BRIDGING THE GAP TO UNSEEN GRAPHS: TARGET-SIDE ADAPTATION FOR FROZEN GRAPH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph foundation models (GFMs) are emerging as a key paradigm for cross-domain graph learning. However, most GFMs typically face two limitations during the adaptation phase: (i) Firstly, the adaptation method of GFMs is closely coupled with the internal architecture of the model, making it difficult for users to adapt published models to new target graphs in black-box scenarios. Furthermore, (ii) most existing methods rely on a limited number of labels to optimize parameters during the adaptation phase. When the target graph differs from the pre-training domain, limited supervision may not effectively guide adaptation, potentially leading to negative transfer. To address these limitations, this paper proposes a target-side adaptation framework for frozen GFMs to improve the practicality and effectiveness of GFMs. Specifically, we design a feature adapter and a structural adapter at the model's input to enable the frozen GFM to adapt to the features and structure of the target data, allowing users to perform adaptation without needing to understand the model's internal design. Subsequently, we employ a maximal coding rate reduction loss to exploit unlabeled target nodes and promote compact but diverse adapted features, while using a spectrum alignment loss to adjust the target graph such that its information spectrum matches the encoding preference of the frozen GFM. Additionally, we theoretically analyzed that our adaptation method achieves a lower generalization error bound. Comprehensive experiments demonstrate the effectiveness and generalization capability of this method on various downstream tasks. *The code is released at https://anonymous.4open.science/r/GFM-adapter.*

1 INTRODUCTION

Graph foundation models (GFMs) have emerged as a promising paradigm in graph representation learning (Liu et al., 2024; Zhao et al., 2024; Wang et al., 2025; Xia & Huang, 2026). Through pre-training across diverse graphs, GFMs aim to learn transferable and generalizable representations from node features and graph structures that can be reused across various domains and downstream tasks (Yu et al., 2025; Sun et al., 2025; Yuan et al., 2025a). Compared with graph neural networks (GNNs) trained for individual graphs, GFMs are expected to reduce repeated training costs and improve model reusability in citation, social, biological, and recommendation networks (Hu et al., 2020; Zhu et al., 2026).

To transfer pre-trained graph knowledge to an unseen target graph, adaptation methods use target-domain information to bridge the gap between the target graph and the pre-trained model (Wang et al., 2025; Chen et al., 2026). Depending on what is adapted, existing GNN-based GFM [1] can be broadly categorized into model-side adaptation and target-graph-side adaptation. Model-side adaptation methods adjust the internal parameters of GFMs to adapt to the target graph. For example, MTG (Chen et al., 2026) introduces learnable message prototypes into each message-passing layer of the frozen pre-trained encoder and dynamically combines them based on node representations to control message passing on the target graph. Target-graph-side adaptation modifies the input node features or graph structure to better suit the pre-trained model.For example, All in One (Sun et al.,

[1] Related work is presented in the appendix B.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

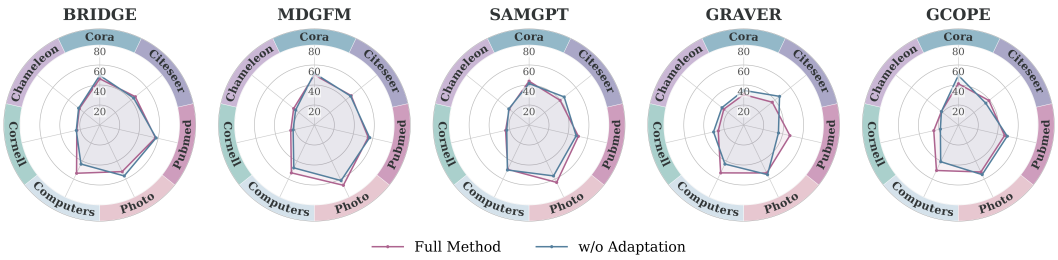

Figure 1: Comparison between full adaptation and no adaptation across seven target graphs.

2023) unifies different downstream tasks into prediction problems on a prompt graph by introducing learnable prompt nodes. Additionally, MDGFM (Wang et al., 2025) uses feature prompts and a structural learner to adjust target attributes and adjacency relationships, respectively, and feeds them into a frozen encoder.

Although existing methods have improved the transferability of GFMs to target graphs, their adaptation processes still face two key limitations. Firstly, GFM is usually a black box for users. Existing adaptation methods often rely on adjusting the model's internal architecture (Chen et al., 2026; Jiang et al., 2026), requiring a deep understanding of model-specific designs and complicating deployment. Second, existing methods often rely on limited target-domain labels to optimize adaptation parameters. When unseen target graphs differ significantly from pre-training graphs in feature distributions and structural patterns, limited supervision may not reliably guide adaptation, and negative transfer may occur (Hu et al., 2020; Zhu et al., 2026).As illustrated in Figure 1, the five GFM adaptation methods were unable to achieve stable positive transfer on all seven datasets.

Based on these observations, fully exploiting target-graph information under black-box conditions may improve the practicality and effectiveness of GFM adaptation. Achieving this requires addressing two key challenges. (i) To simplify adaptation for users, it is necessary to develop a unified adaptation method that is applicable to different frozen models without modifying their backbones. (ii) When the GFM is treated as a black box, it is difficult to convert target-graph information into reliable signals that simultaneously guide the adaptation of node features and graph structure.

In this paper, to address the above challenges, we propose a **T**arget-side **A**daptation framework for frozen **GFM**s,termed TAGFM, as shown in Figure 2. Specifically, we decouple the adaptation process from the internal architecture of a specific model. We design a feature adapter and a structural adapter outside the GFM backbone to adjust the target graph's node features and structure, respectively, making them more conducive to encoding by a specific GFM backbone. This design avoids inserting architecture-specific prompts or adaptation modules into the GFM backbone, allowing the same adaptation framework to support different pre-trained GFMs, thereby addressing **Challenge (i)**. In addition, we construct an adaptation signal from target-graph information and the frozen model's outputs. We optimize the target features and graph structure using a maximal coding rate reduction loss (Yu et al., 2020)and a spectrum alignment loss, respectively, to better align them with the encoding preferences of the GFM backbone and tackle **Challenge (ii)**. Finally, we theoretically analyze how optimization under the proposed adaptation framework can lower a generalization error upper bound.

Compared to existing works, our main contributions are summarized as follows:

- To the best of our knowledge, our proposed target-side adaptation framework is the first adaptation framework to support different GFM backbones without accessing the internal architecture of the model.
- We design a maximal coding rate reduction loss and spectrum alignment loss, optimizing the node features and graph structure generated by the feature and structure adapters, respectively, so that the adapted target graph better aligns with the encoding preferences of the frozen GFMs.
- We theoretically analyze the proposed method's generalization error bound and validate its effectiveness across multiple GFMs. Experimental results show improved target-graph performance and a reduced risk of negative transfer.

