences in heterophilic regions.

4.1.1 The Low-Pass Expert

It is widely recognized that graph convolutional networks (GCNs) [3] function as low-pass filters [45, 46], effectively capturing smooth node signals. Hence, we select GCNs as one of the experts. The core idea of GCNs is to iteratively aggregate information from the target node's neighbors to update its representation. This process can be formally expressed as:

$$\mathbf{H}^{(\ell+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}), \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with added self-loops, and $\tilde{\mathbf{D}}$ is the corresponding degree matrix. $\mathbf{H}^{(\ell)}$ and $\mathbf{W}^{(\ell)}$ denote the node embeddings at layer ℓ and the learned weight matrix, respectively. $\mathbf{H}^{(0)} = \mathbf{X}$ is the initialized original node feature when $\ell = 0$. Moreover, $\sigma(\cdot)$ is the non-linear activation function such as ReLU.

Through the low-pass expert, we can obtain the smoothed node representations $\mathbf{H}_{\text{low}} \in \mathbb{R}^{n\times d'}$ that characterize homophilic connectivity patterns.

4.1.2 The High-Pass Expert

The high-pass expert is designed to amplify the feature differences between connected nodes, thus effectively capturing heterophilic structures where nodes with dissimilar labels are more likely to be

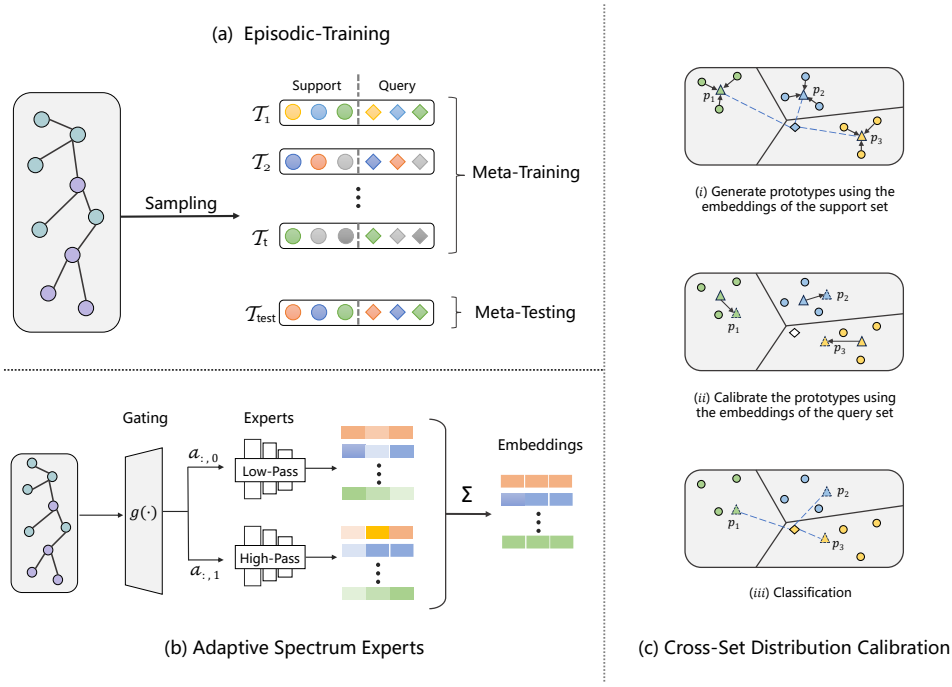


Figure 3: The overall framework of GRACE. (a) Illustration of episodic training. In each episode, an FSNC task is constructed by randomly sampling from the original graph. (b) Adaptive spectrum experts. By introducing multiple experts to capture the diverse frequency components of nodes, we employ a gating module to adaptively weight the spectrum experts. (c) Cross-set distribution calibration. We first compute class prototypes based on the support set. If classification is performed by directly assigning the query sample to the nearest prototype using Euclidean distance, it would be incorrectly assigned to the prototype p_2. However, after applying prototype calibration, the query sample can be correctly classified.

linked. To achieve this, we design the following strategy. First, we employ a linear transformation to project the original node features $\mathbf{X}$ into the feature space where smoothed representations $\mathbf{H}_{\text{low}}$ reside. We then compute the difference $\mathbf{F}$ between the original and smoothed features, which explicitly captures the local discrepancy of the node. Finally, we apply an attention mechanism over the resulting differential features, assigning higher weights to neighboring nodes that are significantly different from the target node, thereby enhancing the model's sensitivity to heterophilic connections. The above procedure can be defined as follows:

$$\mathbf{X}' = \mathbf{X}\mathbf{W}', \quad \mathbf{F} = \mathbf{X}' - \mathbf{H}_{\text{low}}, \quad \mathbf{F} = \text{LayerNorm}(\lambda \cdot \mathbf{F}), \tag{2}$$

where $\mathbf{W}' \in \mathbb{R}^{d \times d'}$ is the trainable weight. To ensure stable model training, we apply a scaling factor λ to the differential features to control their magnitude, followed by a layer normalization operation.

$$\mathbf{F}_Q = \mathbf{F}\mathbf{W}_Q, \quad \mathbf{F}_K = \mathbf{F}\mathbf{W}_K, \quad \mathbf{F}_V = \mathbf{F}\mathbf{W}_V, \quad \mathbf{H}_{\text{high}} = \text{softmax}(\frac{\mathbf{F}_Q\mathbf{F}_K^\top}{\sqrt{d'}})\mathbf{F}_V, \tag{3}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V \in \mathbb{R}^{d' \times d'}$ are the projection matrices. $\mathbf{H}_{\text{high}} \in \mathbb{R}^{n \times d'}$ is the desired high frequency feature, which measures heterophilic connectivity patterns.

4.1.3 Gating Module

To effectively integrate the outputs of the low-pass and high-pass experts, we employ a gating mechanism that adaptively assigns weights to each expert's output. Specifically, we concatenate the raw node features $\mathbf{X}$, the absolute difference between the node and its one-hop neighbors $\mathbf{N} = |\hat{\mathbf{A}}\mathbf{X} - \mathbf{X}|$, the feature-wise standard deviation ϕ of the raw node features, and the node degree

$\mathbf{D}$ to form the composite representation $\mathbf{X}_g \in \mathbb{R}^{n\times 4d}$, which is then fed into the gating module. This design enables the gating module to dynamically allocate appropriate expert weights based on each node's local topological structure. The procedure can be defined as follows:

$$\mathbf{X}_g = \mathbf{X}||\mathbf{N}||\phi||\mathbf{D}, \quad \alpha = \text{softmax}(\mathbf{X}_g\mathbf{W}_g/\tau), \quad \mathbf{Z} = \alpha_{:,0}\mathbf{H}_{\text{low}} + \alpha_{:,1}\mathbf{H}_{\text{high}}, \tag{4}$$

where $\alpha \in \mathbb{R}^{n\times 2}$ is the gating weights and $\mathbf{W}_g \in \mathbb{R}^{4d\times 2}$ is the trainable weight. τ is the temperature parameter. $\mathbf{Z} \in \mathbb{R}^{n\times d'}$ is the learned final node embeddings through the adaptive spectrum expert architecture.

4.2 Cross-Set Distribution Calibration

Due to the distributional discrepancy between the support and query sets within a meta-task, directly inferring the label of a query node as the nearest prototype in Euclidean space, as done in standard prototypical networks, can easily lead to suboptimal or erroneous decision boundaries. To this end, we propose a cross-set distribution calibration strategy. Specifically, we first compute class prototypes $\mathbf{P} \in \mathbb{R}^{n\times d'}$ based on the support set, *i.e.*, $\mathbf{P}_k = \frac{1}{K}\mathbb{I}[\mathbf{Y}_{t,i} = k]\mathbf{Z}^s_{t,i}$, where $\mathbb{I}[\cdot]$ is the indicator function. Next, inspired by the concept of kernel density estimation (KDE), we calibrate the class prototypes using samples located in high-density regions of the query distribution, which are considered to be more reliable, defined as follows:

$$\Delta = \mathbf{Z}^q_t - \mathbf{P}, \quad \Psi = \text{softmax}(\exp(-\frac{||\Delta||^2}{2\sigma^2})), \tag{5}$$

where $\Delta \in \mathbb{R}^{NM\times d'}$ represents the feature difference between query samples and class prototypes. $\Psi \in \mathbb{R}^{NM\times d'}$ denotes the weight that quantifies the contribution of the query sample to the prototype calibration. σ is the bandwidth parameter of the kernel function, which controls the smoothness of the correction process.

