SA^{2}GFM: Enhancing Robust Graph Foundation Models with Structure-Aware Semantic Augmentation

Junhua Shi[1], Qingyun Sun[1]*, Haonan Yuan[1], Xingcheng Fu[2]

[1]SKLCCSE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Key Lab of Education Blockchain and Intelligent Technology, Guangxi Normal University, Guilin, China
{shijunhua, sunqy, yuanhn}@buaa.edu.cn, fuxc@gxnu.edu.cn

Abstract

While Graph Foundation Models (GFMs) have achieved notable progress across diverse tasks recently, their robustness under domain noise, structural perturbations, and even adversarial attacks remains largely underexplored. A core limitation lies in the inadequate modeling of hierarchical structural semantics, which are intrinsic priors and critical for generalization. In this work, we propose **SA^{2}GFM**, a robust **GFM** framework that enhances the domain-adaptable representations through **S**tructure-**A**ware **S**emantic **A**ugmentation. First, to embed the hierarchical structural priors, we transform entropy-based encoding trees into structure-aware textual prompts for feature augmentation. The enriched inputs are processed by a novel self-supervised Information Bottleneck mechanism that distills the robust and transferable representations through structure-guided compression. To mitigate the negative transfer in cross-domain adaptation, we develop an expert adaptive routing mechanism that integrates a mixture-of-experts architecture with a null expert design. To enable efficient downstream adaptation, we propose a fine-tuning module that efficiently optimizes the hierarchical structures through the joint intra- and inter-community structure learning. Extensive experiments validate the superiority of SA^{2}GFM over effectiveness and robustness against random noise and adversarial perturbations on node and graph classification compared with 9 state-of-the-art baselines.

1 Introduction

Graph Neural Networks (GNNs) have achieved impressive success in learning graph-structured data (Kipf and Welling 2017; Veličković et al. 2018) across a wide range of domains. However, traditional GNNs are designed for specific tasks and datasets, limiting their ability to transfer learned knowledge to unseen domains (Hu et al. 2020b). To address this limitation, recent work has advanced towards Graph Foundation Models (GFMs), which aim to learn general-purpose graph representations via large-scale multi-domain pre-training and efficient downstream adaptation with limited supervision (Yu et al. 2024; Wang et al. 2025; Yuan et al. 2025b). The paradigm of "pretrain-then-finetune" promises to build adaptable GFMs that generalize across a broad spectrum of graph-based tasks and domains (Liu et al. 2023a).

*Corresponding author.

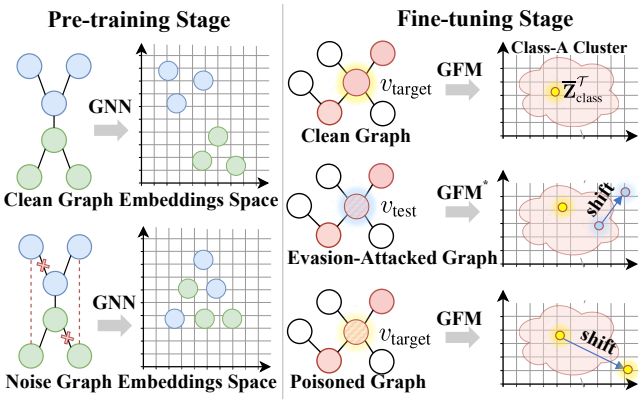

Figure 1: Effects of different attacks in pre-training and fine-tuning. The asterisk (∗) denotes the trained module.

As shown in Figure 1, the robustness of Graph Foundation Models (GFMs) under noisy and adversarial perturbations remains largely underexplored (Wang et al. 2025). Particularly, a fundamental limitation arises from the insufficient modeling of hierarchical structural semantics: the inductive priors that are crucial for ensuring both adaptation and robustness. Existing GFMs predominantly adopt shallow message-passing GNNs as their backbone architectures, which are theoretically limited by the 1-Weisfeiler-Lehman (1-WL) test (Xu et al. 2019; Morris et al. 2019), rendering them incapable of distinguishing structurally similar yet semantically distinct patterns. Consequently, their learned representations often neglect long-range dependencies and higher-order structure semantics, leaving them vulnerable to noise and misalignment under real-world deployment. Moreover, most GFMs overlook the opportunity to embed structure-aware semantics (Yu et al. 2024; Yuan et al. 2025b). They tend to directly encode raw node attributes, which are often incomplete or noisy, while ignoring global community hierarchies that provide stable semantic anchors across domains. Without explicit modeling of such structure-induced semantics.

Beyond architectural expressiveness, GFMs often struggle in real-world deployment where robustness is essential. Existing domain adaptation GFMs (*e.g.*, MDGPT (Yu et al. 2024), BRIDGE) (Yuan et al. 2025b) rely on over-idealized

assumptions like dimension alignment or domain invariance, which frequently break across heterogeneous graphs (Wang et al. 2024). Meanwhile, some other GFMs claim to be perturbation robust by applying global structure learning (Wang et al. 2025), but they often incur high computational costs and remain fragile to localized perturbations or adversarial attacks (Kataria, Kumar, and Jayadeva 2024; Zhang et al. 2024). A more critical yet overlooked problem is negative transfer: when structural or semantic gaps between domains are large, the naïve source aggregation can severely degrade downstream performance, especially under dynamic or non-stationary settings (Dan et al. 2024; Yuan et al. 2023).

The aforementioned challenges expose three critical bottlenecks for constructing a robust GFM: **(1) Pre-training: How to embed structural semantic priors while preserving information purity?** Raw node features are insufficient for capturing transferable patterns across domains, yet naïve feature enhancement may introduce spurious correlations. A principled mechanism is needed to embed high-level structural priors while compressing irrelevant noise. **(2) Knowledge fusion: How to mitigate negative transfer under domain discrepancy?** The diverse nature of real-world graphs necessitates adaptive fusion strategies that can selectively route relevant knowledge and suppress misleading signals from unrelated sources. (3) **Downstream fine-tuning: How to efficiently optimize structures under noise and adversarial perturbations?** Topology is the most vulnerable aspect of graphs, yet precise and efficient structure refinement remains an open problem due to the high cost and fragility of global structure learning strategies.

To address the challenges, we propose a novel **SA^{2}GFM**, a robust **G**raph **F**oundation **M**odel empowered by **S**tructure-**A**ware **S**emantic **A**ugmentation. Our key insight is to embed hierarchical semantic priors using entropy-based encoding trees, which generate structure-aware prompts to enrich node features. The augmented inputs are processed by a self-supervised Information Bottleneck mechanism to learn robust, domain-adaptable representations by compressing redundancy and preserving label-relevant information. To mitigate domain mismatch, we adopt an expert routing module with a Mixture-of-Experts and a null expert to suppress irrelevant sources. Finally, a lightweight fine-tuning strategy refines graph structures through hierarchical optimization for improved robustness. **Our contributions are:**

- We propose SA^{2}GFM, a robust GFM framework by synergistically addressing key interrelated challenges in feature enhancement, principled and efficient structure optimization, and knowledge fusion.
- It integrates a structure-aware pre-training strategy utilizing encoding tree textual prompts and self-supervised Information Bottleneck, a domain-adaptive MoE with a null expert for transfer mitigation, and an efficient fine-tuning module for hierarchical structure refinement.
- Extensive experiments show superiority of SA^{2}GFM over effectiveness and robustness against noise and adversarial perturbations on node and graph classification compared with 9 state-of-the-art baselines.

