Topology-Aware Feature Sorting Enables Universal Modeling on Homophilic and Heterophilic Graphs

Yi Wang[†]
wangyi076@tju.edu.cn
Tianjin University
Tianjin, China

Jitao Zhao[†]
zjtao@tju.edu.cn
Tianjin University
Tianjin, China

Dongxiao He[*]
hedongxiao@tju.edu.cn
Tianjin University
Tianjin, China

Jia Li
jialee@hkust-gz.edu.cn
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China

Yuxiao Huang
yuxiaohuang@gwu.edu
George Washington University
Washington, United States

Zhiyong Feng
zyfeng@tju.edu.cn
Tianjin University
Tianjin, China

Abstract

Recently, Graph Foundation Models (GFMs) have rapidly emerged as a central focus in graph learning research due to their strong universality to generalize across various unseen graphs. However, existing GFMs typically assume that graphs follow the homophily assumption, and the exploration of universality on heterophilic graphs is still in its early stages. Moreover, these GFMs struggle to effectively handle the few but important heterophilic information present in homophilic graphs. Due to the need for universality, the heterophily issue faced by GFMs is more challenging than in classical graph learning, as it requires training only one model to adapt to different topologies, features and tasks. Classic heterophilic graph learning methods primarily based on the node-level homophily or heterophily. However, we highlight that homophily and heterophily exist not only at the node semantic level, but also at a finer granularity across individual features dimensions. This finding enables GFMs to adapt to heterophilic graphs and better utilize the small amount of heterophilic information in homophilic graphs. Based on this, we propose Topology-aware Feature Sorting Graph Foundation Model (TFSGFM), which employs a feature-level topology-aware sorting strategy and a dual-channel GNN framework, addressing the universality issues of both features and topology. Extensive experiments demonstrate the strong generalizability of TFSGFM.

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Neural Networks, Graph Self-Supervised Learning, Graph Pre-training, Graph Foundation Models

ACM Reference Format:
Yi Wang, Jitao Zhao, Dongxiao He, Jia Li, Yuxiao Huang, and Zhiyong Feng. 2018. Topology-Aware Feature Sorting Enables Universal Modeling on Homophilic and Heterophilic Graphs. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Graph learning has emerged as a powerful paradigm for modeling complex and diverse relationships among entities, with applications across a wide range of domains such as social networks [5], recommender systems [9, 36], fraud detection [3, 16], and knowledge graphs [4, 18]. Recently, Graph Foundation Models (GFMs) have rapidly become a central focus in graph learning research, attracting increasing attention from scholars [26, 34, 45]. Specifically, GFMs aim to address the challenges posed by diverse graph topologies and substantial distribution shifts across domains [35], both of which hinder effective cross-domain knowledge transfer.

Many recent studies have attempted to explore GFMs from different perspectives. Existing works can be broadly categorized into LLM-based methods [17] and non-LLM-based methods [33]. LLM-based methods aim to integrate the knowledge and powerful reasoning generalization capabilities of LLM with graph models, thereby enhancing generalization on unseen graphs, particularly text-attribute graphs. Non-LLM-based methods focus on developing universal and powerful graph encoders and downstream fine-tuning algorithms to improve the learning ability and few-shot reasoning ability of graph models for arbitrary graph data.

However, most GFMs focus on homophilic graphs, with limited exploration of heterophilic graphs. While a few recent studies [33, 44] have attempted to address this challenge by leveraging Mixture-of-Experts (MoE) architectures or Graph Structure Learning (GSL) frameworks to adaptively optimize information propagation, these approaches generally rely on determining the homophily or heterophily of each link, which is highly challenging in practice as node semantics are complex and labels are scarce. Outside of

[†]Both authors contributed equally to this research.
[*]Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

GFMs, existing successful heterophilic graph representation learning methods struggle to generalize across diverse graphs.

Therefore, developing a graph foundation model capable of effectively handling heterophilic graphs faces two major challenges: (1) In both the self-supervised pre-training phase and the downstream label-scarce scenarios, it is extremely difficult to recognize the homophily or heterophily of each node. (2) The patterns of homophily and heterophily in graph data are complex. Homophilic graphs often contain some heterophily edges, while heterophilic graphs also include some homophilic edges, making it difficult to develop a universal modeling approach.

To address these challenges, we conduct an in-depth exploration of the relationship between data features and topology. Through a motivation experiment, we highlight that **homophily and heterophily exist not only at the node semantic level, but also at a finer granularity across individual features dimensions**. This finding enables GFMs to adapt to heterophilic graphs and better utilize the small amount of heterophilic information in homophilic graphs. Moreover, real-world graphs are rarely purely homophilic or heterophilic. Even in heterophilic graphs, some feature may exhibit homophilic tendencies, and vice versa. Fully identifying and utilizing both types of structural information can significantly enhance the generalization capability of model. Given that defining homophily or heterophily under label-scarce conditions is unreliable, we aim to mine such information solely from the graph structure and node features, without relying on labels. Additionally, to ensure the universality of the model, it is necessary to achieve semantic alignment of features between different datasets.

Based on the above views, we propose Topology-aware Feature Sorting Graph Foundation Model (TFSGFM), a method with the capability of co-unifying structure and features. Its core lies in a topology-aware feature sorting mechanism: by evaluating the homophily of each feature dimension based on graph structure, the model sorts and splits features accordingly, dividing the graph into a homophilic subgraph and a heterophilic subgraph. Then, a dual-channel GNN encoder is employed to model the two subgraphs separately, thereby addressing the universality of both features and topology. This approach not only achieves co-unification of structure and features, but also fully leverages both homophilic and heterophilic information, significantly enhancing the model's cross-domain generalization capability.

Our main contributions are summarized as follows:

- We reveal that homophily and heterophily exist not only at the node semantic level, but also at a finer granularity across individual feature dimensions. Moreover, we believe that most real-world graphs contain both homophilic and heterophilic components to varying degrees. Effectively identifying, separating, and leveraging them is key to improving the generalization ability of graph models.
- We propose TFSGFM, a graph foundation model framework applicable to both homophilic and heterophilic graphs. It leverages topology-aware feature sorting to split graphs into homophilic and heterophilic subgraphs, achieving co-unification of structure and features.
- We conduct extensive experiments on various tasks involving both homophilic and heterophilic datasets. TFSGFM performs well not only on heterophilic datasets, but also achieves strong performance on homophilic datasets by effectively incorporating heterophilic information.

2 Related Work

Graph Pre-training. Graph pre-training methods aim to utilize the structure and feature information inherent in graph as supervision signals to train models in a self-supervised manner, which are then adapted to downstream tasks via fine-tuning [39] or prompt learning [6, 19, 27]. Mainstream graph pre-training methods can be categorized into contrastive learning methods and generative methods. Contrastive learning methods construct positive and negative sample pairs to pull representations of similar nodes closer while pushing those of dissimilar nodes apart, thereby enhancing the discriminability of node representations. For example, GRACE [47] perturbs graph data to generate different views and obtain positive and negative samples for training. SimGRACE [37] perturbs the graph encoder itself to generate contrastive views. Generative methods achieve self-supervised learning by designing graph structure reconstruction tasks. Typical methods such as GAE [14] use an encoder-decoder structure, where the encoder generates node representations and the decoder reconstructs the graph structure. However, existing methods typically assume that the pre-trained graph and the target graph come from the same or similar domains, lacking the ability to transfer across domains.