Next, we compute a correction vector by performing a weighted summation over the feature differences Δ, using the normalized weights Ψ. The calibrated class prototype is then obtained based on this correction vector. The process can be formulated as:

$$\Delta\mathbf{P} = \Delta \odot \Psi, \quad \hat{\mathbf{P}} = \mathbf{P} + \hat{\beta}\Delta\mathbf{P}, \tag{6}$$

where $\hat{\beta} = 0.5(\tanh(\beta) + 1)$ is a trainable parameter that controls the magnitude of prototype calibration, in which β is a predefined scalar value.

4.3 Model Optimization

After applying the adaptive spectral experts and the cross-set distribution calibration, we adopt the classical metric-based episodic training paradigm for FSNC. Specifically, we optimize the model parameters by minimizing a distance-based cross-entropy loss computed over the query sets of all meta-training tasks in $\mathcal{T}_{\text{train}}$. The objective can be formulated as follows:

$$\mathcal{L} = -\sum_{t=1}^{T}\sum_{i=1}^{NM}\mathbb{I}[\mathbf{Y}_{t,i} = k]\log\frac{\exp(-e(\mathbf{Z}^q_{t,i}\mathbf{W}_l, \hat{\mathbf{P}}_k))}{\sum_{k'}\exp(-e(\mathbf{Z}^q_{t,i}\mathbf{W}_l, \hat{\mathbf{P}}_{k'}))}, \tag{7}$$

where $e(\cdot,\cdot)$ is the Euclidean function and $\mathbf{W}_l$ denotes the trainable vector.

In the meta-testing phase, we also compute the calibrated prototypes using the same strategy as described in Eqs.5 and 6. Then, each query instance is assigned to the class of the nearest calibrated prototype based on Euclidean distance. Formally, the predicted label $\mathbf{Y}^q_i$ for a query instance is defined as follows:

$$\mathbf{P}_k = \frac{1}{|\mathcal{S}_{\text{test},k}|}\sum_{(\mathbf{Z}^s_{\text{test},i}, \mathbf{Y}^s_{\text{test},i})}\mathbb{I}[\mathbf{Y}_{\text{test},i} = k]\mathbf{Z}^s_{\text{test},i}, \ \hat{\mathbf{P}} = \text{Calibration}(\mathbf{P}), \ \mathbf{Y}^q_{\text{test},*} = \text{argmin}_k e(\mathbf{Z}\mathbf{W}_l, \mathbf{P}_k). \tag{8}$$

We present the detailed training procedure and the complexity analysis of GRACE in **Appendices** A.1 and A.2.

5 Theoretical Analysis

In this section, we theoretically analyze the effectiveness of the proposed model. Specifically, we present the following theorem to establish the connection between the model's generalization error and the employed techniques.

Theorem 5.1. *Suppose that the loss function $\mathcal{L}$ is L-Lipschitz continuous, and for $\epsilon_g > 0$, the gating module satisfies $\mathbb{E}_{v\sim P_{\mathcal{V}}}\left[|\alpha_v - \mathbb{I}(d_v^{\text{hom}} > d_v^{\text{het}})|\right] \leq \epsilon_g$, where α_v denotes the gating weight for node v, and d_v^{hom}, d_v^{het} measure the degrees of homophily and heterophily, respectively. Furthermore, for $\delta > 0$, we assume that the Wasserstein distance between the support and query distributions satisfies $W_1(P_{\mathcal{S}}, P_{\mathcal{Q}}) \leq \delta$. Then the generalization error ϵ_{gen} of the proposed model is bounded by:*

$$\epsilon_{gen} \leq C_1\sqrt{\frac{\log T}{T}} + C_2\epsilon_g + C_3\left(\delta + \mathcal{O}(\sigma^2) + \mathcal{O}(|\mathcal{Q}|^{-1/2})\right), \tag{9}$$

where C_1, C_2, and C_3 are the constants. T and $|\mathcal{Q}|$ are the number of training tasks and query samples. σ is the bandwidth.

Theorem 5.1 indicates that, given a fixed number of training tasks and query samples, our model achieves a tighter generalization error bound compared to that of standard approaches. This improvement is attributed to the reduction of the discrepancy measures ϵ_g and δ, which is accomplished through the use of adaptive spectrum experts and the cross-domain distribution calibration strategy.

Moreover, we can derive the following corollary to further illustrate the advantage of the adaptive spectral experts.

Corollary 5.2. *When local topology exhibits strong heterogeneity ($\epsilon_g \to 0$), our model achieves strictly better bound ϵ_{gen}^{MoE} than that of single-filter methods ϵ_{gen}^{Sin}:*

$$\epsilon_{gen}^{MoE} \leq \epsilon_{gen}^{Sin} + \mathcal{O}(|\mathcal{Q}|^{-1/2}). \tag{10}$$

Corollary 5.2 indicates that the generalization error of GRACE is clearly lower than that of models using a single filter. The proofs of Theorem 5.1 and Corollary 5.2 can be found in **Appendix** A.3.

6 Experiments

6.1 Datasets

To empirically validate the effectiveness of our proposed model, we utilize several widely adopted datasets for FSNC tasks, including **Cora** [23], **CiteSeer** [23], **Amazon-Computer** [47], **Coauthor-CS** [47], **DBLP** [48], **CoraFull** [49], and, a large-scale dataset, **ogbn-arxiv** [50]. The statistics of these datasets are presented in Table 1. We present detailed descriptions of these adopted datasets in **Appendix** A.4.

6.2 Baselines

To comprehensively evaluate the effectiveness of the proposed model, we compare it against the following three representative categories of baselines. *Graph embedding methods* contain **DeepWalk** [51], **node2vec** [52], **GCN** [3], and **SGC** [45]. *Meta-learning methods* include **Prototypical Network** (**ProtoNet**) [34] and **MAML** [33]. *Graph meta-learning methods* consist of **GPN** [17], **G-Meta** [18], **TENT** [16], **Meta-GPS** [15], **TEG** [53], **COSMIC** [54], and **Meta-BP** [55]. Detailed descriptions of these baselines are presented in **Appendix** A.5.

Table 1: Statistics of the evaluated datasets.

Dataset	#Nodes	#Edges	#Features	#Labels
Cora	2,708	5,278	1,433	7
CiteSeer	3,327	4,552	3,703	6
Amazon-Computer	13,381	245,778	767	10
Coauthor-CS	18,333	81,894	6,805	15
DBLP	40,672	144,135	7,202	137
CoraFull	19,793	65,311	8,710	70
ogbn-arxiv	169,343	1,166,243	128	40

6.3 Implementation Details

In the stage of *adaptive spectrum experts*, the low-pass expert is implemented using a two-layer GCN, with each layer followed by batch normalization and a ReLU activation. For the high-pass

expert, the dimensions of all projection matrices are uniformly set to 32, *i.e.*, $d' = 32$. The gating network is implemented as a two-layer fully connected network, with the hidden dimension of 96. The temperature τ in Eq.4 is set to 2. Additionally, in the *cross-set distribution calibration* stage, the Gaussian kernel bandwidth σ is set to 1. During training, we use the Adam optimizer [56] with an initial learning rate of 0.001. For evaluation, we randomly generate multiple meta-testing tasks from the test set. Specifically, 100 tasks are sampled per evaluation, with each task comprising 10 query samples. To ensure the fairness and stability of our results, we conduct 5 independent experiments and report the average accuracy, standard deviation, and 95% confidence interval across these runs. All experiments are carried out on an NVIDIA 3090Ti GPU to maintain consistent computational conditions and reproducibility.