2 Related Work

2.1 Multi-domain Graph Foundation Models

Multi-domain pre-training is fundamental for constructing graph foundation models capable of adapting across heterogeneous graphs (Yuan et al. 2025c,a). Existing methods typically align either node features, using domain tokens (Yu et al. 2024) or domain-invariant aligners (Yuan et al. 2025b), or graph structures via Graph Structure Learning (GSL) (Wang et al. 2025). Some approaches also explore flexible transfer frameworks to reduce negative transfer (Ju et al. 2025). However, most of these efforts address feature and structure adaptation in isolation, lacking a unified perspective that jointly considers both levels. Therefore, a more integrated approach is needed to simultaneously optimize both feature and structure adaptation, providing a comprehensive solution for multi-domain graph learning.

2.2 Robust Graph Representation Learning

The line of research aims to improve stability under feature and structure noise. For feature robustness, prior work leverages data augmentation (Zheng, Wang, and Liu 2024; Huang et al. 2025) and adversarial training (Lee and Park 2025). For structural robustness, Graph Structure Learning (GSL) is widely used to counter topological perturbations (Zügner, Akbarnejad, and Günnemann 2018; Yuan et al. 2024). However, most methods treat feature and structure separately, and many GSL techniques remain costly and coarse-grained.

2.3 Graph Strcutural Entropy

Graph Structural Entropy is an information-theoretic measure that captures the hierarchical organization of a graph through community encoding (Li and Pan 2016). This hierarchy has shown promise in guiding representation learning (Bai et al. 2024), yet its use for enhancing raw node features and improving the robustness of graph foundation models remains largely unexplored.

3 Preliminary

Notations. A graph is denoted as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. Let $\mathbf{A} \in \{0,1\}^{N\times N}$ be the adjacency matrix and $\mathbf{X} \in \mathbb{R}^{N\times d_i}$ be the node feature matrix, where $N = |\mathcal{V}|$ is the number of nodes and d_i denotes input dimension. $\mathbf{Z}$ is hidden embeddings of $\mathbf{X}$.

Problem Settings. Our proposed SA^{2}GFM operates under the multi-domain pre-training and few-shot fine-tuning paradigm. Given n source graphs $\{G^{\mathcal{S}}\} \in \mathcal{G}^{\mathcal{S}}$ from multiple domains $\{D^{\mathcal{S}}\} \in \mathcal{D}^{\mathcal{S}}$ with their labels $\{Y^{\mathcal{S}}\} \in \mathcal{Y}^{\mathcal{S}}$, the pre-training goal is to train a graph learner $h = g(\mathcal{F}_{\mathbf{\Theta}}(\cdot))$ on multi-domain graph datasets, after which the pre-training parameter $\mathbf{\Theta}^\star$ is frozen. During fine-tuning, given a set of target graphs $\{G^{\mathcal{T}}\} \in \mathcal{G}^{\mathcal{T}}$ (potentially corrupted by noise or adversarial attacks) from target domain $\{D^{\mathcal{T}}\} \in \mathcal{D}^{\mathcal{T}}$ (seen or unseen) with m accessible labels $\{Y^{\mathcal{T}}\} \in \mathcal{Y}^{\mathcal{T}}$ under m-shot setting ($m \ll n$), the ultimate goal is to leverage the pre-trained $\mathbf{\Theta}^\star$ to achieve robust performance on the unlabeled nodes of the target graph.

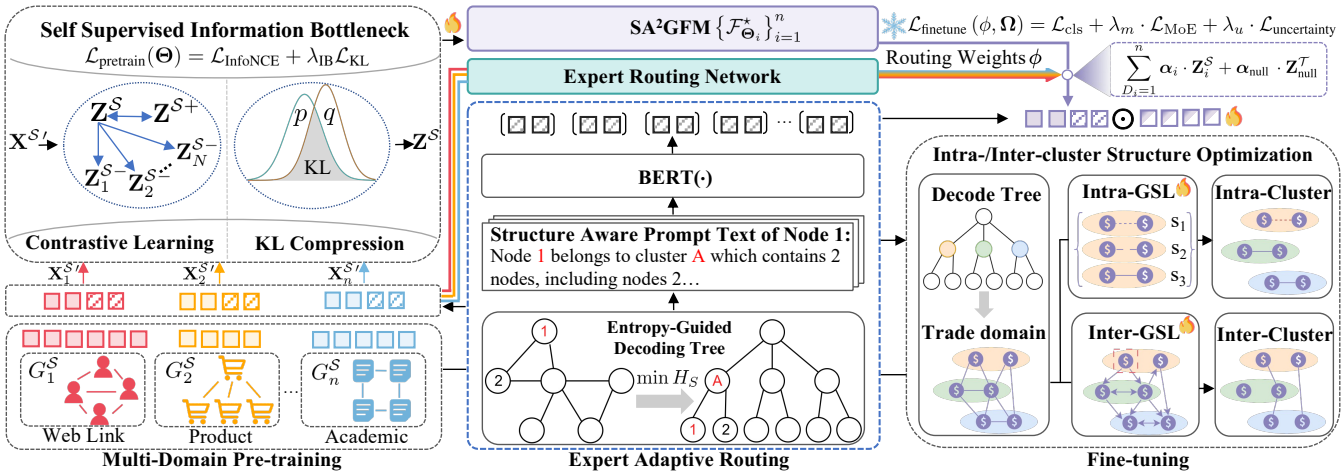

Figure 2: Overview of the SA^{2}GFM framework. The framework first pre-trains an ensemble of expert encoders on multiple source domains using Structure-Aware Semantic Augmentation and an Information Bottleneck objective. During fine-tuning, a novel Expert Adaptive Routing mechanism selects knowledge to mitigate negative transfer, while a Hierarchical Structure Optimization module refines the target graph's topology before final prediction.

4 Proposed Framework: SA^{2}GFM

In this section, we introduce the proposed SA^{2}GFM with its framework illustrated in Figure 2.

4.1 Multi-domain Pre-training with Self-Supervised Information Bottleneck

We begin by encoding structural priors into node features via textual prompts derived from entropy-based encoding trees.

Structure-Aware Semantic Augmentation. Real-world graphs exhibit rich hierarchical priors that are not explicitly encoded in node. To expose such a structure to encoders, we propose to convert the latent community hierarchy into textual prompts, which allows the pre-trained language models to serve as a bridge between structural regularities and semantic representations. To achieve this, we leverage the theory of graph structural entropy (Long et al. 2020), which identifies an optimal recursive clustering by minimizing entropy over volume-weighted partitions. Given a partition $\mathcal{C}^{\mathcal{S}} = \{C_1, \cdots, C_K\}$ of graph $G^{\mathcal{S}}$, the entropy is:

$$H_S(G^{\mathcal{S}}) = -\sum_{k=1}^{K} \frac{\mathrm{Vol}(C_k)}{\mathrm{Vol}(G^{\mathcal{S}})} \log \frac{\mathrm{Vol}(C_k)}{\mathrm{Vol}(G^{\mathcal{S}})}, \quad (1)$$

where $\mathrm{Vol}(C_k)$ represents the sum of degrees in cluster C_k. Based on this tree, we construct a structured textual prompt $\mathbf{t}_i$ for node v_i that captures its structural role, such as: "*There are K structural clusters. Node v_i belongs to cluster C_k, which contains N_K nodes, including $v_i, \ldots, v_j$.*". As shown in Figure 3, the encoding tree guides prompt generation by selecting a node's local cluster, identifying its peer nodes, and transforming this information into language models. This prompt is then embedded via BERT and fused:

$$\mathbf{x}_i^{\mathcal{S}\prime} = \mathrm{SVD}(\mathbf{x}_i^{\mathcal{S}}) \oplus \mathrm{SVD}(\mathrm{BERT}(\mathbf{t}_i)), \quad (2)$$

where $\mathrm{SVD}(\cdot)$ is implemented by truncated SVD (Stewart 1993) that aligns feature dimensions as d_0.