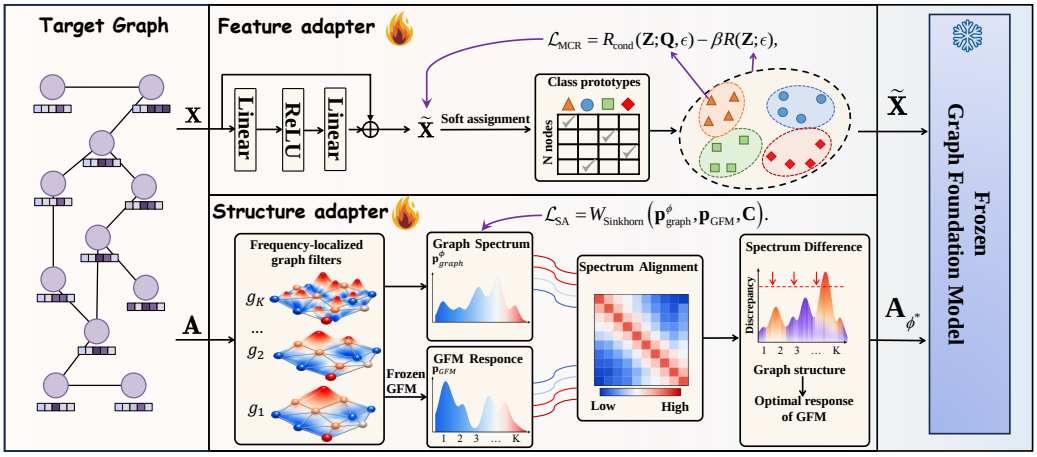

Figure 2: Overview of the proposed TAGFM framework. TAGFM first applies the feature adapter to transform the features and optimizes the adapted representations with the maximal coding rate reduction objective. It then adjusts the target graph structure by estimating the spectral preference of the frozen GFM and performing spectrum alignment. The adapted features and structure are finally fed into the frozen GFM for downstream tasks.

2 METHOD

2.1 NOTATIONS

Let $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A})$ denote a target graph, where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ and $\mathcal{E}$ represent the node set and edge set, respectively. $\mathbf{X} \in \mathbb{R}^{N\times D}$ denotes the node feature matrix, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix, where $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $A_{ij} = 0$ otherwise. Here, N and D denote the number of nodes and the dimension of node features, respectively. Let f_θ denote a pretrained GFM with parameters θ, and let $\mathbf{H} = f_\theta(\mathbf{X}, \mathbf{A}) \in \mathbb{R}^{N\times d}$ denote the node representations produced by the model, where d is the representation dimension. The target graph provides a support set $\mathcal{S} = \{(v_i, y_i)\}_{i\in\mathcal{I}_s}$, where $\mathcal{I}_s \subseteq \{1, \ldots, N\}$ and $y_i \in \{1, \ldots, C\}$ is the label of node v_i.

2.2 MOTIVATION

Published GFMs allow users to transfer pre-trained knowledge to new target graphs. However, existing adaptation methods often depend on model-specific architectures, requiring users to understand the model design. Target-side adaptation reduces this dependency by adjusting target-graph inputs outside the model, allowing the frozen model to process new target graphs in a black-box scenario without accessing or modifying its internal architecture.

Achieving the aforementioned target-side adaptation framework is difficult. From a feature perspective, a few labeled nodes can only provide limited class information, making it difficult to fully constrain the feature adaptation process. This leads to the first question: *How to exploit unlabeled information in the target graph to provide reliable constraints for feature adaptation?* From a structure perspective, the graph structure determines how information propagates through nodes, and different GFMs may exhibit distinct propagation characteristics. It is unreasonable to adjust the structure based on fixed assumptions (*e.g.,* homophily) without understanding the model's internal architecture, as this may further amplify the mismatch between the target graph and the model. Therefore, this leads to the second question: *How to mitigate the mismatch between the target graph structure and the propagation characteristics of a frozen GFM when the GFM architecture is unknown?*

However, few studies on GFM adaptation explore how target-graph information can guide feature and structural adjustments when the model's internal architecture is unknown. We therefore propose a target-side adaptation framework for frozen GFMs, addressing the two questions in Sections 2.3 and 2.4, respectively.

2.3 FEATURE ADAPTATION

To supplement few-shot supervision with unlabeled target-node information, we optimize a feature adapter outside the frozen backbone using maximal coding rate reduction (Yu et al., 2020), encouraging intra-class compactness and overall feature diversity. Given dimension-aligned features $\mathbf{X} \in \mathbb{R}^{N\times D}$, the adapted features $\widetilde{\mathbf{X}}$ are:

$$\Delta\mathbf{X} = \sigma(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \qquad \widetilde{\mathbf{X}} = \mathbf{X} + \gamma\Delta\mathbf{X}, \tag{1}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{D\times D}$ are learnable weight matrices, $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{1\times D}$ are bias vectors, $\sigma(\cdot)$ denotes an element-wise nonlinear activation,and γ controls the strength of the feature adjustment. To supplement limited supervision, we constrain the adapted features by comparing their overall and class-conditional coding rates.

To characterize the overall diversity of the adapted features, let $\mathbf{Z} \in \mathbb{R}^{N\times D}$ denote the adapted features $\widetilde{\mathbf{X}}$ after row-wise ℓ_2 normalization. The overall coding rate is defined as:

$$R(\mathbf{Z};\epsilon) = \frac{1}{2}\log\det\left(\mathbf{I}_d + \frac{d}{N\epsilon^2}\mathbf{Z}^\top\mathbf{Z}\right), \tag{2}$$

where $\mathbf{I}_d$ is the d-dimensional identity matrix and $\epsilon > 0$ controls the coding precision. Under the normalization constraint, increasing $R(\mathbf{Z};\epsilon)$ encourages diversity in the overall feature representation. In addition to overall representation diversity, within-class compactness is encouraged through a class-conditional coding rate. To obtain a class assignment for every target node, we first calculate class prototypes constructed from the labeled support set, *i.e.,* $\mathbf{p}_c = \mathrm{Norm}\left(\frac{1}{|\mathcal{S}_c|}\sum_{i\in\mathcal{S}_c}\mathbf{x}_i\right)$,

where $\mathcal{S}_c = \{i \in \mathcal{I}_s : y_i = c\}$, $\mathbf{x}_i$ denotes the feature vector of node i, and $\mathrm{Norm}(\cdot)$ denotes ℓ_2 normalization. For an unlabeled node i, its soft assignment to class c is computed as:

$$q_{ic} = \frac{\exp(\mathbf{x}_i^\top\mathbf{p}_c/\tau)}{\sum_{j=1}^{C}\exp(\mathbf{x}_i^\top\mathbf{p}_j/\tau)}, \tag{3}$$

where C is the number of classes and $\tau > 0$ is a temperature parameter. For labeled nodes, the assignments are set to one-hot vectors corresponding to their ground-truth labels. The resulting assignment matrix $\mathbf{Q} \in \mathbb{R}^{N\times C}$ is kept fixed during adaptation. Define $\mathbf{\Pi}_c = \mathrm{diag}(q_{1c},\ldots,q_{Nc})$ and $n_c = \sum_{i=1}^{N} q_{ic}$ as the class assignment weight matrix and the effective class size, respectively. The class-conditional coding rate is defined as follows:

$$R_{\mathrm{cond}}(\mathbf{Z};\mathbf{Q},\epsilon) = \sum_{c=1}^{C}\frac{n_c}{2N}\log\det\left(\mathbf{I}_d + \frac{d}{n_c\epsilon^2}\mathbf{Z}^\top\mathbf{\Pi}_c\mathbf{Z}\right). \tag{4}$$

The matrix $\mathbf{\Pi}_c$ weights node features according to their soft class assignments, allowing unlabeled nodes to contribute to the class-conditional coding rate. Reducing $R_{\mathrm{cond}}(\mathbf{Z};\mathbf{Q},\epsilon)$ encourages features assigned to the same class to occupy a more compact subspace. Finally, the coding rate reduction loss for feature adaptation is defined as

$$\mathcal{L}_{\mathrm{MCR}} = R_{\mathrm{cond}}(\mathbf{Z};\mathbf{Q},\epsilon) - \beta R(\mathbf{Z};\epsilon), \tag{5}$$

where $\beta > 0$ controls the trade-off between $R_{\mathrm{cond}}(\mathbf{Z};\mathbf{Q},\epsilon)$ and $R(\mathbf{Z};\epsilon)$. Minimizing $R_{\mathrm{cond}}(\mathbf{Z};\mathbf{Q},\epsilon)$ encourages nodes assigned to the same class to concentrate in a compact subspace. At the same time, the negative term $-\beta R(\mathbf{Z};\epsilon)$ encourages the overall representations to preserve diverse directions, thereby preventing the adapted features from collapsing into a limited subspace. Since the assignment matrix $\mathbf{Q}$ includes both labeled and unlabeled nodes, this objective extends the limited supervision from the support set to the feature distribution of the whole target graph, thus addressing the first question in Section 2.2.