7 Results

Model Performance. We conduct extensive experiments across various few-shot settings on multiple datasets. As shown in Tables 2, 3, and 4, our proposed model GRACE consistently achieves superior performance under all experimental configurations, demonstrating its effectiveness for graph FSL compared to other competitive baselines. We attribute the performance improvements to the two key components introduced in our model. The adaptive spectrum experts module leverages an MoE architecture to assign different weights to high-pass and low-pass experts based on the local topology of each target node. This design mitigates the limitations of using a single graph filter, which may fail to accommodate diverse structural patterns. Moreover, the cross-set distribution calibration module leverages the distributional characteristics of high-density samples in the query set to explicitly calibrate the support-set prototypes, effectively narrowing the support–query distribution gap and improving the discriminative power of the classification boundary.

Moreover, it is evident that graph meta-learning models significantly outperform other types of baselines, owing to their tailored designs for addressing the challenges inherent in graph FSL tasks. In contrast, graph embedding methods and conventional meta-learning models exhibit unsatisfactory performance. This performance gap can be attributed to two main reasons: the former lacks mechanisms to cope with the scarcity of labeled nodes and is prone to overfitting, while the latter fails to exploit the inherent structural information of graphs.

Table 2: Accuracies (%) of different models on the three datasets.

Model	Cora			CiteSeer			Amazon-Computer		
	2 way 1 shot	2 way 3 shot	2 way 5 shot	2 way 1 shot	2 way 3 shot	2 way 5 shot	2 way 1 shot	2 way 3 shot	2 way 5 shot
DeepWalk	32.95±2.70	36.70±2.99	41.51±2.70	39.56±2.79	39.72±3.42	43.22±3.19	46.49±2.35	49.29±2.46	51.24±2.72
node2vec	31.17±3.16	35.66±2.79	40.69±2.90	40.12±3.15	42.39±2.79	47.20±2.92	49.25±2.56	51.46±2.25	53.49±2.69
GCN	55.46±2.16	69.96±2.52	67.95±2.36	51.95±2.45	53.79±2.39	55.76±2.56	60.16±2.20	63.46±2.16	67.39±2.46
SGC	56.75±2.31	70.15±1.99	70.67±2.11	53.72±2.55	55.12±2.59	57.25±2.79	61.29±2.45	65.39±2.06	69.35±2.12
ProtoNet	50.39±2.52	52.67±2.28	57.92±2.34	49.15±2.29	52.19±2.96	53.75±2.49	57.15±2.55	60.49±2.09	65.12±2.69
MAML	52.40±2.29	55.07±2.36	57.39±2.23	49.15±2.25	52.75±2.75	54.36±2.39	53.72±2.25	59.20±2.55	61.20±2.59
Meta-GNN	58.82±2.56	70.40±2.64	72.51±1.91	55.45±2.15	59.71±2.79	61.32±3.22	62.36±2.70	67.49±2.11	70.15±2.16
GPN	60.12±2.12	74.05±1.96	76.39±2.33	57.36±2.20	64.22±2.92	65.59±2.49	65.56±2.60	72.19±2.30	76.19±2.21
G-Meta	59.72±3.15	74.39±2.69	80.05±1.98	54.39±2.19	57.59±2.42	62.49±2.30	64.56±3.10	69.49±2.42	73.50±2.92
TENT	55.39±2.16	58.25±2.23	66.75±2.19	60.03±3.11	65.20±3.19	67.59±2.95	80.75±2.95	85.32±2.10	89.22±2.16
Meta-GPS	62.19±2.12	80.29±2.15	83.79±2.10	58.95±2.12	69.95±2.02	72.56±2.06	82.12±2.55	87.10±2.65	90.16±2.05
X-FNC	61.47±2.99	78.19±3.25	82.70±3.19	58.79±2.56	67.96±3.10	70.29±3.05	81.50±2.29	86.39±2.29	90.25±2.26
TEG	62.52±2.95	80.65±1.53	84.50±2.01	59.70±2.69	73.79±1.59	76.79±2.12	86.49±2.10	89.02±2.57	92.40±2.05
COSMIC	63.16±2.47	65.37±2.49	69.10±2.30	60.95±2.75	70.22±2.56	75.10±2.30	85.49±2.46	88.26±2.02	91.59±2.59
TLP	60.19±2.25	71.10±1.66	85.15±2.19	61.12±2.10	71.10±2.17	75.55±2.03	83.35±2.07	89.49±2.06	92.09±2.12
Meta-BP	66.42±4.12	76.32±4.30	83.12±4.16	60.15±2.45	72.19±3.19	76.11±3.29	86.10±4.10	89.22±4.29	92.39±4.45
GRACE	**66.48±2.88**	**82.40±2.03**	**86.19±1.80**	**63.90±2.84**	**75.67±2.44**	**79.64±1.79**	**90.23±0.90**	**92.46±0.55**	**94.66±0.50**

Ablation Study. To validate the effectiveness of the adopted strategies, we design multiple model variants under different few-shot settings on different datasets. (I) *w/o high*: We exclude the high-pass expert. (II) *w/o low*: We discard the low-pass expert. (III) *w/o cal*: We eliminate cross-set distribution calibration. (IV) *w/o both*: We omit both adaptive spectrum expert and cross-set distribution calibration, resulting in a variant that follows the standard training paradigm of graph meta-learning models. We present the ablation results in Table 5, with additional results provided in **Appendix** A.6.

Based on the results, we have the following in-depth analysis. (I) Our proposed GRACE outperforms all four variants, which validates the necessity of the proposed modules. (II) It can be observed that the variant without the cross-set distribution calibration generally exhibits inferior performance.

Table 3: Accuracies (%) of different models on the two datasets.

Model	Coauthor-CS				DBLP			
	2 way 3 shot	2 way 5 shot	5 way 3 shot	5 way 5 shot	5 way 3 shot	5 way 5 shot	10 way 3 shot	10 way 5 shot
DeepWalk	59.52±2.72	63.12±3.12	33.76±3.21	40.15±2.96	49.12±2.25	59.12±2.32	37.11±2.19	49.16±2.39
node2vec	56.16±4.19	60.22±4.06	30.35±3.93	39.16±3.79	45.65±2.79	55.92±2.36	35.72±2.52	46.19±2.75
GCN	73.52±1.97	77.20±3.01	52.19±2.31	56.35±2.99	64.12±2.15	67.26±2.39	42.16±2.39	56.12±2.10
SGC	75.49±2.15	79.63±2.01	56.39±2.26	59.25±2.16	66.32±2.25	70.19±2.36	40.19±2.26	55.16±2.56
ProtoNet	71.18±3.82	75.51±3.19	47.71±3.92	51.66±2.51	59.95±2.56	62.95±2.72	32.35±1.62	52.95±1.90
MAML	62.32±4.60	65.20±4.20	36.99±4.32	42.12±2.43	55.05±2.30	60.67±2.41	29.59±2.90	40.22±2.61
Meta-GNN	85.76±2.74	87.86±4.79	75.87±3.88	68.59±2.59	73.41±3.20	77.95±3.12	65.22±2.79	69.12±2.51
GPN	85.60±2.15	88.70±2.21	75.88±2.75	81.79±3.18	75.39±3.41	79.90±2.62	67.20±2.40	71.12±1.87
G-Meta	92.14±3.90	93.90±3.18	75.72±3.59	74.18±3.29	76.49±3.29	80.12±2.46	68.95±2.70	72.19±2.11
TENT	89.35±4.49	90.90±4.24	78.38±5.21	78.56±4.42	78.22±2.10	81.30±2.02	69.52±2.16	73.20±1.95
Meta-GPS	90.16±2.72	92.39±1.66	81.39±2.35	83.66±1.79	79.12±1.92	81.66±2.16	70.16±2.20	73.59±1.26
X-FNC	90.95±4.29	92.03±4.14	82.93±2.02	84.36±3.49	77.45±2.39	80.69±2.52	69.72±2.39	72.95±1.76
TEG	92.36±1.59	93.02±1.24	80.78±1.40	84.70±1.42	79.26±2.49	82.19±2.40	72.49±2.12	73.99±2.55
COSMIC	89.35±4.49	93.32±1.93	78.38±5.21	85.47±1.42	78.34±2.06	81.81±2.05	66.53±1.54	70.09±1.53
TLP	90.35±4.49	90.90±4.24	82.30±2.05	78.56±4.42	77.49±3.22	81.95±2.39	71.49±2.35	73.16±2.30
Meta-BP	91.19±2.21	92.32±2.11	81.35±2.02	82.12±2.15	78.22±2.10	81.13±2.55	71.30±2.12	73.15±2.39
GRACE	**95.50±1.30**	**96.20±0.97**	**86.03±1.05**	**86.82±1.01**	**81.72±2.05**	**85.30±1.90**	**74.22±1.56**	**76.70±1.46**

Table 4: Accuracies (%) of different models on the two datasets.