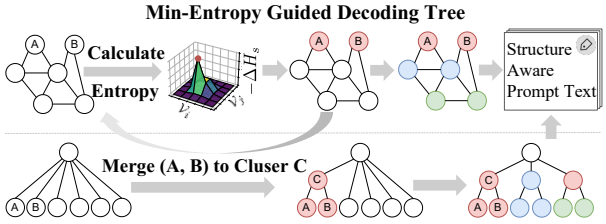

Figure 3: Construct code tree by minimizing structure entropy to generate structure aware prompt text.

Self-Supervised Information Bottleneck. Given aligned features $\mathbf{X}^{\mathcal{S}\prime}$, our goal is to learn the robust and compressed representations $\mathbf{Z}^{\mathcal{S}} = \mathcal{F}_{\mathbf{\Theta}}(\mathbf{X}^{\mathcal{S}\prime}, \mathbf{A}^{\mathcal{S}}) \in \mathbb{R}^d$ that retain task-relevant semantics while suppressing noise. Following the Information Bottleneck (IB) principle (Tishby, Pereira, and Bialek 1999), we formulate a self-supervised objective to maximize consistency between similar nodes while compressing redundant input information. Formally, we define the self-supervised IB (SS-IB) objective as:

$$\mathcal{L}_{\text{SS-IB}} = -I(\mathbf{Z}^{\mathcal{S}}; \mathbf{Z}^{\mathcal{S}+}) + \beta \cdot I(\mathbf{Z}^{\mathcal{S}}; \mathbf{X}^{\mathcal{S}\prime}), \quad (3)$$

where the prediction term $I(\mathbf{Z}^{\mathcal{S}}; \mathbf{Z}^{\mathcal{S}+})$ maximizes consistency between an anchor $\mathbf{Z}^{\mathcal{S}}$ and its positive sample $\mathbf{Z}^{\mathcal{S}+}$, which is sampled from its structural neighborhood based on the encoding tree. The compression term $I(\mathbf{Z}^{\mathcal{S}}; \mathbf{X}^{\mathcal{S}\prime})$ enforces compression by limiting the information retained from the input $\mathbf{X}^{\mathcal{S}\prime}$. β is a balancing hyperparameter.

As directly optimizing Eq. (3) is generally intractable, we adopt variational bounds to implement $\mathcal{L}_{\text{SS-IB}}$.

Derivation 1 (Lower Bound of $I(\mathbf{Z}^{\mathcal{S}}; \mathbf{Z}^{\mathcal{S}+})$)**.** Prediction term $I(\mathbf{Z}^{\mathcal{S}}; \mathbf{Z}^{\mathcal{S}+})$ is lower-bounded by InfoNCE (Oord, Li, and Vinyals 2018). For each anchor v_i and positive node v_i^+:

$$I(\mathbf{Z}^{\mathcal{S}};\mathbf{Z}^{\mathcal{S}+}) \geqslant -\mathcal{L}_{\text{InfoNCE}} = \frac{1}{N^+}\sum\nolimits_{i=1}^{N^+}\log\frac{\exp(\langle\mathbf{Z}_i^{\mathcal{S}},\mathbf{Z}_i^{\mathcal{S}+}\rangle/\tau)}{\sum_{j=0}^{N^-}\exp(\langle\mathbf{Z}_i^{\mathcal{S}},\mathbf{Z}_j^{\mathcal{S}}\rangle/\tau)}, \quad (4)$$

where N^+ and N^- are the number of positive and negative samples, which are non-redundantly and consistently sampled from the source domain depending on whether there is a direct link (Liu et al. 2023b). τ is a temperature.

Derivation 2 (Upper Bound of $I(\mathbf{Z}^{\mathcal{S}};\mathbf{X}^{\mathcal{S}\prime})$)**.** Compression term $I(\mathbf{Z}^{\mathcal{S}};\mathbf{X}^{\mathcal{S}\prime})$ is upper-bounded by the KL divergence between posterior $q(\mathbf{Z}^{\mathcal{S}}|\mathbf{X}^{\mathcal{S}\prime})$ and prior $p(\mathbf{Z}^{\mathcal{S}})$:

$$I(\mathbf{Z}^{\mathcal{S}};\mathbf{X}^{\mathcal{S}\prime}) \leqslant \frac{1}{N}\sum\nolimits_{i=1}^{N}\mathrm{KL}[q(\mathbf{Z}_i^{\mathcal{S}}|\mathbf{X}_i^{\mathcal{S}\prime}) \parallel p(\mathbf{Z}_i^{\mathcal{S}})], \quad (5)$$

We assume a standard Gaussian prior $p(\mathbf{Z}^{\mathcal{S}}) = \mathcal{N}(\mathbf{0},\mathbf{I})$, and apply the graph encoder to produce the mean $\boldsymbol{\mu}_i$ and the log-variance $\log\boldsymbol{\sigma}_i^2$ of $q(\mathbf{Z}_i^{\mathcal{S}}|\mathbf{X}_i^{\mathcal{S}\prime})$ for each node v_i, from which we sample $\mathbf{Z}_i^{\mathcal{S}}$ via reparameterization trick:

$$\mathbf{Z}_i^{\mathcal{S}} \sim q(\mathbf{Z}_i^{\mathcal{S}}|\mathbf{X}_j^{\mathcal{S}\prime}) = \mathcal{N}\big(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)\big). \quad (6)$$

Pre-training Objective. By integrating these bounds, we transform $\mathcal{L}_{\text{SS-IB}}$ in Eq. (3) into a tractable objective given n source domain graphs with trade-off coefficient λ_{IB}:

$$\mathcal{L}_{\text{pretrain}}(\boldsymbol{\Theta}) = \frac{1}{n}\sum\nolimits_{D_i=1}^{n} \underbrace{\mathcal{L}_{\text{InfoNCE}}}_{\text{Eq. (4)}} + \lambda_{\text{IB}} \underbrace{\mathcal{L}_{\text{KL}}}_{\text{Eq. (5)}}, \quad (7)$$

after which the learned $\boldsymbol{\Theta}^\star$ is frozen. Overall, pre-training phase leverages pre-trained language models as a bridge to transform structural priors into node features, enabling the capture of transferable knowledge that lays robust foundation for next expert routing and downstream fine-tuning.