Graph Foundation Models. Graph foundation models aim to learn universal graph knowledge through pre-training on multi-domain graph data, enabling rapid adaptation and generalization to unseen graph domains [40]. In recent years, related studies have mainly focused on two directions: feature unification and structure unification. In terms of feature unification, methods such as One for All [17] leverage large language models (LLMs) to encode textual descriptions of nodes. By mapping node features into a semantically aligned space, this approach achieves effective cross-domain feature alignment for graphs with textual attributes. However, its applicability is limited when dealing with graphs that contain only numerical features. Another representative work, MDGPT [41] introduces domain tokens to project graph features from different domains into a unified semantic space. However, due to the large discrepancies in features across graph domains, constructing such a unified semantic space is extremely challenging. In terms of structure unification, SAMGPT [40] aligns structures across different graphs by regulating the node aggregation process. However, this method is primarily applicable to structure transfer among homophilic graphs and struggles to generalize effectively to heterophilic graphs. MDGFM [33] attempts to unify graph structures by balancing the contributions of structure and features, combined with Graph Structure Learning (GSL) to reconstruct a unified structure, thereby achieving structural unification. However, such methods tend to introduce structural bias, which can result in inconsistent or suboptimal performance when applied to different types of graphs. Therefore, current graph foundation models face significant challenges in both feature and structure unification, urgently requiring a model that can achieve co-unification of structure and features, while being applicable to both homophilic and heterophilic graphs.

3 Preliminaries

Multi-Domain Pre-training. In the pre-training stage of Graph Foundation Models, we assume access to datasets from multiple domains, containing T training graphs. Let the pre-training datasets be denoted as $P = \{G^{(1)}, G^{(2)}, \ldots, G^{(T)}\}$, where each graph $G^{(i)} = (V^{(i)}, \mathcal{E}^{(i)}, A^{(i)}, X^{(i)})$ is composed of a node set $V^{(i)}$, an edge set $\mathcal{E}^{(i)}$, an adjacency matrix $A^{(i)}$, and a node feature matrix $X^{(i)} \in \mathbb{R}^{|V^{(i)}| \times d^{(i)}}$. Here, $|V^{(i)}|$ denotes the number of nodes and $d^{(i)}$ denotes the feature dimension of the graph $G^{(i)}$. Each graph $G^{(i)}$ belongs to a domain $D^{(i)}$. Note that multiple graphs may belong to the same domain, i.e., $G^{(i)} \neq G^{(j)}$ but $D^{(i)} = D^{(j)}$.

Downstream Target-Domain Adaptation. In the downstream phase, we adapt the pre-trained model to an unseen target graph, denoted as $G^{(\text{target})} = (V^{(\text{target})}, \mathcal{E}^{(\text{target})}, A^{(\text{target})}, X^{(\text{target})})$, where $G^{(\text{target})} \notin P$. The corresponding domain of this graph is represented by $D^{(\text{target})}$. The node set $V^{(\text{target})}$ is divided into two subsets: Support set $S = \{v_1, v_2, \ldots, v_l\}$, where each node $v_i \in S$ has a known label $y_i \in \{1, 2, \ldots, C\}$, and C is the total number of classes; Query set $Q = \{v_{l+1}, \ldots, v_n\}$, where $n = |V^{(\text{target})}|$ denotes the number of nodes. The goal of the task is to accurately predict the labels of all nodes in the query set Q.

Feature Homophily Score. For a graph $G = (V, \mathcal{E}, A, X)$, the feature homophily score quantifies the similarity of connected nodes across each feature dimension. Formally, the homophily score for the j-th feature dimension is defined as:

$$\text{Score}_j = -\frac{1}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \left(X_{u,j} - X_{v,j}\right)^2, \tag{1}$$

where $|\mathcal{E}|$ denotes the total number of edges and $X_{u,j}$ denotes the value of the j-th feature dimension of node u.

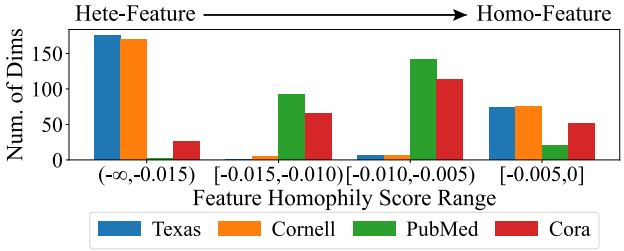

Figure 1: Feature homophily score distribution across different datasets. Prior to computing the feature homophily score, all datasets are processed through PCA-based dimension alignment to 256 dimensions.

4 Method

4.1 Motivation

In label-scarce scenarios, only node features and graph topology can be gained. Existing methods often infer node homophily or heterophily based on node similarity, which may somewhat arbitrary. To more accurately identify homophily and heterophily in graph data, we shift our focus from the node semantic level to the single feature dimension level. Specifically, we calculate the feature homophily score for each feature dimension after the dimension alignment process on two typical homophilic graphs (Cora [21] and PubMed [11]) and two typical heterophilic graphs (Cornell [22] and Texas [22]), with the results illustrated in Figure 1.

The experimental results show that the distributions of feature homophily scores vary significantly across different datasets. This indicates substantial discrepancies in features across domains, making it extremely difficult to construct a unified semantic space. Motivated by this, instead of aligning absolute semantics, we sort features according to their homophily scores and focus on capturing the relative semantics between different feature dimensions.

Furthermore, we observe that homophilic graphs also contain some heterophilic information, while heterophilic graphs also contain some homophilic information. This implies that simply unifying all structures into a single form will inherently introduce bias: either more suitable for homophilic graphs or more suitable for heterophilic graphs. Inspired by this, we propose to split the homophilic and heterophilic information in graph data and jointly leverage both to improve model performance.

To this end, we propose TFSGFM, which captures the relative semantics between feature dimensions and effectively leverages both homophilic and heterophilic information.

4.2 Overview

As illustrated in Figure 2, our model consists of two key modules: a feature unification module and a structure unification module, with the goal of achieving co-unification of structure and features across domains. In the feature unification module, we first unify the feature dimension for each training graph. Then, based on the graph structure, we compute the feature homophily score for each dimension and sort the features according to their scores. This process enables relative semantic alignment across datasets. In the structure unification module, the sorted features are split based on a predefined ratio to generate two subgraphs: a homophilic subgraph and a heterophilic subgraph. For the homophilic subgraph, we retain the original graph structure. For the heterophilic subgraph, due to its low structural homophily, we employ Graph Structure Learning (GSL) [12] to reconstruct the adjacency matrix before applying GNN encoding. This design facilitates the co-unification of structure and features across different domains.

4.3 Feature Unification

In multi-domain graph pre-training, graphs from different domains often exhibit significant differences not only in feature dimension, but more critically in feature semantic. Therefore, unifying both feature dimension and semantic is the foundation for multi-domain pre-training and achieving cross-domain transferability.