2.4 STRUCTURE ADAPTATION

Although feature adaptation can adjust the node features, it cannot change the information propagation relationship among nodes, so there may still be a mismatch between the target graph structure

and the frozen GFM. From a spectrum perspective, GNN-based GFM can be seen as graph filters that propagate and transform graph signals (Bo et al., 2023), while the target graph structure determines the distribution of node information across different frequency bands. When the effective information in the target graph is concentrated in frequency bands that the model is not good at encoding (*e.g.,* when encoding heterophilic graphs via a low-pass graph filter), this information may be weakened during the model's encoding process.

Based on this analysis, aligning the target graph spectrum with the frequency bands preferred by the frozen GFM can mitigate structure mismatch and improve generalization. To this end, we first estimate the model's encoding preferences by analyzing the GFM backbone's spectral response.

Specifically, given a target graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A})$, its normalized graph Laplacian is defined as $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the degree matrix of the target graph. A set of K frequency-localized graph filters is constructed based on $\mathbf{L}$ to characterize information at different frequency bands. The response function of the k-th filter is defined as

$$g_k(\lambda) = \exp\left(-\frac{(\lambda - \mu_k)^2}{2\sigma^2}\right), \tag{6}$$

where λ denotes a graph Laplacian eigenvalue, μ_k denotes the center frequency of the k-th band, and σ controls the bandwidth. Let $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ denote the eigendecomposition of the graph Laplacian. The corresponding matrix filter is defined as $g_k(\mathbf{L}) = \mathbf{U}g_k(\mathbf{\Lambda})\mathbf{U}^\top$, where $g_k(\mathbf{\Lambda})$ applies the scalar function $g_k(\cdot)$ to each eigenvalue. The matrix filter $g_k(\mathbf{L})$ is then applied to a random graph signal $\mathbf{R}$ to extract its component in the k-th frequency band, *i.e.,* , $\mathbf{\Delta}_k = g_k(\mathbf{L})\mathbf{R}$. The structure adapter adds and subtracts this frequency-specific perturbation from the original node features, producing $\mathbf{X} + \eta\mathbf{\Delta}_k$ and $\mathbf{X} - \eta\mathbf{\Delta}_k$, respectively, where η controls the perturbation strength. The change in the model output induced by the k-th frequency perturbation is estimated as:

$$\mathbf{D}_k = \frac{f_\theta(\mathbf{X} + \eta\mathbf{\Delta}_k, \mathbf{A}) - f_\theta(\mathbf{X} - \eta\mathbf{\Delta}_k, \mathbf{A})}{2\eta}. \tag{7}$$

The finite-difference matrix $\mathbf{D}_k$ measures the change in the frozen GFM representation caused by perturbations in the k-th frequency band. We then use its Frobenius norm to quantify the corresponding frequency sensitivity and define the spectrum preference of the frozen GFM as:

$$p_{\mathrm{GFM}}(k) = \frac{\|\mathbf{D}_k\|_{\mathrm{F}}}{\sum_{j=1}^{K} \|\mathbf{D}_j\|_{\mathrm{F}}}. \tag{8}$$

Based on Eq. (8), each frequency band produces a sensitivity $p_{\mathrm{GFM}}(k)$. By collecting these sensitivities across all K bands, we obtain the spectrum preference vector $\mathbf{p}_{\mathrm{GFM}} = [p_{\mathrm{GFM}}(1), \ldots, p_{\mathrm{GFM}}(K)]^\top \in \mathbb{R}^K$. A larger $p_{\mathrm{GFM}}(k)$ indicates that the frozen GFM is more sensitive to information in the k-th frequency band.

Intuitively, low-frequency perturbations exhibit strong consistency between neighboring nodes, and their directions of change are typically quite similar. When such perturbation inputs are applied to GFMs with low-pass characteristics, neighborhood aggregation accumulates similar changes in neighboring nodes, thereby producing a more distinct output response. In contrast, high-frequency perturbations typically alternate between neighboring nodes, and their signals tend to counteract each other during the neighborhood aggregation process, resulting in a relatively weak response after propagation through the model. Therefore, changes in the output caused by perturbations in different frequency bands can reflect the encoding preferences of the frozen graph model for the corresponding frequency information, thereby enabling the estimation of the GFM backbone's filtering preferences without accessing the model's internal architecture.

Given the estimated filtering preference $\mathbf{p}_{\mathrm{GFM}}$ of the frozen GFM, the structure adapter optimizes learnable weights over the candidate undirected edges to align the target spectrum with this preference. These weights form a symmetric adjacency matrix $\mathbf{A}_\phi$, allowing the adapter to reweight existing edges and activate selected candidate connections. The corresponding normalized graph Laplacian is denoted by $\mathbf{L}_\phi$.

Specifically, to compute the spectrum distribution of the target information under the graph structure $\mathbf{A}_\phi$, the node feature matrix $\mathbf{X}$ is kept fixed during structure optimization and treated as a

graph signal defined on the nodes. Applying the k-th frequency filter $g_k(\mathbf{L}_\phi)$ to $\mathbf{X}$ extracts the target information contained in the corresponding frequency band, whose energy is measured by $\|g_k(\mathbf{L}_\phi)\mathbf{X}\|_{\mathrm{F}}^2$. We normalize the energies across all K frequency bands to obtain the spectrum distribution $p_{\mathrm{graph}} \in \mathbb{R}^K$ of the adapted graph $\mathbf{A}_\phi$:

$$p_{\mathrm{graph}}^{\phi}(k) = \frac{\|g_k(\mathbf{L}_\phi)\mathbf{X}\|_{\mathrm{F}}^2}{\sum_{j=1}^{K} \|g_j(\mathbf{L}_\phi)\mathbf{X}\|_{\mathrm{F}}^2}. \tag{9}$$

Changing $\mathbf{A}_\phi$ modifies the corresponding Laplacian $\mathbf{L}_\phi$ and redistributes the node-feature energy across frequency bands. The structure adapter therefore aims to align the spectral distribution $\mathbf{p}_{\mathrm{graph}}^{\phi}$ with the spectrum preference $\mathbf{p}_{\mathrm{GFM}}$ of the frozen GFM.