Model	CoraFull				ogbn-arxiv			
	5 way 3 shot	5 way 5 shot	10 way 3 shot	10 way 5 shot	5 way 3 shot	5 way 5 shot	10 way 3 shot	10 way 5 shot
DeepWalk	23.62±3.99	25.93±3.45	15.32±4.12	17.03±3.73	24.12±3.16	26.16±2.95	20.19±2.35	23.76±3.02
node2vec	23.75±2.93	25.42±3.61	13.90±3.32	15.21±2.64	25.29±2.96	27.39±2.56	22.99±3.15	25.95±3.12
GCN	34.65±2.76	39.83±2.49	29.23±3.25	34.14±2.15	32.26±2.11	36.29±2.39	30.21±1.95	33.96±1.59
SGC	39.56±3.52	44.53±2.92	35.12±2.71	39.53±3.32	35.19±2.76	39.76±2.95	31.99±2.12	35.22±2.52
ProtoNet	33.67±2.51	36.53±3.76	24.90±2.03	27.24±2.67	39.99±3.28	47.31±1.71	32.79±2.22	37.19±1.92
MAML	37.12±3.16	47.51±3.09	26.61±2.19	31.60±2.91	28.35±1.49	29.09±1.62	30.19±2.97	36.19±2.29
Meta-GNN	52.23±2.41	59.12±2.36	47.21±3.06	53.32±3.15	40.14±1.94	45.52±1.71	35.19±1.72	39.02±1.99
GPN	53.24±2.33	60.31±2.19	50.93±2.30	56.21±2.09	42.81±2.34	50.50±2.13	37.36±1.99	42.16±2.19
G-Meta	57.52±3.91	62.43±3.11	53.92±2.91	58.10±3.02	40.48±1.70	47.16±1.73	35.49±2.12	40.95±2.70
TENT	64.80±4.10	69.24±4.49	51.73±4.34	56.00±3.53	50.26±1.73	61.38±1.72	42.19±1.16	46.29±1.29
Meta-GPS	65.19±2.35	69.25±2.52	61.23±3.11	64.22±2.66	52.16±2.01	62.55±1.95	42.96±2.02	46.86±2.10
X-FNC	69.32±3.10	71.26±4.19	49.63±4.45	53.00±3.93	52.36±2.75	63.19±2.22	41.92±2.72	46.10±2.16
TEG	72.14±1.06	76.20±1.39	61.03±1.13	65.56±1.03	57.35±1.14	62.07±1.72	47.41±0.63	51.11±0.73
COSMIC	73.03±1.78	77.24±1.52	65.79±1.36	70.06±1.93	52.98±2.19	65.42±1.69	43.19±2.72	47.59±2.19
TLP	66.32±2.10	71.36±4.49	51.73±4.34	56.00±3.53	41.96±2.29	52.99±2.05	39.42±2.15	42.62±2.09
Meta-BP	72.90±1.90	74.36±2.19	62.35±2.27	67.26±2.59	55.12±4.12	65.39±4.55	46.25±4.52	50.12±3.39
GRACE	**78.22±1.38**	**81.60±1.28**	**70.91±1.08**	**74.54±0.98**	**62.31±1.94**	**68.34±1.73**	**50.18±1.01**	**55.07±0.91**

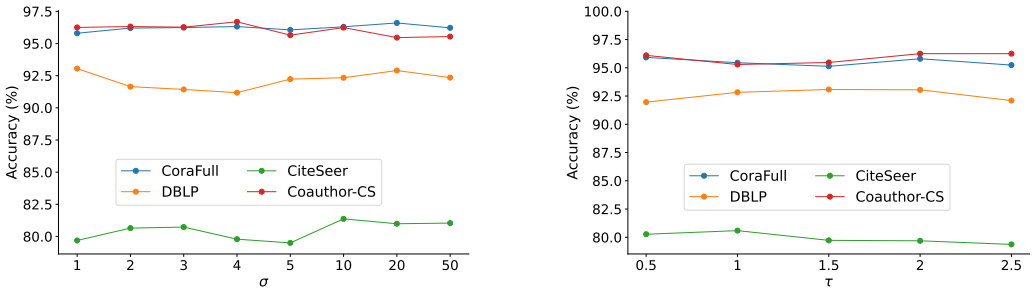
This validates that this strategy can succeed in reducing the distributional discrepancy between the support and query sets within the meta-task. (III) Since real-world graphs often exhibit diverse local connectivity patterns, relying solely on a high-pass or low-pass expert leads to suboptimal performance, highlighting the advantage of GRACE's adaptive combination of both.

Table 5: Results of different model variants on all datasets.

Model	Cora	CiteSeer	Amazon-Computer	Coauthor-CS	DBLP	CoraFull	ogbn-arxiv
	2 way 1 shot	2 way 1 shot	2 way 1 shot	5 way 3 shot	5 way 5 shot	5 way 3 shot	5 way 3 shot
w/o high	64.01±2.67	62.26±2.60	82.89±2.13	79.05±1.23	79.35±2.00	76.73±1.45	59.32±1.90
w/o low	63.94±2.79	59.64±2.75	90.08±0.79	80.31±1.14	85.05±1.83	77.23±1.49	61.98±1.92
w/o cal	65.66±2.80	58.61±2.58	89.58±1.02	85.97±1.13	83.52±1.89	77.18±1.45	61.84±1.96
w/o both	60.12±2.12	55.36±2.20	65.56±2.60	75.88±2.75	79.90±2.62	53.24±2.33	42.81±2.34
Ours	**66.48±2.88**	**63.90±2.84**	**90.23±0.90**	**86.03±1.05**	**85.30±1.90**	**78.22±1.38**	**62.31±1.94**

Hyperparameter Sensitivity. We primarily analyze the impact of two crucial hyperparameters—the bandwidth σ and the gating temperature τ—on model performance under the 2 way 5 shot experimental setting across several datasets, as shown in Figs. 4(a) and 4(b). We observe that the model performance with respect to the bandwidth parameter σ generally exhibits a rise-then-fall trend. When σ is too small, the model only leverages a limited number of neighboring samples. Conversely, an excessively large σ introduces many irrelevant or noisy nodes, thereby impairing the model's discriminative capability. Regarding the gating temperature τ, the model performance remains relatively stable across different values. A plausible explanation is that within the explored

range, whether the softmax distribution becomes slightly sharper or smoother, both the low-pass and high-pass experts maintain sufficient representational capacity.

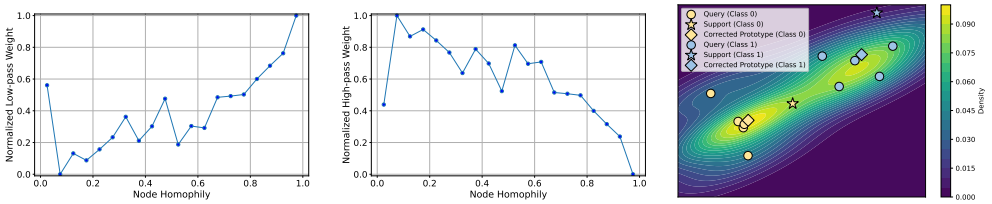


(a) (b)

Figure 4: Hyperparameter sensitivity analysis: (a) Model performance varies with the bandwidth σ. (b) Model performance varies with the gating temperature τ.

Case Study. We visualize the weights learned by the gating module to verify whether the proposed adaptive spectral expert strategy can effectively assign different weights to the experts based on varying local connectivity patterns. Specifically, on the Coauthor-CS dataset, nodes are grouped into 20 equal-width bins according to their homophily scores d_v^{hom}. For each bin, we compute and plot the normalized mean weights assigned to the low-pass expert α_{low} and the high-pass expert α_{high} (Figs. 5(a) and 5(b)). The results show a clear trend: as node homophily increases, the weight allocated to the low-pass expert steadily increases, while that of the high-pass expert decreases accordingly. This behavior is consistent with our intuition and provides empirical evidence that the proposed adaptive spectral expert strategy effectively captures the local structural patterns of nodes.