Firstly, we apply the dimension alignment technique to unify the node features of graph data from different domains to the same dimension, thus achieving the unification of feature dimension:

$$H^{(i)} = \text{Align}(X^{(i)}) \in \mathbb{R}^{|V^{(i)}| \times d}, \tag{2}$$

where $H^{(i)}$ represents the node features after dimension alignment, d represents the unified feature dimension, and $\text{Align}(\cdot)$ represents the dimension alignment operation, which can be implemented using MLP, Singular Value Decomposition (SVD) [11], or Principal

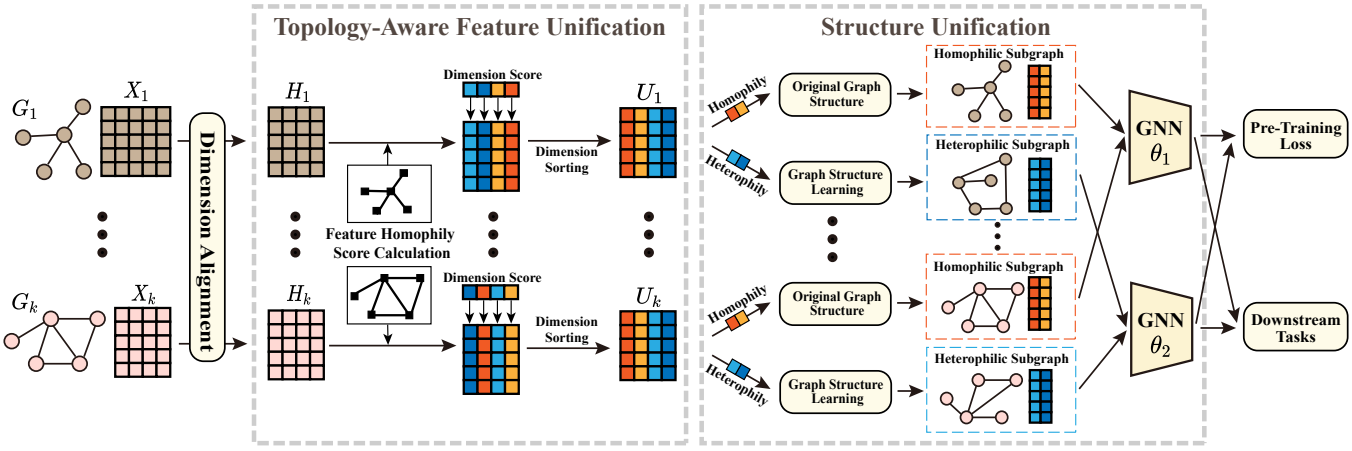

Figure 2: The overview of TFSGFM.

Component Analysis (PCA) [1]. Here, we adopt PCA to perform the dimension alignment operation. It is important to note that PCA can only achieve the unification of feature dimension but does not guarantee semantic consistency of the features. Therefore, semantic alignment operations are also required.

To achieve semantic alignment of features between different datasets, we propose the concept of feature homophily score. In graph, some feature dimensions may exhibit strong similarity across adjacent nodes (i.e., homophilic), while others may show weak or even opposite correlations (i.e., heterophilic). We argue that if two feature dimensions have similar homophily scores, they may have similar effects. This allows us to establish a consistent feature ordering across graphs, thereby achieving semantic alignment.

Specifically, we compute the feature homophily score for each feature dimension based on the structural of the graph, resulting in $\text{Score}_1^{(i)}, \text{Score}_2^{(i)}, \ldots, \text{Score}_d^{(i)}$:

$$\text{Score}_j^{(i)} = -\frac{1}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \left(H_{u,j}^{(i)} - H_{v,j}^{(i)}\right)^2. \tag{3}$$

Next, we sort all feature dimensions in descending order based on their homophily scores:

$$U^{(i)} = \text{SortFeatures}(H^{(i)}, \text{Score}^{(i)}), \tag{4}$$

where $\text{SortFeatures}(H^{(i)}, \text{Score}^{(i)})$ denotes the operation that reorders the columns of the feature matrix $H^{(i)}$ in descending order according to the feature homophily scores $\text{Score}^{(i)}$. The resulting $U^{(i)}$ represents the node features after feature sorting.

By performing both dimension alignment and feature sorting, we achieve the unification of feature dimension and semantic. This not only enhances the consistency and transferability of features across diverse graph domains but also establishes a solid and reliable foundation for the subsequent structure unification phase.

4.4 Structure Unification

Previous research on Graph Foundation Models primarily focused on unifying node features across different datasets, often neglecting the importance of unifying structure [41, 43], resulting in suboptimal performance in downstream tasks. Previous studies have shown that the excellent performance of GNN is closely related to its message-passing mechanisms, which is highly dependent on the graph structure [13]. In fact, the structural differences across datasets may be more pronounced than feature-level differences. Therefore, if the same GNN is used to encode different datasets without unifying the structure, even if the features are aligned perfectly, it will be difficult to achieve optimal performance.

We argue that both homophilic and heterophilic patterns coexist in most graphs, and that both types of structural information are valuable for downstream tasks. If all graphs are simply unified into a single structural form, it inevitably introduces bias (favoring either homophilic or heterophilic structures) and results in inconsistent performance across different domains. To mitigate this, we propose a graph splitting strategy to achieve the unification of structure. The idea is to split the original graph into a homophilic subgraph and a heterophilic subgraph, thereby simultaneously capturing and utilizing both homophily and heterophily information within the graph. During feature unification phase, we have sorted the features based on the feature homophily scores, and the sorting results can reflect the homophilic level of different feature dimensions: higher-ranked dimensions correspond to stronger homophily, while lower-ranked dimensions indicate more pronounced heterophily. Based on this, we split the feature matrix $U^{(i)}$ into two subsets according to a predefined splitting ratio $\rho \in (0, 1)$. The first ρd dimensions are designated as the homophilic features $U^{\text{homo}(i)}$, while the remaining $(1-\rho)d$ dimensions are designated as the heterophilic features $U^{\text{hete}(i)}$:

$$U^{(i)} = [U^{\text{homo}(i)} || U^{\text{hete}(i)}]. \tag{5}$$

Consequently, we can construct two subgraphs by associating each feature subset with the original graph structure. These subgraphs are denoted as $G^{\text{homo}(i)} = (V^{(i)}, \mathcal{E}^{(i)}, A^{(i)}, U^{\text{homo}(i)})$ and $G^{\text{hete}(i)} = (V^{(i)}, \mathcal{E}^{(i)}, A^{(i)}, U^{\text{hete}(i)})$. In this way, the original graph is split into a homophilic subgraph and a heterophilic subgraph, achieving co-unification of structure and features.

At the encoding stage, we use a Dual-GNN architecture to encode both homophilic subgraph and heterophilic subgraph separately. For the homophilic subgraph $G^{\text{homo}(i)}$, we directly apply the GNN encoder to obtain the homophilic representations of the nodes:

$$Z^{\text{homo}(i)} = \text{GNN}_{\theta_1}(A^{(i)}, U^{\text{homo}(i)}), \tag{6}$$

where GNN_{θ_1} denotes the encoder for the homophilic subgraph with learnable parameters θ_1. For the heterophilic subgraph $G^{\text{hete}(i)}$, the original structure is unsuitable to directly apply the GNN encoder due to its low homophily. Therefore, we introduce the GSL approach to adaptively reconstruct its adjacency matrix. Specifically, we first compute a similarity matrix between nodes based on the heterophilic features $U^{\text{hete}(i)}$, then apply the k-Nearest Neighbors (kNN) [7] sparsification to obtain a new adjacency matrix $A^{\text{recon}(i)}$, thus obtaining the reconstructed heterophilic subgraph $G^{\text{recon}(i)} = (V^{(i)}, \mathcal{E}^{\text{recon}(i)}, A^{\text{recon}(i)}, U^{\text{hete}(i)})$. We then use another GNN to extract the heterophilic representations of the nodes:

$$Z^{\text{hete}(i)} = \text{GNN}_{\theta_2}(A^{\text{recon}(i)}, U^{\text{hete}(i)}), \tag{7}$$

where GNN_{θ_2} denotes the encoder specifically designed for the heterophilic subgraph, parameterized by θ_2.