Since the frequency bands are ordered by their center frequencies, their distances should be considered during spectrum alignment. A direct element-wise comparison treats all frequency mismatches equally and cannot distinguish shifts between nearby and distant bands. We therefore formulate spectral alignment as an optimal transport problem and define the transport cost between the i-th and j-th frequency bands as:

$$C_{ij} = (\mu_i - \mu_j)^2, \tag{10}$$

where μ_i and μ_j denote the center frequencies of the two bands. The cost matrix $\mathbf{C}$ assigns lower costs to nearby frequency bands and higher costs to distant bands. Based on this cost matrix, the spectrum alignment loss is defined as the entropy-regularized optimal transport distance:

$$\mathcal{L}_{\mathrm{SA}} = W_{\mathrm{Sinkhorn}}\left(\mathbf{p}_{\mathrm{graph}}^{\phi}, \mathbf{p}_{\mathrm{GFM}}, \mathbf{C}\right). \tag{11}$$

Minimizing $\mathcal{L}_{\mathrm{SA}}$ encourages the spectrum distribution of the target graph structure to approach the filtering preference of the frozen GFM while considering the distances between frequency bands. Since $\mathbf{p}_{\mathrm{graph}}^{\phi}$ depends on the Laplacian $\mathbf{L}_\phi$, which is determined by the learnable adjacency matrix $\mathbf{A}_\phi$, this loss provides a gradient signal for optimizing the candidate edge weights. This procedure does not require access to the internal architecture of the GFM and provides a model-independent way to convert target graph information into a structural adaptation signal, thus addressing the second question in Section 2.2.

2.5 Objective Function and Generalization Error Bound Analysis

Sections 2.3 and 2.4 describe node-feature and graph-structure adaptation, respectively.Specifically, the structural adapter first adjusts the target graph structure based on the spectrum alignment loss calculated by Eq. (11), *i.e.,*

$$\phi^* = \arg\min_{\phi} \mathcal{L}_{\mathrm{SA}}\left(\mathbf{p}_{\mathrm{graph}}^{\phi}, \mathbf{p}_{\mathrm{GFM}}, \mathbf{C}\right). \tag{12}$$

Based on the optimized parameter ϕ^*, we obtain the adapted graph structure $\mathbf{A}_{\phi^*}$. Then, we calculate the adapted node feature $\widetilde{\mathbf{X}}$ using Eq. (1) and input $\mathbf{A}_{\phi^*}$ and $\widetilde{\mathbf{X}}$ into the frozen GFM to obtain the final node representation:

$$\mathbf{H} = f_\theta\left(\widetilde{\mathbf{X}}, \mathbf{A}_{\phi^*}\right). \tag{13}$$

where θ denotes the parameters of the frozen GFM backbone. Given the adapted graph structure $\mathbf{A}_{\phi^*}$, the feature adapter is optimized using the few-shot supervised loss on the support set $\mathcal{S}$ and the coding-rate-reduction loss:

$$\mathcal{J}_{\mathrm{F}} = \mathcal{L}_{\mathrm{sup}}\left(\mathbf{H}; \mathcal{S}\right) + \omega \mathcal{L}_{\mathrm{MCR}}\left(\widetilde{\mathbf{X}}; \mathbf{Q}\right), \tag{14}$$

where ω is a non-negative parameter achieving the trade-off between $\mathcal{L}_{\mathrm{sup}}\left(\mathbf{H}; \mathcal{S}\right)$ and $\mathcal{L}_{\mathrm{MCR}}\left(\widetilde{\mathbf{X}}; \mathbf{Q}\right)$. Pseudocode for TAGFM is provided in Appendix A.

In addition, we analyze the generalization performance of TAGFM, with detailed proofs provided in Appendix C. Let $\mathcal{S}$ and $\mathcal{U}$ denote the support and query sets, with $|\mathcal{S}| = m$ and $|\mathcal{U}| = u$, respectively. Let h_{ϕ,ϕ_F} denote the predictor parameterized by the structure-adapter parameter ϕ and the feature-adapter parameter ϕ_F. Let ϕ^* and ϕ_F^* denote their optimized values. Under Assumptions 1–3, the following theorem gives a transductive generalization bound for h_{ϕ^*,ϕ_F^*}.

Theorem 2.1 (PAC Generalization Bound for TAGFM). *With probability at least $1 - \delta_{\text{lab}} - \delta_{\text{probe}}$ over the support-set sampling and black-box spectral probing randomness, the bound holds:*

$$\mathcal{R}_{\mathcal{U}}\left(h_{\phi^*,\phi_F^*}\right) \leq \underbrace{\mathcal{R}_{\mathcal{S}}\left(h_{\phi^*,\phi_F^*}\right) + 2\varepsilon_{\text{cov}} + \frac{N}{u}\sqrt{\frac{p_{\phi_F}\log\left(1 + \frac{2RL_\ell L_{\phi_F}}{\varepsilon_{\text{cov}}}\right) + \log\left(\frac{2}{\delta_{\text{lab}}}\right)}{2m}}}_{\text{Feature Adapter Branch}} + \underbrace{2L_{\text{sp}}\left[\sqrt{\mathcal{L}_{\text{SA}}\left(\phi^*\right)} + \varepsilon_{\text{OT}} + \varepsilon_{\text{probe}}\right] + 2\varepsilon_{\text{sp}}}_{\text{Structure Adapter Branch}}. \tag{15}$$

Here, $\mathcal{R}_{\mathcal{S}}$ and $\mathcal{R}_{\mathcal{U}}$ denote the risks on the support and query sets, respectively. p_{ϕ_F} represents the number of parameters in the feature adapter, and L_{ϕ_F} denotes the corresponding Lipschitz constant. The remaining ε terms collect the approximation, probing, and spectral-stability errors introduced by the structural adaptation process.

Theorem 2.1 indicates that TAGFM can tighten the generalization bound through its target-side adaptation objectives. The coding-rate-reduction loss exploits unlabeled target nodes to constrain the adapted feature distribution without increasing the feature-adapter complexity term in the bound. More importantly, under the spectral stability and calibration assumptions, the spectral alignment loss directly controls the structural transfer term through $\sqrt{\mathcal{L}_{\text{SA}}(\phi^*)}$. Therefore, reducing the spectral mismatch between the adapted graph and the frozen GFM's preference tightens the bound while the probing and approximation errors remain bounded. These results provide a theoretical explanation for how TAGFM can improve generalization under limited target-domain supervision.

3 Experiment

3.1 Experimental Setup

In this section, we conduct experiments on seven public graph datasets to evaluate the effectiveness of the proposed TAGFM for few-shot cross-domain node classification and graph classification. We evaluate the method on five GFM backbones and compare it with the corresponding original adaptation method and GFMate adaptation method (Jiang et al., 2026).

3.1.1 Datasets

The used datasets include three citation networks (*i.e.,* , Cora, Citeseer, and Pubmed) (Yang et al., 2016), two Amazon co-purchase networks (*i.e.,* , Photo and Computers) (Shchur et al., 2018), and two heterophilous graph datasets (*i.e.,* , Cornell and Chameleon) (Pei et al., 2020).

3.1.2 Comparison Methods

The comparison methods include two traditional semi-supervised methods (*i.e.,* , GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018)), four graph self-supervised or pre-training methods (*i.e.,* , DGI (Veličković et al., 2019), GraphCL (You et al., 2020), DSSL (Xiao et al., 2022), and GraphACL (Xiao et al., 2023)), and five GFMs (*i.e.,* , GCOPE (Zhao et al., 2024), SAMGPT (Yu et al., 2025), BRIDGE (Yuan et al., 2025a), MDGFM (Wang et al., 2025), and GRAVER (Yuan et al., 2025b)). For each GFM backbone, three adaptation settings are considered: the original adaptation method associated with the backbone, GFMate (Jiang et al., 2026), and our proposed adaptation method, *i.e.,* TAGFM.