Moreover, we visualizes the cross-set distribution calibration strategy under the 2-way 1-shot setting in Fig. 5(c). We observe that the uncalibrated prototypes coincide with the corresponding support samples, which often lie in sparse regions relative to the query samples, potentially leading to misclassification. In contrast, the calibrated prototypes are shifted toward the dense regions of their corresponding query points, bringing them closer to the query cluster centers.



(a) (b) (c)

Figure 5: (a) Normalized low-pass expert weight across node homophily d_v^{hom}. (b) Normalized high-pass expert weight across node homophily d_v^{hom}. (c) Cross-set distribution calibration via KDE contours, showing support (stars), query (circles), and corrected prototypes (diamonds).

8 Conclusion

In this work, we propose a novel model, named GRACE for graph FSL. Specifically, our model incorporates two key techniques. First, an adaptive spectral expert strategy is employed to assign different weights to multiple experts based on the diverse local connectivity patterns of nodes, thereby learning expressive node embeddings. Second, a cross-set distribution calibration strategy is introduced to mitigate the distributional shift between the support and query sets, enabling the model to establish more accurate decision boundaries. Theoretically, GRACE offers stronger generalization guarantees by adapting to local structural heterogeneity and mitigating distributional shifts. Empirically, GRACE consistently outperforms other competitive models across multiple benchmark datasets.

Acknowledgments

Our work is supported by the National Science and Technology Major Project under Grant No. 2021ZD0112500, the National Natural Science Foundation of China (No. 62172187 and No. 62372209). Fausto Giunchiglia's work is funded by European Union's Horizon 2020 FET Proactive Project (No.823783).

References

[1] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[2] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE TNNLS*, 32(1):4–24, 2020.

[3] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[4] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. In *ICLR*, 2018.

[5] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *WWW*, pages 1082–1092, 2020.

[6] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *CIKM*, pages 3767–3776, 2021.

[7] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *SIGKDD*, pages 975–985, 2021.

[8] Andrea Mastropietro, Giuseppe Pasculli, and Jürgen Bajorath. Learning characteristics of graph neural networks predicting protein–ligand affinities. *Nature Machine Intelligence*, 5(12): 1427–1436, 2023.

[9] Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s):1–40, 2023.

[10] Yonghao Liu, Mengyu Li, Ximing Li, Lan Huang, Fausto Giunchiglia, Yanchun Liang, Xiaoyue Feng, and Renchu Guan. Meta-gps++: Enhancing graph meta-learning with contrastive learning and self-training. *ACM TKDD*, 18(9):1–30, 2024.

[11] Yonghao Liu, Fausto Giunchiglia, Ximing Li, Lan Huang, Xiaoyue Feng, and Renchu Guan. Enhancing unsupervised graph few-shot learning via set functions and optimal transport. In *SIGKDD*, 2025.

[12] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

[13] Zhen Tan, Song Wang, Kaize Ding, Jundong Li, and Huan Liu. Transductive linear probing: a novel framework for few-shot node classification. In *LoG*, 2022.

[14] Yonghao Liu, Mengyu Li, Fausto Giunchiglia, Lan Huang, Ximing Li, Xiaoyue Feng, and Renchu Guan. Dual-level mixup for graph few-shot learning with fewer tasks. In *WWW*, 2025.

[15] Yonghao Liu, Mengyu Li, Ximing Li, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. Few-shot node classification on attributed networks with graph meta-learning. In *SIGIR*, pages 471–481, 2022.

[16] Song Wang, Kaize Ding, Chuxu Zhang, Chen Chen, and Jundong Li. Task-adaptive few-shot node classification. In *SIGKDD*, 2022.

[17] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *CIKM*, 2020.

[18] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In *NeurIPS*, 2020.

[19] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.

[20] Haoyu Han, Juanhui Li, Wei Huang, Xianfeng Tang, Hanqing Lu, Chen Luo, Hui Liu, and Jiliang Tang. Node-wise filtering in graph neural networks: A mixture of experts approach. *arXiv preprint arXiv:2406.03464*, 2024.

[21] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *ICML*, pages 13242–13256, 2022.

[22] Haitao Mao, Zhikai Chen, Wei Jin, Haoyu Han, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Demystifying structural disparity in graph neural networks: Can one size fit all? In *NeurIPS*, pages 37013–37067, 2023.

[23] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

[24] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *ICLR*, 2021.

[25] Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. Deep attention diffusion graph neural networks for text classification. In *EMNLP*, pages 8142–8152, 2021.

[26] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[27] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *NeurIPS*, pages 20887–20902, 2021.

[28] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, pages 21–29, 2019.

[29] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11–20, 2019.

[30] Shunyu Jiang, Fuli Feng, Weijian Chen, Xiang Li, and Xiangnan He. Structure-enhanced meta-learning for few-shot graph classification. *AI Open*, 2:160–167, 2021.

[31] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, pages 3394–3403, 2021.

[32] Yonghao Liu, Lan Huang, Bowen Cao, Ximing Li, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. A simple but effective approach for unsupervised few-shot graph classification. In *WWW*, 2024.

[33] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[34] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[35] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[36] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[37] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *ICML*, pages 5547–5569, 2022.

[38] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. In *NeurIPS*, volume 35, pages 7103–7114, 2022.

[39] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. In *NeurIPS*, pages 9564–9576, 2022.

[40] Tianlong Chen, Xuxi Chen, Xianzhi Du, Abdullah Rashwan, Fan Yang, Huizhong Chen, Zhangyang Wang, and Yeqing Li. Adamv-moe: Adaptive multi-task vision mixture-of-experts. In *ICCV*, pages 17346–17357, 2023.

[41] Fenyu Hu, Liping Wang, Shu Wu, Liang Wang, and Tieniu Tan. Graph classification by mixture of diverse experts. In *IJCAI*, 2021.

[42] Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. In *NeurIPS*, pages 50825–50837, 2023.

[43] Hanqing Zeng, Hanjia Lyu, Diyi Hu, Yinglong Xia, and Jiebo Luo. Mixture of weak & strong experts on graphs. In *ICLR*, 2023.

[44] Shirley Wu, Kaidi Cao, Bruno Ribeiro, James Y Zou, and Jure Leskovec. Graphmetro: Mitigating complex graph distribution shifts via mixture of aligned experts. In *NeurIPS*, pages 9358–9387, 2024.

[45] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019.

[46] Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

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

[48] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*, pages 990–998, 2008.

[49] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

[50] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

[51] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014.

[52] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

[53] Sungwon Kim, Junseok Lee, Namkyeong Lee, Wonjoong Kim, Seungyoon Choi, and Chanyoung Park. Task-equivariant graph few-shot learning. In *SIGKDD*, 2023.

[54] Song Wang, Zhen Tan, Huan Liu, and Jundong Li. Contrastive meta-learning for few-shot node classification. In *SIGKDD*, pages 2386–2397, 2023.

[55] Qiannan Zhang, Shichao Pei, Yuan Fang, and Xiangliang Zhang. Unlocking the potential of black-box pre-trained gnns for graph few-shot learning. In *AAAI*, pages 22497–22505, 2025.

[56] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[57] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research*, 13(1):1865–1890, 2012.

[58] Michel Talagrand. *Upper and lower bounds for stochastic processes*, volume 60. 2014.

[59] Cédric Villani et al. *Optimal transport: old and new*, volume 338. 2008.

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 abstract and introduction clearly outline the main contributions and scope of our work.

   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: [Yes]

   Justification: In Section A.7, we discuss the limitations of our model.

   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: [Yes]

Justification: All theoretical results are accompanied by complete proofs, which are detailed in Section 5 and Appendix A.3.

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: The two core components of the model are thoroughly introduced in Section 4, while the complete implementation details are provided in Section 6.3.

   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: Our data and code are available at (`https://anonymous.4open.science/r/GRACE-7E41`)

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: We provide detailed descriptions of all training and test settings, including hyperparameters and optimization strategies. Additionally, implementation details are included in Section 6.3.

   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: To ensure the fairness and stability of our results, we conduct 5 independent experiments and report the average accuracy, standard deviation, and 95% confidence interval across these runs.