By combining the graph splitting strategy with a Dual-GNN architecture, our model is able to fully leverage both homophilic and heterophilic structural information, thereby significantly enhancing the model's generalization ability in complex topological scenarios.

4.5 Training Loss

To effectively train our proposed TFSGFM, we employ the Scattering Loss, as proposed in SGRL [10]. This loss is designed to maximize the distance between each node and the global mean vector, while simultaneously minimizing the distance between neighboring nodes. Specifically, the loss is composed of two components:

$$\begin{aligned} \mathcal{L}_{\text{center}}(Z) &= -\frac{1}{n}\sum_{i=1}^{n}\|Z_i - \mu\|^2, \quad \mu = \frac{1}{n}\sum_{i=1}^{n} Z_i, \\ \mathcal{L}_{\text{neighbor}}(Z, A) &= \frac{1}{|\mathcal{E}|}\sum_{i,j} A_{i,j}\|Z_i - Z_j\|^2, \end{aligned} \tag{8}$$

where n denotes the total number of nodes, $Z \in \mathbb{R}^{n\times d}$ is the node representation matrix, $A \in \{0,1\}^{n\times n}$ is the adjacency matrix and $|\mathcal{E}| = \sum_{i,j} A[i,j]$ represents the total number of edges. We apply the above loss functions separately to both the homophilic subgraph $G^{\text{homo}(i)}$ and the reconstructed heterophilic subgraph $G^{\text{recon}(i)}$, defining the training loss as:

$$\begin{aligned} \mathcal{L}_{\text{train}}^{(i)} =& \mathcal{L}_{\text{center}}(Z^{\text{homo}(i)}) + \lambda \cdot \mathcal{L}_{\text{neighbor}}(Z^{\text{homo}(i)}, A^{(i)}) \\ &+ \mathcal{L}_{\text{center}}(Z^{\text{hete}(i)}) + \lambda \cdot \mathcal{L}_{\text{neighbor}}(Z^{\text{hete}(i)}, A^{\text{recon}(i)}), \end{aligned} \tag{9}$$

where λ is a hyper-parameter that balances the two loss components, and $A^{\text{recon}(i)}$ is the adjacency matrix reconstructed by GSL for the heterophilic subgraph.

4.6 Downstream Target-Domain Adaptation

During downstream adaptation, to better leverage both homophilic and heterophilic information in target graph, we concatenate the homophilic and heterophilic node representations to form the overall node representations:

$$Z^{(\text{target})} = \text{Concat}(Z^{\text{homo}(\text{target})}, Z^{\text{hete}(\text{target})}), \tag{10}$$

where $\text{Concat}(\cdot,\cdot)$ denotes the concatenation operation.

We believe that a strong graph foundation model should possess the ability to rapidly adapt to downstream target-domain. Therefore, we adopt a zero-tuning strategy (i.e., without applying any fine-tuning or prompt-based methods) for target-domain adaptation. Specifically, given a target graph $G^{(\text{target})}$ and a support set $S = \{v_1, v_2, ..., v_l\}$, we first compute the prototype vector p_c for each class c :

$$p_c = \frac{1}{|S_c|}\sum_{v_i \in S_c} Z_i^{(\text{target})}, \tag{11}$$

where $S_c \subset S$ is the subset of support nodes labeled with class c, and $Z_i^{(\text{target})}$ is the overall node representation of node v_i. For any query node $v_j \in Q$, its predicted label $\hat{y}_j$ is determined by computing the similarity between its overall representation $Z_j^{(\text{target})}$ and all class prototypes, and selecting the one with the highest similarity:

$$\hat{y}_j = \arg\max_{c\in\{1,\ldots,C\}} \text{Sim}(Z_j^{(\text{target})}, p_c), \tag{12}$$

where $\text{Sim}(\cdot,\cdot)$ denotes the similarity function, and in this work, we employ the cosine similarity.

By adopting this zero-tuning strategy, there is no need for any downstream training process. Therefore, the excellent performance of our model can be fully attributed to the effectiveness of the proposed graph foundation model architecture, rather than relying on domain-specific fine-tuning or prompt methods.

5 Experiments

In this section, we evaluate the performance of TFSGFM in the following three scenarios: (1) the mixed homophily-heterophily scenario involving both homophilic and heterophilic datasets; (2) the homophily-only scenario consisting solely of homophilic datasets; and (3) the graph classification scenario.

5.1 Datasets

In the mixed homophily-heterophily scenario, we conduct experiments on six datasets: Cora, CiteSeer, PubMed, Cornell, Chameleon, and Squirrel. In the homophily-only scenario, we use five datasets: Cora, CiteSeer, PubMed, Photo, and Computers. In the graph classification scenario, we evaluate on three datasets: IMDB-BINARY, ENZYMES, and DD. In the data ablation study, we additionally introduce Wisconsin and Texas as source domains. The statistics of all datasets used in our experiments are summarized in Tables 1 and 2, and detailed descriptions are provided in Appendix A.

5.2 Compared Methods

We compare TFSGFM with the four categories of baselines: (1) Supervised Graph Neural Networks: we select GCN [15] and GAT [29]. (2) Graph Self-Supervised Learning Methods: we select DGI [30] and GraphCL [39]. (3) Graph Prompting Methods: we select GPPT

Table 1: Node classification dataset statistics.

Dataset	Nodes	Edges	Feature Dims	Classes
Cora	2,708	10,556	1,433	7
CiteSeer	3,327	9,104	3,703	6
PubMed	19,717	88,648	500	3
Photo	7,650	238,162	745	8
Computers	13,752	491,722	767	10
Cornell	183	298	1,703	5
Chameleon	2,277	36,101	2,325	5
Squirrel	5,201	217,073	2,089	5
Wisconsin	251	512	1,703	5
Texas	183	325	1,703	5

[27], GraphPrompt [19], GPF [6], and GPF-plus [6]. (4) Universal Graph Pretraining Models: we select All-in-One [28], GCOPE [42], ULTRA [8], SCORE [31], MDGPT [41], and MDGFM [33].

5.3 Experimental Settings

To thoroughly evaluate the cross-domain transferability of TFSGFM, we follow the principle that target domain must remain unseen during pre-training in all experimental scenarios. Specifically, when a dataset is designated as the downstream target domain, the remaining datasets are used for multi-domain pre-training.

In the mixed homophily-heterophily scenario, we conduct one-shot (1-shot) node classification experiments, following the settings of MDGFM [33], where 50 few-shot tasks are generated for each dataset, and we report the average accuracy of these 50 trials. In the homophily-only scenario, we conduct 1-shot, 3-shot, and 5-shot node classification experiments, following the settings of MDGPT [41]. For each dataset, we generate 500 few-shot tasks and report the average accuracy of these 500 trials. In the graph classification scenario, we strictly follow the experimental settings of SCORE [31], conduct zero-shot graph classification experiments, and report classification accuracy and F1 score as the evaluation metric. More details of experimental settings can be found in the Appendix B.