Table 1: One-shot node classification accuracy (mean ± sample SD, %) over 50 splits. Source lists backbone pretraining datasets. GFMate (blue) and TAGFM (red) values show mean gains in percentage points over each backbone's original adaptation across seven target datasets.

Source	CS PM Ph	CR PM Ph	CR CS Ph	CR CS PM	CR CS PM	CR CS PM	CR CS PM
	Com Co Ch	Com Co Ch	Com Co Ch	Com Co Ch	Ph Co Ch	Ph Com Ch	Ph Com Co
Model / Target	CR	CS	PM	Ph	Com	Co	Ch
GCN	38.63 ± 7.7	43.36 ± 6.2	54.63 ± 6.1	58.51 ± 6.6	42.86 ± 7.1	21.12 ± 5.9	28.42 ± 4.2
GAT	39.78 ± 7.5	42.94 ± 7.0	55.38 ± 8.1	55.14 ± 7.8	43.18 ± 8.6	18.55 ± 8.4	26.89 ± 4.1
DGI	38.79 ± 6.3	38.25 ± 5.1	42.94 ± 6.1	61.17 ± 5.2	47.56 ± 5.4	27.38 ± 6.0	24.60 ± 3.5
GraphCL	38.55 ± 6.1	38.28 ± 4.5	43.46 ± 6.5	63.95 ± 5.2	48.51 ± 5.5	27.24 ± 6.5	24.87 ± 3.2
DSSL	29.09 ± 6.8	29.76 ± 5.4	36.41 ± 6.7	62.87 ± 6.7	50.93 ± 6.9	21.58 ± 7.1	21.77 ± 4.1
GraphACL	50.10 ± 7.0	42.19 ± 5.9	54.85 ± 7.2	64.22 ± 6.4	50.48 ± 6.7	25.27 ± 5.6	28.93 ± 3.9
GCOPE	41.31 ± 8.1	39.23 ± 6.5	47.52 ± 6.0	52.12 ± 8.1	50.46 ± 7.6	25.24 ± 11.7	21.71 ± 3.2
+ GFMate +4.76	50.24 ± 7.5	41.55 ± 4.8	54.06 ± 5.6	56.48 ± 6.9	41.61 ± 9.3	**37.18 ± 7.7**	**29.77 ± 4.6**
+ TAGFM +7.53	**57.94 ± 8.2**	**41.94 ± 9.5**	**62.48 ± 12.2**	**65.26 ± 7.2**	**56.39 ± 9.0**	21.31 ± 8.1	25.00 ± 3.7
SAMGPT	43.81 ± 7.8	39.68 ± 5.8	50.68 ± 5.6	63.68 ± 7.1	49.25 ± 6.8	24.04 ± 6.5	25.47 ± 4.3
+ GFMate +3.75	44.48 ± 6.7	47.32 ± 5.5	55.65 ± 4.6	60.67 ± 7.0	49.01 ± 7.2	**36.78 ± 6.0**	28.95 ± 4.7
+ TAGFM +9.32	**60.31 ± 7.8**	**50.42 ± 7.7**	**63.09 ± 12.1**	**71.10 ± 6.0**	**59.36 ± 8.9**	27.76 ± 7.3	**29.82 ± 3.4**
BRIDGE	46.24 ± 6.6	45.48 ± 6.7	57.21 ± 7.0	51.71 ± 11.3	53.52 ± 7.8	23.57 ± 6.4	26.15 ± 4.7
+ GFMate +2.48	48.07 ± 5.9	45.22 ± 5.7	59.58 ± 5.6	57.50 ± 6.5	43.12 ± 5.6	**39.01 ± 7.4**	28.73 ± 4.6
+ TAGFM +10.60	**64.34 ± 6.9**	**52.54 ± 7.4**	**65.30 ± 9.9**	**73.17 ± 5.4**	**59.95 ± 9.2**	32.03 ± 6.3	**30.76 ± 5.0**
MDGFM	50.14 ± 7.6	47.15 ± 7.1	55.29 ± 7.7	66.88 ± 8.7	53.08 ± 7.6	24.28 ± 8.0	26.55 ± 4.4
+ GFMate +1.35	51.53 ± 6.9	47.36 ± 6.8	57.85 ± 6.4	61.94 ± 6.9	48.02 ± 6.2	**36.82 ± 9.0**	29.32 ± 4.6
+ TAGFM +7.21	**64.18 ± 7.4**	**54.79 ± 7.4**	**63.64 ± 10.3**	**73.66 ± 6.6**	**61.46 ± 9.4**	26.06 ± 4.5	**30.06 ± 4.5**
GRAVER	30.44 ± 11.7	36.59 ± 13.9	47.46 ± 8.1	53.24 ± 9.0	53.13 ± 9.8	26.17 ± 9.7	25.43 ± 4.0
+ GFMate +4.63	38.17 ± 5.5	47.48 ± 6.2	41.06 ± 6.0	62.48 ± 5.3	46.84 ± 5.0	**38.45 ± 8.5**	**30.40 ± 4.6**
+ TAGFM +11.00	**58.69 ± 8.8**	**55.89 ± 6.4**	**54.23 ± 10.7**	**70.22 ± 5.2**	**60.25 ± 8.9**	22.42 ± 8.9	27.74 ± 4.9

3.2 Effectiveness Analysis

3.2.1 Node classification

To evaluate TAGFM for few-shot cross-domain node classification, we conduct 1-shot and 5-shot experiments on seven target graph datasets. The 1-shot results are reported in Table 1, while the 5-shot results are provided in Table 2 of Appendix D.1. All experiments use 50 random splits and report the mean accuracy and sample standard deviation.

As shown in Table 1, TAGFM achieves positive average improvements over the original adaptation strategy for all five backbones, with gains of 7.53%, 9.32%, 10.60%, 7.21%, and 11.00% for GCOPE, SAMGPT, BRIDGE, MDGFM, and GRAVER, respectively. TAGFM also outperforms GFMate on average for most backbones, demonstrating that jointly adapting target-side node features and graph structure provides effective transfer signals across different GFMs and target graphs.

3.2.2 Graph classification

To evaluate the transferability of TAGFM beyond node-level prediction, we also conduct 1-shot and 5-shot graph classification experiments on the same seven target datasets. The detailed results and analysis are provided in Appendix D.1.