4.2 Expert Adaptive Routing with Negative Transfer Mitigation

To prevent negative transfer, we introduce an expert adaptive routing mechanism, which not only "selects" helpful experts but also "rejects" negative ones via a learnable null expert.

Gated Routing with Null-Expert. To enable fine-grained adaptation across multiple source domains while mitigating negative transfer, we introduce a gated routing network $\mathcal{R}_{\boldsymbol{\phi}}$ that dynamically modulates the contribution of each source expert from $\mathcal{F}^\star_{\boldsymbol{\Theta}} = \{f^\star_{\boldsymbol{\Theta}_1}, f^\star_{\boldsymbol{\Theta}_2}, \cdots, f^\star_{\boldsymbol{\Theta}_n}\}$ via a learned routing weights $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n,\boldsymbol{\alpha}_{\text{null}}\}$, where the null expert explicitly captures and suppresses irrelevant knowledge.

Specifically, we first derive prototypes for the target and source domains. Let $\bar{\mathbf{Z}}^{\mathcal{T}}$ denote the mean-pooled embedding of m ($m \ll n$) few-shot supporting samples (node or graph) in the target graph(s). Similarly, let $\{\bar{\mathbf{Z}}_i^{\mathcal{S}}\}_{i=1}^n$ represent the corresponding prototypes from each of the n source domains. We define the cosine similarity vector $\mathbf{S} \in \mathbb{R}^n$:

$$\mathbf{S}_i = \langle\bar{\mathbf{Z}}^{\mathcal{T}},\bar{\mathbf{Z}}_i^{\mathcal{S}}\rangle/\|\bar{\mathbf{Z}}^{\mathcal{T}}\|\cdot\|\bar{\mathbf{Z}}_i^{\mathcal{S}}\|, \quad (8)$$

where $\mathbf{S}$ is passed through a learnable router $\mathcal{R}_{\boldsymbol{\phi}}$, yielding the unnormalized logits for the gating distribution:

$$\boldsymbol{\alpha} = \mathrm{Softmax}(\mathcal{R}_{\boldsymbol{\phi}}(\mathbf{S})) \in \Delta^{n+1}, \quad (9)$$

where Δ^{n+1} is the $(n+1)$-dimensional probability simplex. Let $\mathbf{Z}_i^{\mathcal{S}} \in \mathbb{R}^{N_i\times d}$ be the node-level embeddings from the i-th expert, and let $\mathbf{Z}^{\mathcal{T}}_{\text{null}}$ denote the embedding generated by a shallow GCN trained solely on the target graph. The final task-specific representation is a convex combination:

$$\mathbf{Z}^{\mathcal{T}}_{\text{final}} = \sum\nolimits_{D_i=1}^{n}\boldsymbol{\alpha}_i\cdot\mathbf{Z}_i^{\mathcal{S}} + \boldsymbol{\alpha}_{\text{null}}\cdot\mathbf{Z}^{\mathcal{T}}_{\text{null}}, \quad (10)$$

which enables the model to selectively integrate transferable domain knowledge, and further assign high mass to null experts when there is no source expert that aligns semantically or structurally, thus mitigating negative transfer.

Uncertainty-Aware Routing Regularization. To inspire confident expert selection and avoid the overly diffuse gating, we incorporate an entropy-based regularization over the mixture-of-experts (MoE) architecture:

$$\mathcal{L}_{\text{MoE}}(\boldsymbol{\phi}) = \mathcal{H}(\boldsymbol{\alpha}) = -\sum\nolimits_{j=1}^{k+1}\boldsymbol{\alpha}_j\log\boldsymbol{\alpha}_j, \quad (11)$$

where $\mathcal{H}(\cdot)$ is the entropy. It penalizes uncertain mixtures and promotes sparse, decisive routing behaviors. In particular, when all source experts are irrelevant, this regularization amplifies the model's preference for the null expert.

4.3 Fine-tuning with Efficient Hierarchical Structure Optimization

While expert routing has aligned the source knowledge, performance still hinges on target graphs. Potential structural noise and adversarial perturbations can greatly hinder message passing. Though structure learning improves robustness (Zügner, Akbarnejad, and Günnemann 2018; Zhu et al. 2021), it is often inefficient and coarse-grained. We thus propose a lightweight structure optimization strategy.

To ensure a seamless transition from pre-training to fine-tuning, we preserve structural consistency by reusing the entropy encoding tree to partition the target graph $G^{\mathcal{T}}$ into a set of clusters $\mathcal{C}^{\mathcal{T}}$. This partition forms the basis for the hierarchical structure optimization during fine-tuning, enabling targeted refinement at both the intra- and inter-cluster levels. While intra-cluster optimization enhances local structural fidelity, inter-cluster optimization focuses on global signal propagation and robustness.

Intra-cluster Structure Learning. For each of cluster $c \in \mathcal{C}^{\mathcal{T}}$, we refine its internal topology by refining edges with multi-head attention. Let $\mathbf{Z}_c^{\mathcal{T}}$ denote the node embedding in cluster c, which are projected to $\mathbf{H}_c = \mathrm{ReLU}(\mathbf{Z}_c^{\mathcal{T}}\mathbf{W}_p)$. Attention matrix $\mathbf{S}_{\text{attn},c}$ is computed via:

$$\mathbf{S}_{\text{attn},c} = \frac{1}{H}\sum\nolimits_{h=1}^{H}\mathrm{Softmax}\Big(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_k}}\Big), \quad (12)$$

where $\mathbf{Q}_h = \mathbf{Z}_c\mathbf{W}_h^Q$, and $\mathbf{K}_h = \mathbf{H}_c\mathbf{W}_h^K$. To enforce the cross-head consistency, we introduce an uncertainty loss that penalizes variance among heads:

$$\mathcal{L}_{\text{uncertainty}} = \frac{\sum_{c\in\mathcal{C}^{\mathcal{T}}}\sum_{i,j\in c, i\neq j}\mathrm{Var}_{h=1}^{H}\big(\mathbf{S}_{ij}^{(h)}\big)}{|\{(i,j)|i,j\in c, c\in\mathcal{C}^{\mathcal{T}}, i\neq j\}|}, \quad (13)$$

where $\mathrm{Var}(\cdot)$ denotes variation. Refined intra-cluster $\mathbf{A}^{\mathcal{T}'}_{\text{intra},c}$ fuses the original $\mathbf{A}^{\mathcal{T}}_c$ and attention-derived scores:

$$\mathbf{A}^{\mathcal{T}'}_{\text{intra},c} = (1 - \mathbf{W}_c) \cdot \mathbf{A}^{\mathcal{T}}_c + \mathbf{W}_c \cdot \mathbf{S}_{\text{attn},c}, \quad (14)$$

where $\mathbf{W}_c$ is a learnable fusion weight per cluster.