5.4 Mixed Homophilic-Heterophilic Scenario

In this scenario, we conduct one-shot node classification experiments to evaluate TFSGFM's ability to adapt to both homophilic and heterophilic graphs. As shown in Table 3, TFSGFM achieves the best performance on five out of six datasets, significantly outperforming all baselines in datasets such as Cora, CiteSeer, PubMed, and Cornell, demonstrating its strong generalization ability and structural adaptability. Although TFSGFM does not achieve the best result on Squirrel, it is important to note that TFSGFM performs node classification under the zero-tuning setting, while MDGFM employs a complex prompt strategy in the downstream stage. Even under such constraints, TFSGFM still exhibits competitive performance. Notably, MDGFM shows excellent performance on heterophilic graphs but performs poorly on homophilic graphs. This is mainly because MDGFM attempts to unify homophilic and heterophilic structures by reconstructing a single unified graph structure. However, this approach inevitably introduces certain bias, making it difficult to exhibit optimal performance in both homophilic and heterophilic graphs. In contrast, TFSGFM unifies homophilic and heterophilic structures by explicitly splitting homophilic and heterophilic information in graph data, exhibiting outstanding performance on both homophilic and heterophilic graphs. In heterophilic graphs, MDGPT performs the worst among all GFMs, primarily because it overly relies on homophily assumptions and neglects heterophily.

5.5 Homophilic-Only Scenario

Considering that most real-world graphs are homophilic, testing TFSGFM's performance in homophilic graphs is particularly meaningful. Therefore, we additionally conduct node classification experiments in the homophily-only scenario. Furthermore, since homophilic datasets typically have larger scales and more accessible labeled samples, in addition to one-shot experiments, we also conduct 3-shot and 5-shot node classification tasks.

The one-shot results are presented in Table 4. The results indicate that TFSGFM exhibits exceptional performance the homophily-only scenario, achieving the best results across all five datasets, and significantly outperforming other baselines, especially on Cora, where it outperforms the best competitor by 10%. We attribute TFSGFM's outstanding performance to three factors: (1) Topology-aware feature sorting enables TFSGFM to capture relative semantics between different dimensional features, offering better transferability than MDGPT's absolute alignment. (2) TFSGFM achieve co-unification of structure and features across domains, while MDGPT only unifies features, ignoring the importance of graph structure. (3) TFSGFM fully leverages heterophilic information present within homophilic graphs, enhancing the distinguishability of node representations and significantly improving the model's expressive capacity.

Comparing the results in Tables 3 and 4, we observe a counterintuitive conclusion: compared to the homophily-only scenario, TFSGFM exhibits better performance on homophilic datasets (e.g., Cora, CiteSeer, PubMed) in the mixed homophily-heterophily scenario. We attribute this to the richer structural diversity in the mixed homophily-heterophily scenario, which enables the model to learn more generalizable parameters. Moreover, injecting appropriate heterophilic information during training may help the model better capture the heterophily within the homophilic graphs, enhancing the model's ability to model homophilic.

Experimental results of one-shot node classification have demonstrated that TFSGFM significantly outperforms the existing GFMs in cross-domain scenarios. Therefore, in the 3-shot and 5-shot experiments, we directly compare the cross-domain pretrained TFSGFM against in-domain pretrained models. Results are shown in Table 5. To further verify that TFSGFM's superior performance is attributed to its model design rather than to fine-tuning or prompt-based methods, we maintain the zero-tuning setting in these tasks. In contrast, for in-domain baselines, we report the best results from [48], where each prompt framework is paired with its optimal pretrained model. Despite operating under dual disadvantages (cross-domain setting and zero-tuning setting), TFSGFM still exhibits outstanding performance, significantly outperforming other methods on Cora and CiteSeer, and achieving competitive results on PubMed, verifying TFSGFM's cross-domain transferability and rapid adaptability.

Table 2: Graph classification dataset statistics.

Dataset	Graphs	Avg.nodes	Avg.edges	Feature Dims	Classes
IMDB-BINARY	1,000	19.8	212.8	136	2
ENZYMES	600	32.6	156.9	3	6
DD	1,178	284.3	1,715.6	89	2

Table 3: One-shot node classification performance in the mixed homophily-heterophily scenario. The best results are highlighted in bold. Methods with "*" are reported from [33].

Methods	Cora	CiteSeer	PubMed	Cornell	Squirrel	Chameleon
GCN*	28.57 ± 5.07	31.27 ± 4.53	40.55 ± 5.65	31.81 ± 4.71	20.00 ± 0.29	24.17 ± 5.21
GAT*	28.40 ± 6.25	30.76 ± 5.40	39.99 ± 4.96	28.03 ± 13.19	21.55 ± 2.30	23.93 ± 4.11
DGI*	29.30 ± 5.82	30.03 ± 4.88	41.85 ± 7.78	31.54 ± 15.66	21.15 ± 1.68	21.73 ± 5.47
GraphCL*	34.94 ± 6.49	30.58 ± 4.58	40.37 ± 7.81	27.15 ± 12.64	21.42 ± 2.23	22.49 ± 3.02
GPPT*	17.52 ± 5.52	21.45 ± 3.45	36.56 ± 5.31	25.09 ± 2.92	20.09 ± 0.91	24.53 ± 2.55
GPF*	37.84 ± 11.07	37.61 ± 8.87	46.36 ± 7.48	34.54 ± 7.73	21.92 ± 3.50	25.90 ± 8.51
GCOPE*	34.23 ± 8.16	39.05 ± 8.82	44.85 ± 6.72	34.02 ± 11.94	22.46 ± 1.96	24.61 ± 3.99
MDGPT*	39.54 ± 9.02	39.24 ± 8.95	45.39 ± 11.01	33.58 ± 10.38	22.35 ± 3.77	23.68 ± 1.56
MDGFM*	44.83 ± 7.41	42.18 ± 6.41	46.84 ± 7.31	40.77 ± 5.96	**24.30 ± 3.26**	28.36 ± 3.65
TFSGFM	**54.10 ± 8.40**	**48.57 ± 10.15**	**56.22 ± 8.47**	**46.08 ± 10.15**	22.40 ± 1.77	**28.45 ± 5.94**

Table 4: One-shot node classification performance in the homophily-only scenario. The best results are highlighted in bold. Methods with "*" are reported from [41].