3.3 Generalization Ability Analysis

To further evaluate the generalization behavior of different adaptation methods during optimization, we record their test error curves on Photo and Computers across five GFM backbones, as shown in Figure 3. TAGFM consistently achieves the lowest test error in all ten backbone–dataset combinations. For example, after adaptation, TAGFM reduces the test error to approximately 0.29, 0.27, and

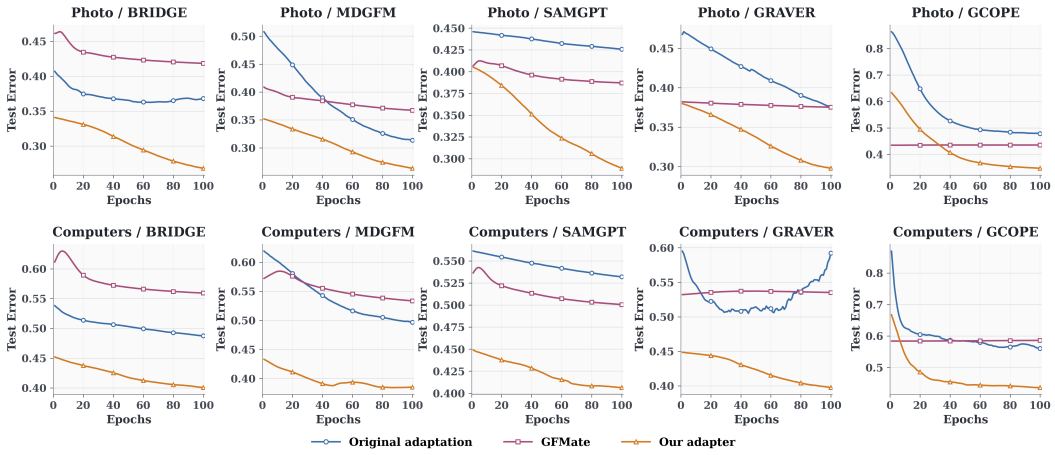

Figure 3: Test error curves on Photo and Computers across five GFM backbones

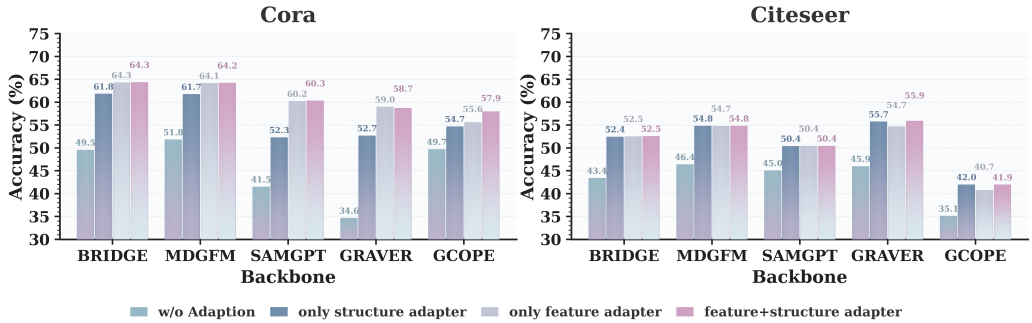

Figure 4: Ablation results on Cora and Citeseer under the 1-shot setting.

0.30 on Photo with BRIDGE, MDGFM, and GRAVER, respectively, while the competing methods remain at noticeably higher error levels.

TAGFM also exhibits a more stable decreasing trend throughout the adaptation process. In contrast, the original adaptation strategy shows clear fluctuations or even an increase in test error, as observed for GRAVER on Computers, while GFMate often converges to a higher error plateau. The consistently lower and more stable test error indicates that TAGFM is less sensitive to target-domain shifts and generalizes more reliably across different target graphs and GFM backbones.

3.4 Ablation Study

To evaluate the contributions of the two adapters, we conduct 1-shot ablation experiments on Cora and Citeseer, as shown in Figure 4. We compare four settings: without adaptation, using only the structure adapter, using only the feature adapter, and using both adapters. Results on the remaining datasets are provided in Appendix D.1. Both adapters substantially improve over the non-adapted setting. On Cora, the joint model achieves the highest average accuracy of 61.08%, compared with 45.42% without adaptation. On Citeseer, the structure adapter provides a stronger individual gain, while the joint model remains comparable or superior for most backbones. These results confirm the complementary contributions of the two adapters.

4 Conclusion

In this paper, we propose a target-side adaptation framework for frozen GFM, *i.e.,* TAGFM. TAGFM operates through the model input interface and jointly adapts target node features and graph structure without modifying the backbone. Maximum coding rate reduction improves the separability of adapted node representations, while spectrum alignment reduces the mismatch between the target graph and the model's propagation preference. Experiments across multiple backbones and target graphs demonstrate the effectiveness and generalization ability of TAGFM under few-shot adaptation settings.

AI USE STATEMENT

We used OpenAI's GPT solely to polish the language of the manuscript, including correcting grammar, refining wording, and improving clarity and readability. We did not use generative AI for any tasks requiring disclosure under the ICLR 2027 AI policy, including translation, theoretical development, formulation of mathematical claims or hypotheses, proof construction or writing, methodology or experimental design, method implementation, dataset generation or processing, or analysis and interpretation of results. The authors manually reviewed all AI-assisted edits and revised them as needed to preserve the intended scientific meaning and ensure consistency with the methods and results reported in the paper. We take full responsibility for the final content of this work, including all AI-assisted text, scientific claims, experimental results, and code.

ETHICS STATEMENT

This work uses publicly available benchmark graph datasets, including Cora, Citeseer, PubMed, Photo, Computers, Cornell, and Chameleon. The datasets and their sources are cited in the paper, and they are used in accordance with their applicable terms and licenses. This study did not involve experiments with human participants or animals. The datasets are used only as standard research benchmarks, and we do not attempt to re-identify individuals or disclose personal information.The experiments are conducted for methodological evaluation of graph foundation model adaptation. We do not identify any specific high-risk or harmful application arising directly from this work. Nevertheless, the proposed method may inherit biases or limitations present in the underlying datasets and pretrained models. We report the method, experimental settings, and limitations as transparently as possible, and the authors take responsibility for the integrity and ethical conduct of this research.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the feature and structure adaptation components of TAGFM in Sections 2.3 and 2.4, and provide the adaptation algorithm in Appendix A. The assumptions and proofs supporting the generalization bound are presented in Appendix C. Our experiments use publicly available benchmark graph datasets. We report classification accuracy as the mean and sample standard deviation over 50 random splits, with additional classification results, ablation studies, and parameter sensitivity analyses provided in Appendix D. The source code is available in the anonymous repository linked in the abstract to facilitate reproduction of the reported results.

REFERENCES

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *International Conference on Learning Representations*, 2023.

Qin Chen, Liang Wang, Bo Zheng, and Guojie Song. DAGPrompT: Pushing the Limits of Graph Prompting with a Distribution-aware Graph Prompt Tuning Approach. In *Proceedings of the ACM on Web Conference 2025*, pp. 4346–4358. Association for Computing Machinery, 2025. doi: 10.1145/3696410.3714917.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large Language and Graph Assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7809–7823. PMLR, 2024.

Yancheng Chen, Dun Ma, Shuai Zhang, Yang Liu, Xixun Lin, Xiangyu Zhao, Wenguo Yang, Wei Chen, and Chuan Zhou. Message tuning outshines graph prompt tuning: A prismatic space perspective. In *Forty-third International Conference on Machine Learning*, 2026.

Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal Prompt Tuning for Graph Neural Networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52464–52489, 2023.

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 594–604. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539321.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. GraphMAE2: A Decoding-Enhanced Masked Self-Supervised Graph Learner. In *Proceedings of the ACM Web Conference 2023*, pp. 737–746. Association for Computing Machinery, 2023. doi: 10.1145/3543507.3583379.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*, 2020.

Yan Jiang, Ruihong Qiu, and Zi Huang. GFMate: Empowering Graph Foundation Models with Test-time Prompt Tuning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph Structure Learning for Robust Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 66–74. Association for Computing Machinery, 2020. doi: 10.1145/3394486.3403049.

Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering Graph Representation Learning with Test-Time Graph Transformation. In *International Conference on Learning Representations*, 2023.

Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. GraphPatcher: Mitigating Degree Bias for Graph Neural Networks via Test-time Augmentation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55785–55801, 2023.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One For All: Towards Training One Graph Model For All Classification Tasks. In *International Conference on Learning Representations*, 2024.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2023*, pp. 417–428. Association for Computing Machinery, 2023. doi: 10.1145/3543507.3583386.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1150–1160. Association for Computing Machinery, 2020. doi: 10.1145/3394486.3403168.

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

Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip Yu. RiemannGFM: Learning a Graph Foundation Model from Riemannian Geometry. In *Proceedings of the ACM on Web Conference 2025*, pp. 1154–1165. Association for Computing Machinery, 2025. doi: 10.1145/3696410.3714952.

Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1717–1727. Association for Computing Machinery, 2022. doi: 10.1145/3534678.3539249.

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in One: Multi-Task Prompting for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2120–2131. Association for Computing Machinery, 2023. doi: 10.1145/3580305.3599256.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep Graph Infomax. In *International Conference on Learning Representations*, 2019.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-Domain Graph Foundation Models: Robust Knowledge Transfer via Topology Alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64806–64821. PMLR, 2025.

Lianghao Xia and Chao Huang. AnyGraph: Graph Foundation Model in the Wild. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 882–896. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.44.

Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for graphs. In *Advances in Neural Information Processing Systems*, volume 35, pp. 620–634, 2022.

Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *Advances in neural information processing systems*, volume 36, pp. 16129–16152, 2023.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pp. 40–48. PMLR, 2016.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in neural information processing systems*, volume 33, pp. 5812–5823, 2020.

Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. SAMGPT: Text-free Graph Foundation Model for Multi-domain Pre-training and Cross-domain Adaptation. In *Proceedings of the ACM on Web Conference 2025*, pp. 1142–1153. Association for Computing Machinery, 2025. doi: 10.1145/3696410.3714828.

Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning Diverse and Discriminative Representations via the Principle of Maximal Coding Rate Reduction. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9422–9434, 2020.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S. Yu. How Much Can Transfer? BRIDGE: Bounded Multi-Domain Graph Foundation Model with Generalization Guarantees. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73604–73644. PMLR, 2025a.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S. Yu. GRAVER: Generative graph vocabularies for robust graph foundation models fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-0560.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. 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*, pp. 4443–4454. Association for Computing Machinery, 2024. doi: 10.1145/3637528.3671913.

Jiajun Zhu, Ying Chen, Peihao Wang, Yixuan He, Pan Li, Aditya Akella, and Zhangyang Wang. When Do Graph Foundation Models Transfer? A Data-Centric Theory. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. ProG: A Graph Prompt Learning Benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95406–95437, 2024.

A Adaptation Algorithm

Algorithm 1: Pseudo-code of TAGFM

Input: Node features X, adjacency A;
Support set $\mathcal{S}$ and query node set $\mathcal{U}$;
Pre-trained GFM with fixed parameters θ;
Loss weight ω, adaptation budgets T_S, T_F
Output: Final predictions $\widehat{y}_{\mathcal{U}}$ for query nodes

% Stage 1: Semantic Guidance;
Initialize class prototypes using support features;
Construct fixed class assignments Q (Eq. equation 3);

% Stage 2: Model-Guided Structure Adaptation;
Estimate model spectrum preference p_{GFM} by frequency-band probing (Eqs. equation 6–equation 8);
Initialize structural parameters ϕ;
for $t \leftarrow 1$ **to** T_S **do**
  Compute target information spectrum $p^{\phi}_{\mathrm{graph}}$ (Eq. equation 9);
  Calculate spectrum alignment loss (Eqs. equation 10 and equation 11):
    $\mathcal{L}_{\mathrm{SA}} \leftarrow W_{\mathrm{Sinkhorn}}(p^{\phi}_{\mathrm{graph}}, p_{\mathrm{GFM}}, C)$;
  Update ϕ to minimize $\mathcal{L}_{\mathrm{SA}}$ (Eq. equation 12);
Obtain the adapted structure $\widetilde{A} \leftarrow A_{\phi^*}$;

% Stage 3: Feature Adaptation via Coding-Rate Reduction;
Initialize feature adapter parameters ϕ_F;
for $t \leftarrow 1$ **to** T_F **do**
  Adapt node features (Eq. equation 1):
    $\widetilde{X} \leftarrow X + \gamma\,\mathrm{MLP}_{\phi_F}(X)$;
  Encode the adapted graph: $H \leftarrow f_\theta(\widetilde{X}, \widetilde{A})$ (Eq. equation 13);
  Compute support classification loss $\mathcal{L}_{\mathrm{sup}}(H; \mathcal{S})$;
  Compute coding-rate reduction loss $\mathcal{L}_{\mathrm{MCR}}(\widetilde{X}; Q)$ (Eqs. equation 2, equation 4, and equation 5);
  Calculate the feature adaptation objective (Eq. equation 14):
    $\mathcal{J}_F \leftarrow \mathcal{L}_{\mathrm{sup}} + \omega\mathcal{L}_{\mathrm{MCR}}$;
  Update ϕ_F to minimize $\mathcal{J}_F$, keeping θ frozen;

% Stage 4: Target Prediction;
Obtain final adapted features using ϕ^*_F (Eq. equation 1);
Encode the adapted graph (Eq. equation 13) and predict $\widehat{y}_{\mathcal{U}}$;
return $\widehat{y}_{\mathcal{U}}$

B Related Work

This section briefly reviews topics related to this work, including Graph Foundation Models in Section B.1, and Graph Prompt Tuning Methods in Section B.2, and Adapters and Test-Time Graph Adaptation in Section B.3.

B.1 Graph Foundation Models

Graph pretraining learns reusable representations through mutual-information maximization and contrastive learning (Veličković et al., 2019; You et al., 2020; Qiu et al., 2020), or masked reconstruction (Hou et al., 2022; 2023). Transfer-oriented pretraining studies also identify negative transfer when pretraining does not support downstream tasks (Hu et al., 2020). Cross-domain GFMs address heterogeneity through different interfaces: OFA and LLaGA use language representations to connect attributes (Liu et al., 2024; Chen et al., 2024), while text-free models use coordinator nodes, topology alignment, or Riemannian geometry (Zhao et al., 2024; Wang et al., 2025; Sun et al., 2025). These designs improve knowledge sharing across domains. Nevertheless, target feature distributions and structural patterns can differ from those encountered during pretraining. Learning a shared backbone therefore leaves a downstream question: how to adapt an unseen graph to a fixed checkpoint without relying on scarce labels alone.

B.2 Graph Prompt Tuning Methods

GPPT and GraphPrompt align pretraining with downstream prediction through task reformulation (Sun et al., 2022; Liu et al., 2023). All in One introduces graph prompts across tasks, while GPF supports different pretraining strategies through input-feature prompts (Sun et al., 2023; Fang et al., 2023). DAGPrompT addresses distribution differences, and ProG standardizes the evaluation of prompt methods (Chen et al., 2025; Zi et al., 2024). For cross-domain transfer, SAMGPT combines structure tokens with dual prompts, whereas BRIDGE couples domain alignment with mixture-of-experts prompting (Yu et al., 2025; Yuan et al., 2025a). These cross-domain mechanisms use components developed during their respective pretraining procedures, making them less directly reusable with arbitrary published checkpoints. Input-level prompting offers a more portable interface. Portability alone, however, does not specify how limited target supervision should guide adaptation under simultaneous feature and structural shifts.