   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 have provided the information on our GPUs used for training in Section 6.3.

   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: Our research fully complies with all requirements of the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: In Section A.8, we discuss the broader impacts of our model.

    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: Our work 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: [Yes]

    Justification: We have cited all the sources we used to conduct the experiments.

    Guidelines:

    - 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: [Yes]

    Justification: In the abstract, we provide an anonymous GitHub link to offer open access to our code and data.

    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: Our work 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: Our work 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.

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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: In our work, the LLM is used only for writing and editing.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Appendix

A.1 Training Procedure

Algorithm 1 Training Procedure of GRACE

Require: A graph $\mathcal{H} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}\}$.
Ensure: The trained GRACE.
1: Perform the low-pass expert using Eq.1.
2: Perform the high-pass expert using Eqs.2 and 3.
3: Intergrate the outputs of low-pass and high-pass experts using Eq.4.
4: Perform cross-set distribution calibration using Eqs.5 and 6.
5: Optimize the proposed model by minimizing the loss in Eq.7.
6: Evaluate the model performance using query set with Eq.8.

We present the detailed training procedure of GRACE in Algorithm 1.

A.2 Complexity Analysis

In this section, we provide a detailed analysis of the time complexity of the proposed model. During the graph feature extraction phase, the primary computational cost arises from the low-pass branch and the high-pass expert's forward propagation. The low-pass branch employs two graph convolutional layers implemented using sparse matrix multiplications, resulting in an approximate complexity of $\mathcal{O}(2|E| \cdot d')$, where $|E|$ denotes the number of edges and d' represents the dimensionality of the hidden units. In the high-pass expert module, aside from an initial linear projection, a sparse self-attention computation is performed on all edges, incurring a complexity of $\mathcal{O}(|E| \cdot d')$. Furthermore, the gating network integrates the raw node features with their first- and second-order neighborhood differences via fully-connected mappings, and its forward pass operates with a complexity of $\mathcal{O}(n \cdot d)$, where n is the total number of nodes and d is the input feature dimension. Regarding the few-shot prototype correction mechanism, the complexity is primarily determined by the Euclidean distance computations and subsequent weighted corrections between the support and query samples, which is approximately $\mathcal{O}(N_Q \cdot d')$, with N_Q denoting the number of query samples. Additionally, the supervised contrastive loss employed in the model necessitates concatenating, normalizing, and computing the similarity matrix for the features of both support and query samples, leading to an overall complexity of $\mathcal{O}((N_S + N_Q)^2 \cdot d')$, where N_S represents the number of samples in the support set. In summary, the overall time complexity of the model is chiefly influenced by the number of edges, nodes, and the feature dimensions; by employing sparse matrix operations and localized sampling strategies, we have effectively reduced the computational burden, ensuring that the model's runtime efficiency remains within acceptable bounds for practical applications.

A.3 Theoretical Proofs

A.3.1 Proof of Theorem 5.1

Before formally proving Theorem 5.1, we first present several key definitions.

1. **(Lipschitz Continuity)** The loss function $\mathcal{L} : \mathbb{R}^N \times \mathcal{Y} \to \mathbb{R}^+$ is L-Lipschitz continuous if the following is satisified:

$$|\mathcal{L}(\mathbf{Z}, y) - \mathcal{L}(\mathbf{Z}', y)| \leq L\|\mathbf{Z} - \mathbf{Z}'\|_2, \quad \forall y \in \mathcal{Y}.$$

2. **(Heterogenity Balance)** For node $v \in \mathcal{V}$, define homophily degree $d_v^{\text{hom}} = |\{(u, v) \in \mathcal{E} : y_u = y_v\}|$ and heterophily degree $d_v^{\text{het}} = |\mathcal{E}_v| - d_v^{\text{hom}}$. For $\epsilon_g > 0$, the gating network satisfies:

$$\mathbb{E}_{v \sim P_{\mathcal{V}}} \left[|\alpha_v - \mathbb{I}(d_v^{\text{hom}} > d_v^{\text{het}})|\right] \leq \epsilon_g.$$

3. **(Distribution Shift)** For $\delta > 0$, the Wasserstein-1 distance between support set distribution $P_{\mathcal{S}}$ and query set distribution $P_{\mathcal{Q}}$ satisfies:

$$W_1(P_{\mathcal{S}}, P_{\mathcal{Q}}) \leq \delta.$$

The complete error decomposition is:

$$\epsilon_{gen} = \underbrace{\epsilon_{MoE}}_{\text{Expert Variance}} + \underbrace{\epsilon_{Gate}}_{\text{Gating Error}} + \underbrace{\epsilon_{Dist}}_{\text{Distribution Shift}}. \tag{11}$$

Next, we present the following lemmas to assist the proof of Theorem 5.1.

Lemma A.1 (Expert Variance Bound)**.** *Let $\mathcal{F}_{MoE}$ be the hypothesis class of the adaptive spetrum experts module. The expected error from expert diversity satisfies:*

$$\epsilon_{MoE} \leq 2L\mathfrak{R}_N(\mathcal{F}_{MoE}) \leq C_1\sqrt{\frac{\log T}{T}},$$

where $\mathfrak{R}_N$ is the Rademacher complexity.

Proof. The proof follows three key steps:

Step 1: Define Empirical Rademacher Complexity. For T *i.i.d.* tasks $\{\mathcal{T}_i\}_{i=1}^T$ and Rademacher variables $\beta_i \in \{\pm 1\}$:

$$\mathfrak{R}_T(\mathcal{F}_{MoE}) = \mathbb{E}\left[\sup_{f\in\mathcal{F}_{MoE}} \frac{1}{T}\sum_{i=1}^{N} \beta_i f(\mathcal{T}_i)\right].$$

Step 2: Apply Rademacher Bound for Ensembles. Using the theorem in [57] for Π-expert models:

$$\begin{aligned}\mathfrak{R}_T(\mathcal{F}_{MoE}) &\leq \sqrt{\frac{\log T}{T}} \sum_{\pi=1}^{\Pi} \mathbb{E}\left[\|\mathbf{w}_\pi\|_2\right] \\ &\leq \sqrt{\frac{\log T}{T}} \cdot \Pi\end{aligned}$$

Step 3: Link to Generalization Error. By Talagrand's contraction lemma [58]:

$$\begin{aligned}\mathbb{E}[\mathcal{E}_{MoE}] &\leq 2L\mathfrak{R}_T(\mathcal{F}_{MoE}) \\ &\leq 2L\sqrt{\frac{\log T}{T}} \\ &= C_1\sqrt{\frac{\log T}{T}}.\end{aligned}$$

□

Lemma A.2 (Gating Error Propagation)**.** *Under Definition 2 (Heterogenity Balance), the gating-induced error satisfies:*

$$\epsilon_{Gate} \leq L\sqrt{\mathbb{E}_v\left[(\alpha_v - \alpha_v^*)^2\right]} \leq C_2\epsilon_g.$$

Proof. The proof consists of three key phases:

Step 1: Error Vector Representation. Define the representation discrepancy between ideal and actual gating:

$$\Delta\mathbf{H}_v = (\alpha_v - \alpha_v^*)\mathbf{H}_{\text{low},v} + (\alpha_v^* - \alpha_v)\mathbf{H}_{\text{high},v}$$

Using the Lipschitz continuity of the loss function:

$$|\mathcal{L}(\mathbf{H}_v) - \mathcal{L}(\mathbf{H}_v^*)| \leq L\|\Delta\mathbf{H}_v\|_2$$

Step 2: Norm Analysis. By the filter energy bound $\|\mathbf{H}_{\text{low},v}\|_2, \|\mathbf{H}_{\text{high},v}\|_2 \leq 1$ (normalized representations):

$$\|\Delta\mathbf{H}_v\|_2 \leq |\alpha_v - \alpha_v^*|(\|\mathbf{H}_{\text{low},v}\|_2 + \|\mathbf{H}_{\text{high},v}\|_2) \leq 2|\alpha_v - \alpha_v^*|$$

Taking expectation over nodes:

$$\mathbb{E}_v\|\Delta\mathbf{H}_v\|_2 \leq 2\mathbb{E}_v|\alpha_v - \alpha_v^*| \leq 2\epsilon_g$$

Step 3: Concentration Inequality. Applying Cauchy-Schwarz inequality to the loss difference:

$$\epsilon_{\text{Gate}} = \mathbb{E}\left[|\mathcal{L}(\mathbf{H}_v) - \mathcal{L}(\mathbf{H}_v^*)|\right] \leq L\mathbb{E}\|\Delta\mathbf{H}_v\|_2 \leq 2L\epsilon_g = C_2\epsilon_g$$

□

Lemma A.3 (Distribution Calibration Error)**.** *Let $\hat{P}_{\mathcal{S}}$ be the calibrated support distribution using KDE with bandwidth σ, and $P_{\mathcal{Q}}$ be the query distribution. Under Definition 3 ($W_1(P_{\mathcal{S}}, P_{\mathcal{Q}}) \leq \delta$), the distribution shift error satisfies:*

$$\epsilon_{Dist} \leq C_3 \left(\delta + \mathcal{O}(\sigma^2) + \mathcal{O}\left(|\mathcal{Q}|^{-1/2} \right) \right)$$

where $C_3 = L \cdot \operatorname{diam}(\mathcal{Y})$ depends on the label space diameter.