Methods	Cora	CiteSeer	PubMed	Photo	Computers
GCN*	28.57 ± 5.07	26.82 ± 5.92	40.03 ± 8.53	46.37 ± 10.58	34.77 ± 11.76
GAT*	28.40 ± 6.25	23.79 ± 2.96	38.99 ± 4.95	29.42 ± 7.96	31.57 ± 5.87
DGI*	29.30 ± 5.82	30.03 ± 4.88	41.85 ± 7.78	46.18 ± 7.48	39.37 ± 8.07
GraphCL*	34.94 ± 6.49	30.58 ± 4.58	40.36 ± 7.81	42.26 ± 7.31	45.28 ± 6.59
GPPT*	15.37 ± 4.51	23.24 ± 2.94	36.56 ± 5.31	16.19 ± 4.73	19.22 ± 8.71
GraphPrompt*	35.90 ± 7.10	32.76 ± 7.66	43.34 ± 10.66	49.88 ± 8.31	43.03 ± 10.35
GPF*	37.84 ± 11.07	37.61 ± 8.87	46.36 ± 7.48	49.42 ± 7.04	37.00 ± 6.52
GCOPE*	33.38 ± 6.86	35.56 ± 6.81	42.10 ± 8.07	48.52 ± 7.78	40.22 ± 7.82
MDGPT*	42.26 ± 10.18	42.40 ± 9.26	49.82 ± 8.38	64.82 ± 10.53	49.77 ± 11.00
TFSGFM	**52.34 ± 8.16**	**46.66 ± 9.16**	**53.93 ± 11.08**	**68.44 ± 9.76**	**55.51 ± 11.89**

5.6 Graph Classification Scenario

To comprehensively evaluate the generalization ability of TFSGFM, we also conduct zero-shot graph classification experiments, and the results are shown in Table 6. On IMDB-BINARY and DD, TFSGFM achieves the best performance, while on ENZYMES, it also demonstrates competitive performance, further verifying the strong generalization capability of TFSGFM. It is worth noting that in graph classification tasks, the scale of graph data is generally relatively small. This also supports our viewpoint that every graph contains both heterophily and heterophily, and even in small-scale graph data, utilizing both homophilic and heterophilic information can help improve the performance of the model.

5.7 Data Ablation Study

To investigate the impact of the number of source domains on model performance, we vary the number of source domains from 1 to 3 and evaluate TFSGFM on one-shot node classification in the target domain. The results are presented in Table 7. From the results, we observe that TFSGFM consistently outperforms all baseline methods under all settings, demonstrating its strong effectiveness.

Table 5: Few-shot node classification performance in the homophily-only scenario. The best results are highlighted in bold. Methods with "*" are reported from [48]. Methods with "-MAX" denote the best performance achieved with different pre-trained models.

		3-shot			5-shot		
		Cora	CiteSeer	PubMed	Cora	CiteSeer	PubMed
In-domain	Fine-tuning-MAX*	51.97 ± 2.84	45.08 ± 2.09	65.40 ± 3.00	62.66 ± 3.55	39.54 ± 3.54	**70.91 ± 4.87**
	GPPT-MAX*	43.84 ± 6.11	42.34 ± 8.31	**67.43 ± 2.96**	51.98 ± 3.43	45.77 ± 7.41	66.97 ± 3.70
	AIO-MAX*	48.09 ± 4.83	48.09 ± 8.18	65.79 ± 5.79	30.45 ± 0.19	27.93 ± 10.59	46.16 ± 15.83
Cross-domain	TFSGFM	**66.27 ± 5.14**	**56.73 ± 5.63**	63.30 ± 7.04	**70.55 ± 3.28**	**61.89 ± 3.64**	66.22 ± 5.47

Table 6: Zero-shot graph classification performance in the graph classification scenario. The best results are highlighted in bold. Methods with "*" are reported from [31].

Setting	Methods	IMDB-BINARY		ENZYMES		DD	
		Acc	F1	Acc	F1	Acc	F1
Supervised	GCN*	57.30 ± 0.98	54.62 ± 1.12	20.58 ± 2.00	15.25 ± 3.96	44.74 ± 4.23	55.33 ± 6.22
One-shot	GPPT*	50.15 ± 0.75	44.16 ± 6.70	21.29 ± 3.79	19.87 ± 2.99	51.50 ± 6.54	57.69 ± 6.89
	GPF*	59.65 ± 5.06	56.22 ± 6.17	22.00 ± 1.25	17.34 ± 2.45	48.52 ± 7.11	59.36 ± 1.18
	GPF-plus*	57.93 ± 1.62	55.55 ± 2.03	22.92 ± 1.64	18.39 ± 2.76	46.24 ± 4.86	57.62 ± 2.42
Zero-shot	ULTRA (3g)*	49.25 ± 0.00	38.87 ± 0.00	15.21 ± 0.00	5.84 ± 0.00	63.50 ± 0.00	70.38 ± 0.00
	SCORE*	61.83 ± 1.60	60.91 ± 2.18	22.92 ± 2.03	**21.77 ± 2.17**	69.85 ± 0.51	69.96 ± 0.74
	TFSGFM	**63.67 ± 1.32**	**63.25 ± 1.37**	**23.07 ± 1.71**	19.22 ± 1.83	**73.54 ± 0.78**	**72.84 ± 0.73**

Furthermore, as the number of source domains increases, the performance of TFSGFM exhibits an upward trend, which confirms that multi-domain pretraining helps the model acquire richer and more generalizable graph knowledge. In addition, we find that when the target domain is a homophilic (or heterophilic) dataset, incorporating heterophilic (or homophilic) graphs into the source domain leads to a notable performance improvement. This observation aligns with our conclusion in the homophily-only scenario: injecting appropriate heterophilic (or homophilic) information during training may help the model better capture heterophily within homophilic graphs (or homophily within heterophilic graphs), thus enhancing the model's representation capabilities.

5.8 Hyper-Parameter Analysis

We investigated the effect of the feature splitting ratio ρ on the performance of our model by conducting experiments with $\rho \in \{0.125, 0.25, 0.5\}$. The results, illustrated in Figure 3, exhibit a rise-then-fall trend as ρ increases. This phenomenon can be attributed to the trade-off between homophilic and heterophilic information: a smaller ρ results in insufficient homophilic features, weakening the model's ability to capture homophilic information; conversely, a larger ρ decreases the proportion of heterophilic features, limiting the model's ability to capture heterophilic information. When $\rho = 0.25$, an optimal balance between these two types of information is achieved, leading to the best overall performance across all datasets.

Table 7: Ablation study on the number of source domains. The best results are highlighted in bold. Methods with "*" are reported from [41].

Source domain	Accuracy (%)		
	GCOPE*	MDGPT*	TFSGFM
Target domain: PubMed			
Cora	40.92	45.59	**56.69**
Cora + CiteSeer	40.58	46.72	**56.46**
Cora + CiteSeer + Wisconsin	43.14	48.51	**57.54**
Target domain: Wisconsin			
Texas	27.37	27.75	**45.39**
Texas + Cornell	27.55	27.73	**48.67**
Texas + Cornell + Cora	27.14	29.78	**53.03**
Target domain: Chameleon			
Wisconsin	23.14	24.81	**25.39**
Wisconsin + Squirrel	24.13	24.95	**25.53**
Wisconsin + Squirrel + Cora	22.61	25.36	**28.28**

6 Conclusion

In this paper, we propose the Topology-aware Feature Sorting Graph Foundation Model (TFSGFM), a novel graph foundation