Inter-cluster Structure Learning. As intra-cluster learning focuses on refining dense local structures, inter-cluster learning aims to regulate potentially noisy global connections, which are critical for propagation but often include irrelevant edges. To selectively retain informative inter-cluster dependencies, we approximate personalized propagation of neural predictions (Gasteiger, Bojchevski, and Günnemann 2019) to compute a soft influence matrix:

$$\mathbf{S}^{\mathcal{T}} = (1 - \alpha) \cdot \sum\nolimits_{t=0}^{T} (\alpha \cdot \tilde{\mathbf{A}}^{\mathcal{T}})^t, \quad (15)$$

where $\tilde{\mathbf{A}}^{\mathcal{T}}$ is normalized adjacency. We then derive a probabilistic pruning mask via thresholded activation to refine the original inter-cluster structure $\mathbf{A}^{\mathcal{T}}_{\text{inter}}$:

$$\mathbf{A}^{\mathcal{T}'}_{\text{inter}} = \mathbf{A}^{\mathcal{T}}_{\text{inter}} \odot \mathbf{P}^{\mathcal{T}},\ \mathbf{P}^{\mathcal{T}} = \sigma(\mathbf{S}^{\mathcal{T}} - \boldsymbol{\theta}_{\text{thres}}), \quad (16)$$

where σ is the Sigmoid and $\boldsymbol{\theta}_{\text{thres}}$ is a learnable threshold. Let $\mathbf{A}^{\mathcal{T}'}_{\text{intra, full}}$ be the block-diagonal matrix formed by assembling the refined intra-cluster matrices $\{\mathbf{A}^{\mathcal{T}'}_{\text{intra},c}\}_{c \in \mathcal{C}^{\mathcal{T}}}$. Optimized $\mathbf{A}^{\mathcal{T}\prime}$ for target graph is then an adaptive combination:

$$\mathbf{A}^{\mathcal{T}'} = \mathbf{W}_s \cdot \mathbf{A}^{\mathcal{T}'}_{\text{intra, full}} + (1 - \mathbf{W}_s) \cdot \mathbf{A}^{\mathcal{T}'}_{\text{inter}}, \quad (17)$$

where $\mathbf{W}_s$ is a learnable trade-off weights.

Fine-tuning with Prompted Structure. Given $\mathbf{A}^{\mathcal{T}'}$, we embed the learnable prompts to guide downstream adaptation. For each few-shot sample, the prompted embedding is:

$$\mathbf{Z}^{\mathcal{T}}_i = \mathcal{F}^{\star}_{\boldsymbol{\Theta}}\big(\boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}} \odot \mathbf{X}^{\mathcal{T}'}_i,\ \mathbf{A}^{\mathcal{T}'}\big) \in \mathbb{R}^d, \quad (18)$$

where $\boldsymbol{\mathcal{P}}_{\boldsymbol{\Omega}}$ is the learnable prompt and $\mathbf{X}^{\mathcal{T}'}_i$ is the feature aligned similarly as Eq. (1). To maintain consistency with the pre-training objective, we leverage a universal task template based on contrastive loss that aligns query embeddings with class prototypes under the updated structure. Given m few-shot support samples $(G^{\mathcal{T}}_i, Y^{\mathcal{T}}_i)$, the objective is:

$$\mathcal{L}_{\text{cls}}(\boldsymbol{\Omega}) = -\sum\nolimits_{i=1}^{m} \log \frac{\exp\big(\langle \mathbf{Z}^{\mathcal{T}}_i, \bar{\mathbf{Z}}^{\mathcal{T}}_{Y_i} \rangle / \tau\big)}{\sum_{Y_j} \exp\big(\langle \mathbf{Z}^{\mathcal{T}}_i, \bar{\mathbf{Z}}^{\mathcal{T}}_{Y_j} \rangle / \tau\big)}, \quad (19)$$

where $\bar{\mathbf{Z}}^{\mathcal{T}}_{Y_j}$ is prototype of Y_j and $\langle \cdot, \cdot \rangle$ is the inner product. Thus, the overall fine-tuning objective is:

$$\mathcal{L}_{\text{finetune}}(\phi, \boldsymbol{\Omega}) = \mathcal{L}_{\text{cls}} + \lambda_m \cdot \mathcal{L}_{\text{MoE}} + \lambda_u \cdot \mathcal{L}_{\text{uncertainty}}, \quad (20)$$

where λ_1, λ_2 are hyperparameters.

4.4 Complexity Analysis

Suppose f is an L-layer GNN. Then, the per-epoch computational complexity of the pre-training stage is $\mathcal{O}(nL|\mathcal{E}^{\mathcal{S}}|d)$, mainly arising from the SS-IB pre-training objective over n source graphs. The per-epoch complexity of the fine-tuning stage is $\mathcal{O}(nL|\mathcal{E}^{\mathcal{T}}|d + \sum_c |\mathcal{V}^{\mathcal{T}}_c|^2)$, where the main cost comes from the intra-cluster attention (with $|\mathcal{V}^{\mathcal{T}}_c|$ denoting the number of nodes in cluster c). The overall complexity is comparable to that of baselines such as MDGPT, while the structure optimization is more efficient than MDGFM.

5 Experiment

We evaluate the proposed SA^{2}GFM with the following research questions:

- ***RQ1:*** How robust is SA^{2}GFM against random noise and adversarial attacks on features and structures?
- ***RQ2:*** How does the performance change as the intensity of noise or attack severity increases?
- ***RQ3:*** Which part contributes most to the performance?
- ***RQ4:*** How sensitive to hyperparameters fluctuation?
- ***RQ5:*** How well does pre-training support transferable and task-adaptive representations?

5.1 Experimental Settings

Datasets. To highlight multi-domain pre-training capability, we adopt **seven** benchmark datasets from **three** domains:

- **Citation:** `Cora` (McCallum et al. 2000), `CiteSeer` (Giles, Bollacker, and Lawrence 1998), `PubMed` (Sen et al. 2008), `ogbn-arXiv` (Hu et al. 2020a).
- **Products:** `ogbn-Tech`, `ogbn-Home` (Hu et al. 2020a).
- **Web Page:** `Wiki-CS` (Mernyei and Cangea 2020).

Baselines. Include **nine** baselines from **four** categories:

- `GCN` (Kipf et al. 2017), `GAT` (Veličković et al. 2018).
- **Single Domain:** `DGI` (Veličković et al. 2019), `GraphCL` (You et al. 2020), `GraphPrompt` (Liu et al. 2023b).
- **Multi Domain:** `MDGPT` (Yu et al. 2024), `GCOPE` (Zhao et al. 2024), `GraphBridge` (Ju et al. 2025).
- **Robust GFM:** `MDGFM` (Wang et al. 2025).

Pre-training and Fine-tuning. We focus on **two** settings:

- **Cross-Datasets:** Pre-training and fine-tuning on **different** datasets from the **same** domain. While one dataset is selected as the target, the rest are sources.
- **Cross-Domain:** Pre-training and fine-tuning on **different** datasets from the **different** domain.

Few-shot Fine-tuning. We adopt a C-way 5-shot setting for each class. For graph classification, ego-graphs centered on target nodes are used, inheriting center label. Accuracy with standard deviation is used for evaluation over 20 runs.

Noise and Attack. We assess robustness in **two** scenarios:

- **Non-targeted Attack (Noise):** We inject random noise by perturbing either the structure or node features, with a perturbation rate $\lambda \in \{0.4, 0.8\}$. **(1)** For node feature attacks, we randomly inject Gaussian noise. **(2)** For structure attacks, we randomly remove edges from the graph.
- **Targeted Attack:** We employ the powerful NETTACK (Zügner, Akbarnejad, and Günnemann 2018) platform to perform attacks on specific nodes with the default number of perturbations $p \in \{1, 2, 3\}$. **(1)** For evasion attacks, the model is finetuned on a clean support set and attacked during testing on the query set. **(2)** For poisoning attacks, the entire support set is perturbed before fine-tuning.