Proof. We analyze the distribution calibration error via three steps:

Step 1: Error Decomposition. Using the triangle inequality of Wasserstein distance:

$$W_1(\hat{P}_{\mathcal{S}}, P_{\mathcal{Q}}) \leq \underbrace{W_1(P_{\mathcal{S}}, P_{\mathcal{Q}})}_{\text{Original shift}} + \underbrace{W_1(\hat{P}_{\mathcal{S}}, P_{\mathcal{S}})}_{\text{KDE estimation error}}$$

By definition 3, the first term is bounded by δ.

Step 2: KDE Estimation Error. Let $\hat{P}_{\mathcal{S}}(y) = \frac{1}{|\mathcal{Q}|} \sum_{x_j \in \mathcal{Q}} \mathcal{K}_\sigma(y - \tilde{y}_j)$ be the KDE-calibrated distribution, where $\tilde{y}_j$ are perturbed prototypes. Using the Kantorovich-Rubinstein duality [59]:

$$W_1(\hat{P}_{\mathcal{S}}, P_{\mathcal{S}}) = \sup_{\|f\|_L \leq 1} \left| \mathbb{E}_{y \sim \hat{P}_{\mathcal{S}}}[f(y)] - \mathbb{E}_{y \sim P_{\mathcal{S}}}[f(y)] \right|$$

where f is 1-Lipschitz. This can be bounded by:

$$W_1(\hat{P}_{\mathcal{S}}, P_{\mathcal{S}}) \leq \underbrace{\mathbb{E}[|\hat{P}_{\mathcal{S}}(y) - P_{\mathcal{S}}(y)|]}_{\text{Bias}} + \underbrace{\sqrt{\operatorname{Var}(\hat{P}_{\mathcal{S}}(y))}}_{\text{Variance}}$$

Step 3: Bias-Variance Analysis. For Gaussian kernel $\mathcal{K}_\sigma$ with bandwidth σ:

- **Bias term**: By Taylor expansion,

$$\mathbb{E}[\hat{P}_{\mathcal{S}}(y) - P_{\mathcal{S}}(y)] = \mathcal{O}(\sigma^2)$$

- **Variance term**: By the central limit theorem,

$$\operatorname{Var}(\hat{P}_{\mathcal{S}}(y)) = \mathcal{O}\left(\frac{1}{|\mathcal{Q}|\sigma^d} \right)$$

  where d is the feature dimension. Choosing $\sigma \sim |\mathcal{Q}|^{-1/(d+4)}$ optimizes the trade-off:

$$\sqrt{\operatorname{Var}(\hat{P}_{\mathcal{S}}(y))} = \mathcal{O}\left(|\mathcal{Q}|^{-1/2} \right)$$

Step 4: Final Bound. Combining all terms with the Lipschitz loss:

$$\mathcal{E}_{Dist} \leq L \cdot \operatorname{diam}(\mathcal{Y}) \cdot W_1(\hat{P}_{\mathcal{S}}, P_{\mathcal{Q}}) \leq C_3 \left(\delta + \mathcal{O}(\sigma^2) + \mathcal{O}\left(|\mathcal{Q}|^{-1/2} \right) \right)$$

where $\operatorname{diam}(\mathcal{Y}) = \sup_{y, y' \in \mathcal{Y}} \|y - y'\|_2$. □

Next, we formally prove the Theorem 5.1.

Proof. Recall Eq.11 and Lemmas A.1, A.2, A.3, the following inequality holds:

$$\begin{aligned} \epsilon_{gen} = & \underbrace{\epsilon_{MoE}}_{\text{Expert Variance}} + \underbrace{\epsilon_{Gate}}_{\text{Gating Error}} + \underbrace{\epsilon_{Dist}}_{\text{Distribution Shift}} \\ & \leq C_1 \sqrt{\frac{\log T}{T}} + C_2 \epsilon_g + C_3 \left(\delta + \mathcal{O}(\sigma^2) + \mathcal{O}(|\mathcal{Q}|^{-1/2}) \right) \end{aligned} \tag{12}$$

□

Thus, we complete the proof of Theorem 5.1.

A.3.2 Proof of Corollary 5.2

Proof. We compare the generalization error of the proposed model (ϵ_{gen}^{MoE}) with a baseline using a single graph filter (ϵ_{gen}^{Sin}).

Step 1: Baseline Error Characterization. For the single-filter baseline, Theorem 5.1 implies:

$$\epsilon_{gen}^{Sin} \leq C_1\sqrt{\frac{\log T}{T}} + C_2\epsilon_g^{Sin}.$$

Step 2: Error Difference Analysis. Subtract the proposed model's bound (Theorem 5.1) from the baseline:

$$\begin{aligned}\Delta\epsilon &= \epsilon_{gen}^{MoE} - \epsilon_{gen}^{Sin} \\ &\leq C_2\left(\epsilon_g - \epsilon_g^{Sin}\right) + \mathcal{O}\left(|\mathcal{Q}|^{-1/2}\right).\end{aligned}$$

Step 3: Gating Advantage. Under the strong heterogeneity ($\epsilon_g \to 0$), and noting $\epsilon_g^{Sin} \geq \epsilon_g$:

$$C_2(\epsilon_g - \epsilon_g^{Sin}) \leq L(\epsilon_g - \epsilon_g) = 0.$$

Step 4: Final Inequality. Combining these results:

$$\Delta\epsilon \leq \mathcal{O}\left(|\mathcal{Q}|^{-1/2}\right).$$

Thus, we complete the proof of Corallary 5.2. □

A.4 Dataset Descriptions

We conduct experiments on a variety of graph datasets from different domains. Each dataset is divided into disjoint class sets for meta-training, meta-validation, and meta-testing. The details are as follows:

Cora [23]: A citation graph where nodes represent academic papers and edges indicate citation relationships. Each node is assigned a label based on the paper's research topic. We divide the dataset into 3, 2, and 2 classes for meta-training, meta-validation, and meta-testing, respectively.

CiteSeer [23]: A document-level citation network consisting of scientific publications as nodes and citation links as edges. Labels reflect the thematic area of each document. The dataset is split into 2 classes for each of the three meta-learning phases.

Amazon-Computer [47]: A co-purchase network constructed from Amazon product data. Nodes denote products, and edges connect items frequently purchased together. Each product is categorized based on its functional type. We apply a 4/3/3 class split for training, validation, and testing.

Coauthor-CS [47]: A collaboration graph in which nodes correspond to authors and edges indicate co-authored publications within the computer science domain. Labels are derived from research specialties. A 5-class split is used for each meta stage.

DBLP [48]: A bibliographic co-authorship network where each node denotes a researcher and edges indicate joint publications. Node labels reflect academic fields. We partition the dataset into 77, 30, and 30 classes for training, validation, and testing.

CoraFull [49]: An extended version of the Cora dataset that includes a broader range of categories. Nodes represent papers, and citation links define the graph structure. We use 40 classes for meta-training, 15 for validation, and 15 for testing.

ogbn-arxiv [50]: A large-scale graph built from arXiv submissions in computer science. Each node corresponds to a paper, and edges are formed based on citation patterns. Labels are based on subject areas defined in the arXiv taxonomy. The dataset is split into 20 classes for training and 10 classes each for validation and testing.