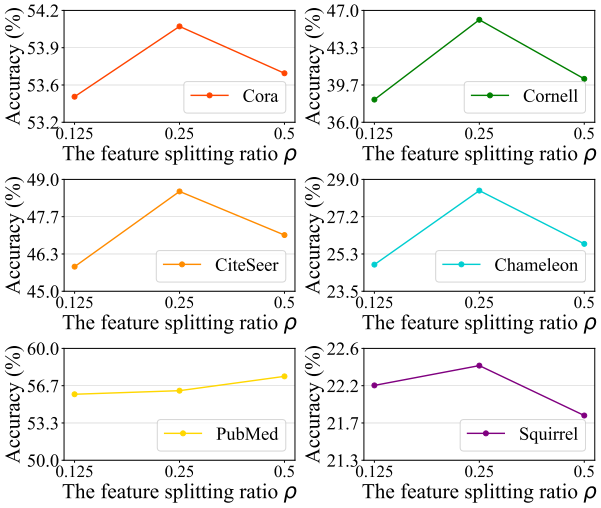

Figure 3: Sensitivity analysis of the feature splitting ratio ρ.

model that enables unified modeling of both homophilic and heterophilic graphs through the topology-aware feature sorting mechanism. By evaluating the homophily of each feature dimension based on graph structure, our method identifies and splits homophilic and heterophilic information within a graph, achieving co-unification of structure and features. Extensive experiments across multiple cross-domain few-shot tasks demonstrate the strong generalizability and adaptability of TFSGFM. We hope that the topology-aware feature sorting mechanism introduced in this work provides meaningful insights for the further development of graph foundation models. Furthermore, we provide a summary of the limitations of this work in Appendix C, which we hope will guide future extensions.

References

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[2] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alexander J. Smola, and Hans-Peter Kriegel. 2005. Protein function prediction via graph kernels. In *Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005*. 47–56. doi:10.1093/BIOINFORMATICS/BTI1007

[3] Jing Chen, Quanzhen Chen, Feng Jiang, Xuyao Guo, Kaiyue Sha, and Yuxuan Wang. 2024. SCN_GNN: A GNN-based fraud detection algorithm combining strong node and graph topology information. *Expert Syst. Appl.* 237, Part C (2024), 121643. doi:10.1016/J.ESWA.2023.121643

[4] Junliang Du, Guiran Liu, Jia Gao, Xiaoxuan Liao, Jiacheng Hu, and Linxiao Wu. 2024. Graph Neural Network-Based Entity Extraction and Relationship Reasoning in Complex Knowledge Graphs. *CoRR* abs/2411.15195 (2024). arXiv:2411.15195 doi:10.48550/ARXIV.2411.15195

[5] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 417–426. doi:10.1145/3308558.3313488

[6] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2023. Universal Prompt Tuning for Graph Neural Networks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).

[7] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. 2021. SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 22667–22681. https://proceedings.neurips.cc/paper/2021/hash/bf499a12e998d178afd964adf64a60cb-Abstract.html

[8] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards Foundation Models for Knowledge Graph Reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=jVEoydFOl9

[9] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li. 2023. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *Trans. Recomm. Syst.* 1, 1 (2023), 1–51. doi:10.1145/3568022

[10] Dongxiao He, Lianze Shan, Jitao Zhao, Hengrui Zhang, Zhen Wang, and Weixiong Zhang. 2024. Exploitation of a Latent Mechanism in Graph Contrastive Learning: Representation Scattering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/d0ffb35aaa7faa894afe5060c694d674-Abstract-Conference.html

[11] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker (Eds.). ACM, 135–142. doi:10.1145/1864708.1864736

[12] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph Structure Learning for Robust Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 66–74. doi:10.1145/3394486.3403049

[13] Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. 2024. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *J. Big Data* 11, 1 (2024), 18. doi:10.1186/S40537-023-00876-4

[14] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016). arXiv:1611.07308 http://arxiv.org/abs/1611.07308

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[16] Pengbo Li, Hang Yu, Xiangfeng Luo, and Jia Wu. 2023. LGM-GNN: A Local and Global Aware Memory-Based Graph Neural Network for Fraud Detection. *IEEE Trans. Big Data* 9, 4 (2023), 1116–1127. doi:10.1109/TBDATA.2023.3234529

[17] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2023. One for All: Towards Training One Graph Model for All Classification Tasks. *CoRR* abs/2310.00149 (2023). arXiv:2310.00149 doi:10.48550/ARXIV.2310.00149

[18] Shuwen Liu, Bernardo Cuenca Grau, Ian Horrocks, and Egor V. Kostylev. 2021. INDIGO: GNN-Based Inductive Knowledge Graph Completion Using Pair-Wise Encoding. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 2034–2045. https://proceedings.neurips.cc/paper/2021/hash/0fd600c953cde8121262e322ef09f70e-Abstract.html

[19] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 417–428. doi:10.1145/3543507.3583386

[20] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto (Eds.). ACM, 43–52. doi:10.1145/2766462.2767755

[21] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the Construction of Internet Portals with Machine Learning. *Inf. Retr.* 3, 2 (2000), 127–163. doi:10.1023/A:1009953814988

[22] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=S1e2agrFvS

[23] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-Scale attributed node embedding. *J. Complex Networks* 9, 2 (2021). doi:10.1093/COMNET/CNAB014

[24] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. *AI Mag.* 29, 3 (2008), 93–106. doi:10.1609/AIMAG.V29I3.2157

[25] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR* abs/1811.05868 (2018). arXiv:1811.05868 http://arxiv.org/abs/1811.05868

[26] Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip S. Yu. 2025. RiemannGFM: Learning a Graph Foundation Model from Riemannian Geometry. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (Eds.). ACM, 1154–1165. doi:10.1145/3696410.3714952

[27] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 1717–1727. doi:10.1145/3534678.3539249

[28] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2024. All in One: Multi-task Prompting for Graph Neural Networks (Extended Abstract). In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 8460–8465. https://www.ijcai.org/proceedings/2024/942

[29] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[30] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rklz9iAcKQ

[31] Kai Wang and Siqiang Luo. 2024. Towards Graph Foundation Models: The Perspective of Zero-shot Reasoning on Knowledge Graphs. *CoRR* abs/2410.12609 (2024). arXiv:2410.12609 doi:10.48550/ARXIV.2410.12609

[32] Song Wang, Yushun Dong, Xiao Huang, Chen Chen, and Jundong Li. 2022. FAITH: Few-Shot Graph Classification with Hierarchical Task Graphs. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 2284–2290. doi:10.24963/IJCAI.2022/317

[33] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. 2025. Multi-Domain Graph Foundation Models: Robust Knowledge Transfer via Topology Alignment. *CoRR* abs/2502.02017 (2025). arXiv:2502.02017 doi:10.48550/ARXIV.2502.02017

[34] Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, Qingkai Zeng, Xiusi Chen, Jianan Zhao, Jundong Li, Meng Jiang, Pietro Lio, Nitesh V. Chawla, Chuxu Zhang, and Yanfang Ye. 2025. Graph Foundation Models: A Comprehensive Survey. *CoRR* abs/2505.15116 (2025). arXiv:2505.15116 doi:10.48550/ARXIV.2505.15116

[35] Man Wu, Xin Zheng, Qin Zhang, Xiao Shen, Xiong Luo, Xingquan Zhu, and Shirui Pan. 2024. Graph Learning under Distribution Shifts: A Comprehensive Survey on Domain Adaptation, Out-of-distribution, and Continual Learning. *CoRR* abs/2402.16374 (2024). arXiv:2402.16374 doi:10.48550/ARXIV.2402.16374

[36] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2023. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 5 (2023), 97:1–97:37. doi:10.1145/3535101

[37] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 1070–1079. https://doi.org/10.1145/3485447.3512156