Source	Cross-Dataset								Cross-Domain							
	Cora CiteSeer ogbn-Home Wiki-CS				Cora CiteSeer PubMed Wiki-CS				Cora CiteSeer PubMed ogbn-Home				ogbn-Home ogbn-Tech Wiki-CS			
Target	PubMed				ogbn-Home				Wiki-CS				ogbn-arxiv			
Noise & Attacks	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.
GCN (backbone)	44.6±8.7	40.9±9.8	36.7±11.0	32.4±9.9	42.9±11.9	51.0±12.9	47.8±13.6	46.4±13.4	38.4±6.9	42.5±9.9	33.8±7.6	38.5±8.8	39.3±5.2	38.8±4.7	36.5±5.3	38.3±6.1
GAT	43.9±7.8	44.2±9.1	37.5±8.6	38.3±8.3	43.9±9.3	53.5±6.7	46.5±6.3	56.0±7.7	37.1±6.2	46.6±4.9	34.4±5.6	37.6±6.7	40.9±5.7	41.4±5.6	34.5±6.7	38.5±5.4
DGI	46.7±7.9	41.2±7.5	42.1±7.6	35.9±8.9	48.7±7.0	61.0±6.2	54.0±4.9	51.0±10.6	45.6±5.6	44.1±7.7	41.1±6.5	45.9±5.1	39.3±4.2	46.6±4.9	36.7±5.4	44.8±4.5
GraphCL	54.9±9.5	48.9±8.8	42.7±7.8	37.5±7.6	47.7±9.1	48.9±7.8	55.9±6.6	53.0±9.1	42.8±6.2	49.3±5.6	42.4±6.1	41.3±4.7	38.8±4.6	43.0±4.9	37.8±4.5	30.0±5.5
GraphPrompt	47.8±8.8	45.2±8.1	45.3±7.5	39.6±6.1	56.0±7.3	58.0±8.4	55.3±8.3	53.5±6.6	51.9±5.9	41.5±7.1	39.4±8.1	32.5±3.7	41.4±4.6	43.4±5.2	39.0±4.7	42.0±4.0
MDGPT	48.6±4.2	56.5±5.5	52.0±6.4	42.1±5.8	59.3±25.1	54.8±25.1	54.0±24.1	57.4±16.4	42.1±7.0	50.4±5.0	49.5±8.9	40.4±7.2	42.5±7.2	46.3±6.7	45.6±5.3	48.3±4.7
GCOPE	53.2±13.3	55.6±11.4	48.7±10.9	44.7±9.4	57.0±23.5	56.0±24.4	55.6±24.3	50.0±23.9	46.6±9.3	46.8±11.8	42.0±10.6	43.6±11.4	48.7±5.9	50.0±7.2	39.1±5.9	49.8±7.6
GraphBridge	51.1±6.8	52.4±4.3	44.0±10.1	46.9±7.4	63.0±4.5	62.2±5.2	57.1±3.7	51.3±6.6	50.1±6.6	47.3±5.7	43.2±10.1	42.4±8.3	46.5±5.7	47.3±6.6	47.5±6.2	44.5±6.0
MDGFM	57.3±6.7	58.4±7.3	53.4±7.3	50.8±5.2	65.0±15.9	65.3±17.0	62.9±15.3	62.1±16.2	53.2±6.9	52.0±5.2	50.1±6.2	46.4±5.7	55.9±4.1	55.7±4.9	50.8±5.0	50.4±4.7
SA²GFM (ours)	**60.0**±5.2	**60.1**±1.4	**56.9**±9.8	**54.5**±1.2	**68.9**±6.2	**69.0**±5.2	**65.9**±2.7	**64.0**±1.3	**55.9**±5.3	**55.9**±1.4	**53.3**±4.9	**50.6**±1.2	**57.9**±7.3	**57.8**±1.4	**55.0**±5.8	**53.0**±1.2

Table 1: Accuracy (% ± std. for 20 runs) of **5-shot node classification**. Best scores are in **bold**, runner-ups are underlined. "feat." and "struct." denote non-targeted attacks ($\lambda = 0.4$), while "evas." and "pois." denote targeted attacks ($p = 3$).

Source	Cross-Dataset								Cross-Domain							
	CiteSeer PubMed ogbn-Home Wiki-CS				Cora CiteSeer PubMed ogbn-Home Wiki-CS				Cora CiteSeer PubMed ogbn-Home				ogbn-Home ogbn-Tech Wiki-CS			
Target	Cora				ogbn-Tech				Wiki-CS				ogbn-arxiv			
Noise & Attacks	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.	feat.	struct.	evas.	pois.
GCN (backbone)	44.9±8.0	51.2±5.8	42.9±5.8	42.3±7.4	68.7±12.9	67.4±11.6	60.7±10.6	59.2±9.8	52.6±10.9	43.8±11.7	40.8±6.7	41.4±9.6	38.4±9.0	41.9±10.1	32.8±7.5	35.7±5.8
GAT	47.1±5.3	48.2±5.8	45.4±5.4	48.2±6.8	71.1±8.4	69.0±10.0	64.6±10.3	59.1±9.7	46.6±8.2	45.2±4.0	45.7±5.4	48.8±5.7	44.5±9.2	43.8±8.5	40.1±8.7	42.2±7.7
DGI	49.6±5.4	53.0±5.2	54.6±6.5	50.7±5.1	71.7±7.0	71.9±8.3	70.2±7.7	62.9±7.4	52.0±5.7	49.2±4.8	45.1±5.1	49.0±4.5	45.5±7.4	47.4±6.8	43.4±5.9	42.4±5.2
GraphCL	54.5±4.6	58.5±5.5	52.2±6.6	43.9±5.5	77.2±5.2	76.9±6.3	73.3±6.6	54.0±6.8	61.6±6.3	50.5±3.5	48.9±4.2	43.1±3.7	51.7±6.6	45.3±6.1	42.8±5.9	49.5±4.9
GraphPrompt	69.0±5.8	57.1±5.3	50.3±4.8	50.8±4.8	83.0±9.6	73.8±10.2	75.3±9.4	61.1±9.5	52.3±6.3	60.1±4.2	47.8±5.0	47.2±5.0	50.5±6.6	48.6±5.0	43.2±5.6	41.2±4.9
MDGPT	63.5±6.2	64.2±6.2	60.0±8.1	53.9±6.1	73.3±8.7	79.1±6.4	73.6±7.3	68.8±8.9	56.0±8.3	53.6±7.4	55.2±7.5	49.0±8.3	47.1±6.8	52.9±8.5	50.1±6.1	43.8±6.3
GCOPE	59.2±10.1	63.5±7.1	58.7±11.0	58.5±8.3	76.3±11.2	81.5±8.5	72.1±6.5	79.6±6.6	50.1±12.1	54.5±17.6	52.4±14.7	40.1±10.6	57.5±12.9	56.8±12.6	52.8±12.1	46.5±9.7
GraphBridge	62.4±5.0	61.0±4.8	62.1±5.7	56.6±5.0	81.0±7.5	80.8±10.6	76.9±7.7	73.9±7.0	61.5±3.6	56.5±3.6	58.6±3.6	48.6±3.6	54.7±9.2	57.9±9.4	55.0±6.5	42.9±7.7
MDGFM	65.9±4.7	67.8±4.9	65.8±6.3	62.6±5.8	85.3±9.7	85.9±7.6	83.1±8.8	**83.5**±9.6	**66.4**±5.2	62.9±4.6	61.5±6.7	56.5±8.0	62.0±6.0	63.0±5.0	59.2±5.5	56.7±6.1
SA²GFM (ours)	**69.5**±6.5	**69.7**±5.5	**68.4**±1.5	**64.6**±1.4	**87.7**±9.5	**87.8**±9.4	**86.7**±6.5	82.6±1.4	64.4±8.5	**64.2**±6.4	**63.5**±3.5	**59.0**±1.2	**63.9**±6.4	**63.9**±5.4	**62.8**±9.4	**58.1**±1.2

Table 2: Accuracy (% ± std. over 20 runs) of **5-shot graph classification**. Notations are consistent with Table 1.