A.5 Baseline Descriptions

A.5.1 Graph Embedding Methods

DeepWalk [51]: It leverages random walks inspired by the word2vec algorithm to generate low-dimensional node embeddings for graphs.
GCN [3]: It employs a first-order Chebyshev approximation graph filter to derive hidden node embeddings, which are then utilized for downstream task analysis.
SGC [45]: It streamlines the GCN architecture by eliminating non-linear activations and collapsing weight matrices, resulting in a simpler yet efficient model.

A.5.2 Meta-Learning Methods

ProtoNet [34]: It learns a metric space and predicts query sample categories by measuring their similarity to class prototypes derived from support samples.
MAML [33]: By optimizing model parameters through one or few gradient updates, it enables fast adaptation to new tasks with limited labeled data, providing a well-initialized meta-learner.

A.5.3 Graph Meta-Learning Methods

GPN [17]: It adapts ProtoNet by integrating a graph encoder and evaluator to learn node embeddings, assess node importance, and classify new samples based on their proximity to the nearest class prototype.
G-Meta [18]: By constructing node-specific subgraphs, it propagates localized node information and employs meta-gradients to extract transferable knowledge across tasks.
TENT [16]: It introduces an adaptive framework with node-level, class-level, and task-level components to bridge the generalization gap between meta-training and meta-testing, while minimizing performance fluctuations caused by task variations.
Meta-GPS [15]: Enhancing MAML, it incorporates prototype-based parameter initialization, scaling, and shifting transformations to improve meta-knowledge transfer and enable faster adaptation to new tasks.
TEG [53]: It designs a task-equivariant graph framework using equivariant neural networks to learn task-adaptive strategies, effectively capturing inductive biases from diverse tasks.
COSMIC [54]: It proposes a contrastive meta-learning framework that aligns node embeddings within each episode through a two-step optimization process for improved few-shot learning.

Meta-BP [55]: It proposes a lightweight graph meta-learner that extracts relevant knowledge from a black-box pre-trained GNN and leverages task-relevant information to quickly adapt to new tasks, while pruning the meta-learner to enhance its generalization ability on unseen tasks.

A.6 More Ablation Study

We conduct extensive ablation studies to examine the contribution of individual components in our proposed framework. By systematically removing or altering specific modules, we aim to assess their impact on overall performance and provide insights into the design choices. The detailed results are summarized in Tables 6, 7, 8, and 9. Specifically, Table 9 presents an additional ablation on the gating inputs defined in Eq. 4, where the input vector is constructed as $\mathbf{X}_g = \mathbf{X}\|\mathbf{N}\|\phi\|\mathbf{D}$, with $\mathbf{X}$ denoting the original node feature, $\mathbf{N} = |\hat{\mathbf{A}}\mathbf{X} - \mathbf{X}|$ the one-hop neighborhood difference, ϕ the feature-wise standard deviation, and $\mathbf{D}$ the node degree. We design four variants accordingly: (I) *w/o* $\mathbf{X}$: We remove the original feature; (II) *w/o* $\mathbf{N}$: We discard the neighborhood difference; (III) *w/o* ϕ: We eliminate the standard deviation; (IV) *w/o* $\mathbf{D}$: We exclude the degree information.

The ablation results clearly demonstrate that each designed module contributes significantly to the overall performance, which is consistent with our analysis in the ablation study section of the main text.

A.7 Limitation

Although our model achieves outstanding performance in graph few-shot learning, it currently considers only high-pass and low-pass filters. Incorporating a broader range of spectrum experts could potentially further enhance the model's performance. Moreover, it introduces several critical

Table 6: Results of different model variants on three datasets.

Model	Cora		CiteSeer		Amazon-Computer	
	2 way 3 shot	2 way 5 shot	2 way 3 shot	2 way 5 shot	2 way 3 shot	2 way 5 shot
w/o high	74.82±2.49	82.89±2.04	73.76±2.43	77.62±2.03	88.34±1.30	90.57±1.13
w/o low	78.91±2.11	83.37±1.93	67.46±2.39	70.62±2.23	92.06±0.60	94.31±5.54
w/o cal	82.35±2.04	85.35±1.77	71.17±2.44	79.32±1.69	92.32±0.55	94.62±0.52
w/o both	74.05±1.96	76.39±2.33	64.22±2.92	65.59±2.49	72.19±2.30	76.19±2.21
Ours	**82.40±2.03**	**86.19±1.80**	**75.67±2.44**	**79.64±1.79**	**92.46±0.55**	**94.66±0.50**

Table 7: Results of different model variants on two datasets.

Model	Coauthor-CS			DBLP		
	2 way 3 shot	2 way 5 shot	5 way 5 shot	5 way 3 shot	10 way 3 shot	10 way 5 shot
w/o high	93.46±1.41	93.00±1.40	80.82±1.19	76.85±2.11	66.81±1.63	70.08±1.59
w/o low	94.60±1.34	96.18±0.96	85.31±1.03	79.75±2.03	72.50±1.49	76.65±1.42
w/o cal	94.98±1.38	95.36±1.21	86.27±0.95	80.14±2.08	73.75±1.55	75.90±1.49
w/o both	85.60±2.15	88.70±2.21	81.79±3.18	75.39±3.41	67.20±2.40	71.12±1.87
Ours	**95.50±1.30**	**96.20±0.97**	**86.82±1.01**	**81.72±2.05**	**74.22±1.56**	**76.70±1.46**

Table 8: Results of different model variants on two datasets.

Model	CoraFull			ogbn-arxiv		
	5 way 5 shot	10 way 3 shot	10 way 5 shot	5 way 5 shot	10 way 3 shot	10 way 5 shot
w/o high	79.07±1.35	67.99±1.13	70.55±1.00	65.64±1.78	47.29±1.01	51.55±0.92
w/o low	81.32±1.23	67.57±1.15	73.95±0.97	67.01±1.62	49.98±1.04	54.94±0.91
w/o cal	80.38±1.30	70.09±1.11	74.20±0.93	67.58±1.75	48.46±1.06	53.77±0.95
w/o both	60.31±2.19	50.93±2.30	56.21±2.09	50.50±2.13	37.36±1.99	42.16±2.19
Ours	**81.60±1.28**	**70.91±1.08**	**74.54±0.98**	**68.34±1.73**	**50.18±1.01**	**55.07±0.91**

Table 9: Results of different model variants on seven datasets.

Model	Cora	CiteSeer	Amazon-Computer	Coauthor-CS	DBLP	CoraFull	ogbn-arxiv
	2 way 1 shot	2 way 1 shot	2 way 1 shot	5 way 3 shot	5 way 5 shot	5 way 5 shot	5 way 3 shot
w/o $\mathbf{X}$	63.46±2.75	63.83±2.74	89.95±0.80	85.36±1.15	84.81±1.91	80.91±1.30	62.46±1.86
w/o $\mathbf{N}$	63.12±2.93	60.99±2.91	89.53±0.84	84.63±1.19	84.55±1.84	80.80±1.27	62.06±1.93
w/o ϕ	61.46±2.74	63.68±3.07	89.47±1.01	85.39±1.10	84.83±1.90	81.01±1.20	62.19±1.92
w/o $\mathbf{D}$	63.80±2.84	62.27±2.93	89.41±1.03	85.26±1.09	85.01±1.83	81.52±1.28	62.26±1.93
Ours	**66.48±2.88**	**63.90±2.84**	**90.23±0.90**	**86.03±1.05**	**85.30±1.90**	**81.60±1.28**	**62.31±1.94**

hyperparameters that influence the final performance. Determining the optimal settings for these enhancements remains a challenging task. This indicates that there is still room for improvement in our model's performance due to the impact of hyperparameters.

A.8 Broader Impacts

This study aims to develop an effective approach for graph few-shot learning. Our proposed method not only advances the development of graph-based few-shot learning but may also offer insights for few-shot learning in other domains. While our work does not involve any ethical concerns, it may carry potential societal implications. However, we believe it is not necessary to emphasize them here.