[38] Pinar Yanardag and S. V. N. Vishwanathan. 2015. Deep Graph Kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, Longbing Cao, Chengqi Zhang, Thorsten Joachims, Geoffrey I. Webb, Dragos D. Margineantu, and Graham Williams (Eds.). ACM, 1365–1374. doi:10.1145/2783258.2783417

[39] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html

[40] Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. 2025. SAMGPT: Text-free Graph Foundation Model for Multi-domain Pre-training and Cross-domain Adaptation. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (Eds.). ACM, 1142–1153. doi:10.1145/3696410.3714828

[41] Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. 2024. Text-Free Multi-domain Graph Pre-training: Toward Graph Foundation Models. *CoRR* abs/2405.13934 (2024). arXiv:2405.13934 doi:10.48550/ARXIV.2405.13934

[42] Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 2024. All in One and One for All: A Simple yet Effective Method towards Cross-domain Graph Pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, Ricardo Baeza-Yates and Francesco Bonchi (Eds.). ACM, 4443–4454. doi:10.1145/3637528.3671913

[43] Jitao Zhao, Di Jin, Meng Ge, Lianze Shan, Xin Wang, Dongxiao He, and Zhiyong Feng. 2024. FUG: Feature-Universal Graph Contrastive Pre-training for Graphs with Diverse Node Features. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper_files/paper/2024/hash/075b7d4bd7fc32d9cf468a7b67c38d15-Abstract-Conference.html

[44] Jianan Zhao, Hesham Mostafa, Mikhail Galkin, Michael M. Bronstein, Zhaocheng Zhu, and Jian Tang. 2024. GraphAny: A Foundation Model for Node Classification on Any Graph. *CoRR* abs/2405.20445 (2024). arXiv:2405.20445 doi:10.48550/ARXIV.2405.20445

[45] Zihao Zhao, Xinlong Zhai, Jinyu Yang, and Chuan Shi. 2025. Towards Text-free Graph Foundation Models: Rethinking Multi-Domain Graph Contrastive Learning. *CoRR* abs/2506.22510 (2025). arXiv:2506.22510 doi:10.48550/ARXIV.2506.22510

[46] Yizhen Zheng, Shirui Pan, Vincent Lee, Yu Zheng, and Philip S Yu. 2022. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems* 35 (2022), 10809–10820.

[47] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. *CoRR* abs/2006.04131 (2020). arXiv:2006.04131 https://arxiv.org/abs/2006.04131

[48] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. 2024. ProG: A Graph Prompt Learning Benchmark. *CoRR* abs/2406.05346 (2024). arXiv:2406.05346 doi:10.48550/ARXIV.2406.05346

A Detailed Dataset Descriptions

The datasets used in our experiments can be categorized into three types: homophilic graph datasets, heterophilic graph datasets, and graph classification datasets. The homophilic graph datasets include (1) Cora [21] and (2) CiteSeer [24], which are citation networks in the computer science domain; (3) PubMed [11], a citation network in the biomedical domain; (4) Photo [25] and (5) Computers [20], which are co-purchase networks in different product categories. The heterophilic graph datasets include (1) Cornell [22], (2) Chameleon [23], (3) Squirrel [23], (4) Wisconsin [22], and (5) Texas [22], all of which are web page link networks on various topics, commonly used to benchmark heterophilic graph learning. The graph classification datasets include (1) IMDB-BINARY [38], which is a movie collaboration dataset, used for binary classification; (2) ENZYMES [32] and (3) DD [2], which are protein graph datasets used for biochemical classification tasks. It is worth noting that the IMDB-BINARY dataset does not provide raw node features; therefore, we generated them by applying one-hot encoding to the node degrees.

B Detailed Experimental Settings

B.1 Implementation Details

In the m-shot node classification tasks (with $m \in \{1, 3, 5\}$), we follow the experimental settings of MDGPT [41] and MDGFM [33]. We first pre-train the TFSGFM model on the training datasets. In the downstream phase, we freeze the model parameters and sample m examples per class. When computing embeddings in this phase, we follow the approach used in [43, 46], applying topological propagation on the original graph structure to strengthen the similarity between connected nodes. The embeddings of these m samples are

Table 8: Hyper-parameters settings of TFSGFM.

Scenarios	Dataset	Learning rate	Weight decay	Epochs	λ	d	Hidden_Dim	Out_Dim	k	ρ	Num_Hop
Mixed homophily-heterophily scenario	Cora	0.00001	5e-4	400	0.1	256	256	256	10	0.25	5
	CiteSeer	0.00001	5e-4	400	0.1	256	256	256	10	0.25	5
	PubMed	0.00001	5e-4	400	0.1	256	256	256	10	0.25	5
	Cornell	0.00001	5e-4	500	0.1	16	256	32	10	0.25	0
	Squirrel	0.00001	5e-4	400	0.1	512	16	256	10	0.25	0
	Chameleon	0.00001	5e-4	300	0.1	16	2048	16	10	0.25	4
Homophily-only scenario	All	0.00001	5e-4	500	0.1	256	256	256	10	0.25	5
Graph classification scenario	All	0.000001	5e-4	100	0.1	512	256	128	1	0.25	0

then averaged to form the prototype vector for each class. Finally, classification is performed based on the similarity between the embedding of each query node and the class prototypes.

In the graph classification task, we pre-train the TFSGFM model at the subgraph level. In the downstream phase, TFSGFM generates node embeddings for each node, followed by average pooling to obtain the final subgraph representations. During testing, we follow the experimental settings of SCORE [31], where the prototype vectors for each class are computed from the training set, and classification is performed by measuring the similarity between the subgraph representations and the class prototypes.

All experiments were conducted on a device equipped with an Intel 12400 CPU and NVIDIA RTX 3090 GPU.

B.2 Hyper-Parameter Settings

In the homophily-only scenario and the graph classification scenario, we use a two-layer GCN as the graph encoder for both the homophilic and heterophilic subgraphs. In the mixed homophily-heterophily scenario, to better capture the homophilic information within the graph, we employ a three-layer GCN to encode the homophilic subgraph and a two-layer GCN to encode the heterophilic subgraph. For simplicity, these two graph encoders share the same hidden and output dimensions.

The other hyper-parameters of TFSGFM are summarized in Table 8, where d denotes the unified feature dimension after feature alignment, Hidden_Dim denotes the hidden layer dimension of the graph encoder, Out_Dim denotes the output layer dimension, and k refers to the hyper-parameter used in k-nearest neighbors (kNN). To ensure experimental reproducibility, the random seed is fixed to 0 for all experiments. For the homophily-only scenario and the graph classification scenario, we follow the experimental settings of MDGPT [41] and SCORE [31], where different datasets share the same hyper-parameters. For the mixed homophily-heterophily scenario, we follow the experimental settings of MDGFM [33], where different parameters are assigned to different datasets.

C Limitations

Although the proposed method demonstrates strong performance, it still has certain limitations. The topology-aware feature sorting mechanism relies on access to both the complete graph structure and the full feature matrix to accurately compute the feature homophily score for each feature dimension. Consequently, in scenarios where the full graph is unavailable, such as in distributed systems where only partial subgraphs can be accessed, the effectiveness of TFSGFM may degrade. Addressing this limitation, for instance, by developing a sorting mechanism applicable to federated learning, represents an important direction for future work.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009