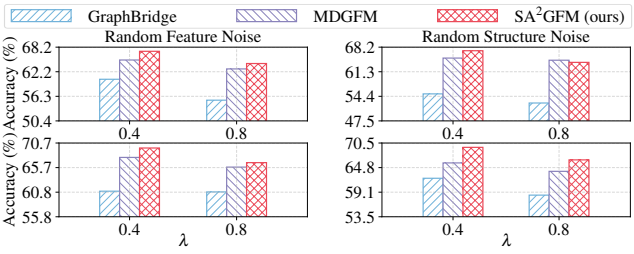

Figure 4: Performance on Cora with random noise perturbations (λ): node (top) and graph (bottom) classification.

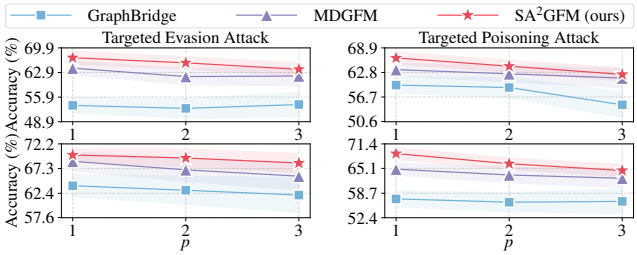

Figure 5: Performance on Cora with adversarial attack degree (p): node (top) and graph (bottom) classification.

5.2 *RQ1:* Against Noise and Adversarial Attacks

Results (Tables 1, Table 2) demonstrate that: ❶ Under both the node and graph classification, SA²GFM consistently and scalably outperforms all attacks. On average, it achieves +5.9% (node) and +2.4% (graph) accuracy improvements over the runner-up, indicating the strong robustness. ❷ Compared to MDGFM, the strongest baseline with a robust domain adaptation module, SA²GFM achieves +5.1% average gain in the challenging cross-domain settings, confirming its superiority in mitigating negative transfer under the large domain shifts. ❸ Compared to other baselines, SA²GFM achieves an average improvement of +12.5%, verifying the impact of our structure-aware pretraining and null-expert routing, which together suppress spurious correlations and prevent negative transfer, contributing to stable and robust adaptation under various perturbation scenarios.

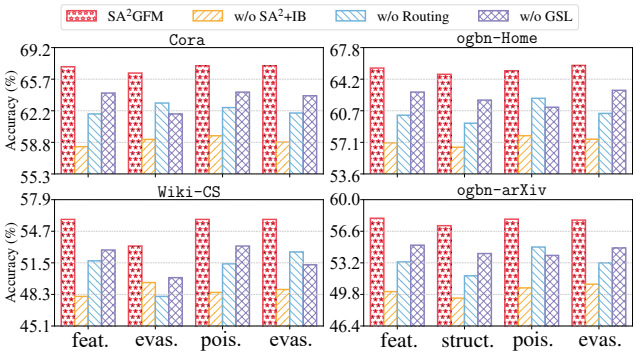

Figure 6: Ablation study on node classification under the non-targeted ($\lambda = 0.4$) and targeted ($p = 1$) attacks.

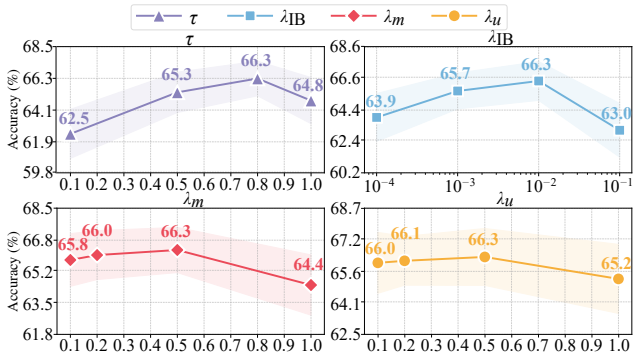

Figure 7: Hyperparameter sensitivity study on node classification under the targeted poisoning attack ($p = 1$).

5.3 *RQ2:* Analysis of Perturbation Degree

We evaluate performance degradation under progressively stronger perturbations. Results (Figure 4 and Figure 5) show that: ❶ The performance drop of SA^{2}GFM is significantly slower and more stable. ❷ This trend indicates that SA^{2}GFM achieves stronger base-level robustness while maintaining stable predictions under increasing perturbation levels. ❸ Notably, under the most severe settings (*e.g.*, $\lambda = 0.8$ or $p = 3$), SA^{2}GFM consistently outperforms the strong baselines MDGFM and GraphBridge by large margins in both node and graph classification.

5.4 *RQ3:* Ablation Study

We construct three variants to assess the key components.

- **SA^{2}GFM (*w/o* SA^2 + IB):** removes both the Structure-Aware Semantic-Augmentation and the self-supervised IB module described in Section 4.1.
- **SA^{2}GFM (*w/o* Routing):** removes the selective expert routing mechanism described in Section 4.2.
- **SA^{2}GFM (*w/o* GSL):** removes the hierarchical structure optimization module described in Section 4.3.

Results (Figure 6) show that all three components contribute to robustness with different impacts. ❶ Removing "SA^2+IB" causes the largest accuracy drop, highlighting its critical role in learning transferable and noise-resistant representations. ❷ Removing "Routing" degrades performance, particularly under targeted attacks. ❸ Removing "GSL" reduces accuracy, especially under structural perturbations. ❹ Overall, the full SA^{2}GFM achieves superior robustness through the coordinated contributions of all modules.

5.5 *RQ4:* Analysis of Hyperparameter Sensitivity

We analyze the sensitivity of SA^{2}GFM to four key hyperparameters: τ, λ_{IB}, λ_m, and λ_u. Results (Figure 7) show that SA^{2}GFM is stable across a wide range of values. ❶ All parameters exhibit smooth performance curves with optima near the default settings. ❷ In particular, $\lambda_{\text{IB}} = 10^{-2}$ and $\lambda_m = 0.5$ achieve the highest robustness under targeted poisoning attacks, while extreme values cause moderate degradation. ❸ Overall, SA^{2}GFM maintains reliable robustness without requiring extensive hyperparameter tuning.

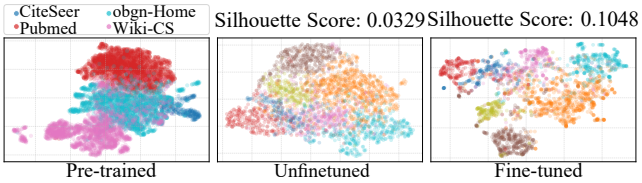

Figure 8: Node embeddings on `Cora` (cross-dataset).

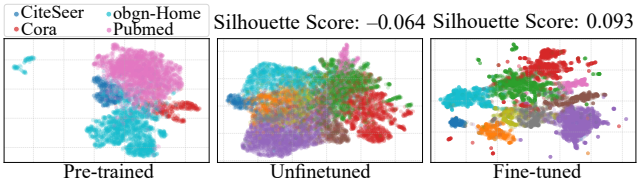

Figure 9: Node embeddings on `Wiki-CS` (cross-domain).

5.6 *RQ5:* Embedding Visualizations

We utilize the UMAP (McInnes et al. 2018) to visualize the node embeddings and employ the Silhouette Score to evaluate the quality of class-wise clustering (the larger, the better). As shown in Figure 8 and Figure 9, the pre-trained embeddings have captured transferable structural and semantic patterns, evidenced by the domain-level separation. However, class-wise separability remains weak without fine-tuning. After fine-tuning, the embeddings become more clustered along class boundaries, confirming task-specific alignment.

6 Conclusion

In this work, we propose a robust graph foundation model SA^{2}GFM. Through unified pre-training and fine-tuning, our method effectively bridges upstream transferable knowledge with downstream structural adaptation. Extensive experiments demonstrate that SA^{2}GFM achieves superior robustness against both the random noise and adversarial perturbations, as well as strong generalization across domains, and consistently yields gains from its key modules, offering a promising foundation for reliable and adaptable GFM.

Acknowledgments

The corresponding author is Qingyun Sun. This work is supported in part by the National Natural Science Foundation of China (NSFC) under grants No. 62427808 and No. 623B2010, by the Fundamental Research Funds for the Central Universities, and by the Basic Ability Enhancement Program for Young and Middle-aged Teachers of Guangxi (No. 2024KY0073). We extend our sincere thanks to all authors for their valuable contributions.

References

Bai, L.; Xu, Z.; Cui, L.; Li, M.; Wang, Y.; and Hancock, E. R. 2024. HC-GAE: The hierarchical cluster-based graph auto-encoder for graph representation learning. In *NeurIPS*.

Dan, J.; Liu, W.; Xie, C.; Yu, H.; Dong, S.; and Tan, Y. 2024. TFGDA: Exploring topology and feature alignment in semi-supervised graph domain adaptation through robust clustering. In *NeurIPS*, volume 37, 50230–50255.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2019. Combining neural networks with personalized pagerank for classification on graphs. In *ICLR*.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital libraries*, 89–98.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020a. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, volume 33, 22118–22133.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020b. Strategies for pre-training graph neural networks. In *ICLR*.

Huang, S.; Xu, Y.; Zhang, H.; and Li, X. 2025. Learn beneficial noise as graph augmentation. In *ICML*.

Ju, L.; Yang, X.; Li, Q.; and Wang, X. 2025. GraphBridge: Towards arbitrary transfer learning in GNNs. In *ICLR*.

Kataria, M.; Kumar, S.; and Jayadeva. 2024. UGC: Universal graph coarsening. In *NeurIPS*, 63057–63081.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Lee, W.; and Park, H. 2025. Self-supervised adversarial purification for graph neural networks. In *ICML*.

Li, A.; and Pan, Y. 2016. Structural Information and Dynamical Complexity of Networks. *IEEE Transactions on Information Theory*, 62(6).

Liu, J.; Yang, C.; Lu, Z.; Chen, J.; Li, Y.; Zhang, M.; Bai, T.; Fang, Y.; Sun, L.; Yu, P. S.; et al. 2023a. Towards graph foundation models: A survey and beyond. *CoRR*.

Liu, Z.; Yu, X.; Fang, Y.; and Zhang, X. 2023b. GraphPrompt: Unifying pre-training and downstream tasks for graph neural networks. In *WWW*, 417–428.

Long, Q.; Jin, Y.; Song, G.; Li, Y.; and Lin, W. 2020. Graph structural-topic neural network. In *KDD*, 1065–1073.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3: 127–163.

McInnes, L.; Healy, J.; Saul, N.; and Großberger, L. 2018. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29): 861.

Mernyei, P.; and Cangea, C. 2020. Wiki-CS: A Wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI*, volume 33, 4602–4609.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Stewart, G. W. 1993. On the early history of the singular value decomposition. *SIAM Review*, 35(4): 551–566.

Tishby, N.; Pereira, F. C.; and Bialek, W. 1999. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, 368–377.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep graph infomax. In *ICLR*.

Wang, Q.; Wang, Y.; Wang, Y.; and Ying, X. 2024. Dissecting the failure of invariant learning on graphs. In *NeurIPS*.

Wang, S.; Wang, B.; Shen, Z.; Deng, B.; and Kang, Z. 2025. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In *ICML*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *ICLR*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. In *NeurIPS*, 1–12.

Yu, X.; Zhou, C.; Fang, Y.; and Zhang, X. 2024. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*.

Yuan, H.; Sun, Q.; Fu, X.; Ji, C.; and Li, J. 2024. Dynamic Graph Information Bottleneck. In *WWW*, 469–480.

Yuan, H.; Sun, Q.; Fu, X.; Zhang, Z.; Ji, C.; Peng, H.; and Li, J. 2023. Environment-aware dynamic graph learning for out-of-distribution generalization. *NeurIPS*, 36.

Yuan, H.; Sun, Q.; Shi, J.; Fu, X.; Hooi, B.; Li, J.; and Yu, P. S. 2025a. GRAVER: Generative Graph Vocabularies for Robust Graph Foundation Models Fine-tuning. In *NeurIPS*.

Yuan, H.; Sun, Q.; Shi, J.; Fu, X.; Hooi, B.; Li, J.; and Yu, P. S. 2025b. How much can transfer? BRIDGE: Bounded multi-domain graph foundation model with generalization guarantees. In *ICML*.

Yuan, H.; Sun, Q.; Wang, Z.; Fu, X.; Ji, C.; Wang, Y.; Jin, B.; and Li, J. 2025c. DG-Mamba: Robust and efficient dynamic graph structure learning with selective state space models. In *AAAI*, volume 39, 22272–22280.

Zhang, G.; Dong, H.; Zhang, Y.; Li, Z.; Chen, D.; Wang, K.; Chen, T.; Liang, Y.; Cheng, D.; and Wang, K. 2024. GDeR: Safeguarding efficiency, balancing, and Robustness via Prototypical Graph Pruning. In *NeurIPS*.

Zhao, H.; Chen, A.; Sun, X.; Cheng, H.; and Li, J. 2024. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *KDD*, 4443–4454.

Zheng, S.; Wang, H.; and Liu, X. 2024. IntraMix: Intra-class mixup generation for accurate labels and neighbors. In *NeurIPS*, 8951–8980.

Zhu, Y.; Xu, W.; Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2021. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036*, 14: 1–1.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *KDD*, 2847–2856.