B.3 Adapters and Test-Time Graph Adaptation

Adapters and LoRA preserve pretrained parameters but require integration with internal layers or weight matrices (Houlsby et al., 2019; Hu et al., 2022). Input-side graph methods offer another route: GDC reshapes connectivity through diffusion, while Pro-GNN learns graph structure alongside the predictor (Gasteiger et al., 2019; Jin et al., 2020). At test time, GTrans adapts features and structure through self-supervision, and GraphPatcher augments neighborhoods to reduce degree bias (Jin et al., 2023; Ju et al., 2023). These methods already exploit information beyond target labels, but use different adaptation criteria and address different graph conditions. For frozen GFMs, the remaining design question is how to use target data and model responses to guide both feature and structural adaptation through a common input interface. Maximal coding rate reduction provides an information-theoretic criterion for representation learning (Yu et al., 2020). TAGFM combines this feature regularization with backbone-informed spectral alignment to supplement few-shot supervision without inserting architecture-specific modules into the backbone.

C Proof of the Generalization Bound

This section provides the assumptions and proofs for the generalization bound stated in Theorem 2.1. We consider a fixed target graph with node set $\mathcal{V}$, where $|\mathcal{V}| = N$. The support set $\mathcal{S}$ contains m labeled nodes, and the query set is defined as $\mathcal{U} = \mathcal{V} \setminus \mathcal{S}$ with $|\mathcal{U}| = u = N - m$.

Let ϕ denote the parameters of the structure adapter and let ϕ_F denote the parameters of the feature adapter. The frozen GFM is parameterized by θ. We use h_{ϕ,ϕ_F} to denote the predictor obtained by applying the two adapters before the frozen GFM. For a node subset $\mathcal{A} \subseteq \mathcal{V}$, let $\mathcal{R}_{\mathcal{A}}(h_{\phi,\phi_F})$ denote the average prediction risk on $\mathcal{A}$.

C.1 Assumptions

Assumption 1 (Bounded feature-adapter parameterization)**.** *The parameter space of the feature adapter is a compact set $\Phi_F \subseteq [-R, R]^{p_{\phi_F}}$. Moreover, there exists a constant $L_{\phi_F} > 0$ such that, for every admissible structure parameter ϕ and every $\phi_F, \phi'_F \in \Phi_F$,*

$$\max_{i \in \mathcal{V}} \left\| h_{\phi,\phi_F}(i) - h_{\phi,\phi'_F}(i) \right\|_\infty \leq L_{\phi_F} \left\| \phi_F - \phi'_F \right\|_\infty . \tag{16}$$

Assumption 2 (Spectral stability)**.** *There exists an ideal reference structure parameter $\phi^\dagger$, a spectral discrepancy d_{sp}, and constants $L_{\mathrm{sp}} \geq 0$ and $\varepsilon_{\mathrm{sp}} \geq 0$ such that, for every admissible ϕ, every $\phi_F \in \Phi_F$, and every node subset $\mathcal{A} \subseteq \mathcal{V}$,*

$$\left| \mathcal{R}_{\mathcal{A}} \left(h_{\phi,\phi_F} \right) - \mathcal{R}_{\mathcal{A}} \left(h_{\phi^\dagger,\phi_F} \right) \right| \leq L_{\mathrm{sp}} \, d_{\mathrm{sp}} \left(\mathbf{p}^{\phi}_{\mathrm{graph}}, \mathbf{p}_{F_0} \right) + \varepsilon_{\mathrm{sp}}, \tag{17}$$

where $\mathbf{p}_{F_0}$ denotes the ideal spectrum preference of the frozen GFM.

Assumption 3 (Spectral probing and transport calibration)**.** *With probability at least $1 - \delta_{\mathrm{probe}}$ over the randomness of black-box spectral probing,*

$$d_{\mathrm{sp}} \left(\mathbf{p}_{\mathrm{GFM}}, \mathbf{p}_{F_0} \right) \leq \varepsilon_{\mathrm{probe}}. \tag{18}$$

Moreover, for every admissible structure parameter ϕ,

$$d_{\mathrm{sp}}\left(\mathbf{p}^{\phi}_{\mathrm{graph}}, \mathbf{p}_{\mathrm{GFM}}\right) \leq \sqrt{\mathcal{L}_{\mathrm{SA}}(\phi)} + \varepsilon_{\mathrm{OT}}, \tag{19}$$

where $\varepsilon_{\mathrm{OT}}$ accounts for the approximation introduced by entropy-regularized optimal transport, finite Sinkhorn iterations, and frequency-band discretization.

We assume that the evaluation loss is bounded in $[0, 1]$ and is L_ℓ-Lipschitz with respect to the prediction score. The support set is sampled uniformly without replacement from the target graph.

C.2 Feature-Branch Bound

We first establish a bound for the feature-adaptation branch under a fixed graph structure.

Lemma C.1 (Finite-class transductive bound)**.** *Let $\mathcal{H}$ be a finite class of predictors. With probability at least $1 - \delta_{\mathrm{lab}}$ over the sampling of the support set, the following inequality holds simultaneously for every $h \in \mathcal{H}$:*

$$\mathcal{R}_{\mathcal{U}}(h) \leq \mathcal{R}_{\mathcal{S}}(h) + \frac{N}{u}\sqrt{\frac{\log\left(2|\mathcal{H}|/\delta_{\mathrm{lab}}\right)}{2m}}. \tag{20}$$

Proof. Let $\mu(h)$ denote the average risk of h over all nodes in $\mathcal{V}$. Since the support set is sampled uniformly without replacement, Hoeffding's inequality gives

$$\Pr\left(\mu(h) - \mathcal{R}_{\mathcal{S}}(h) \geq t\right) \leq \exp(-2mt^2) \tag{21}$$

for any fixed predictor h. Applying the union bound over the finite class $\mathcal{H}$ gives, with probability at least $1 - \delta_{\mathrm{lab}}$,

$$\mu(h) - \mathcal{R}_{\mathcal{S}}(h) \leq \sqrt{\frac{\log\left(2|\mathcal{H}|/\delta_{\mathrm{lab}}\right)}{2m}} \tag{22}$$

simultaneously for all $h \in \mathcal{H}$. Since

$$N\mu(h) = m\mathcal{R}_{\mathcal{S}}(h) + u\mathcal{R}_{\mathcal{U}}(h),$$

we have

$$\mathcal{R}_{\mathcal{U}}(h) - \mathcal{R}_{\mathcal{S}}(h) = \frac{N}{u}\left(\mu(h) - \mathcal{R}_{\mathcal{S}}(h)\right).$$

Substituting the previous inequality proves the lemma. □

Lemma C.2 (Feature-adaptation bound)**.** *Under Assumption 1, for any $\varepsilon_{\mathrm{cov}} > 0$, with probability at least $1 - \delta_{\mathrm{lab}}$ over the support-set sampling, the following inequality holds simultaneously for every $\phi_F \in \Phi_F$:*

$$\begin{aligned}\mathcal{R}_{\mathcal{U}}\left(h_{\phi^\dagger,\phi_F}\right) \leq\ & \mathcal{R}_{\mathcal{S}}\left(h_{\phi^\dagger,\phi_F}\right) + 2\varepsilon_{\mathrm{cov}} \\ & + \frac{N}{u}\sqrt{\frac{p_{\phi_F}\log\left(1 + \frac{2RL_\ell L_{\phi_F}}{\varepsilon_{\mathrm{cov}}}\right) + \log\left(\frac{2}{\delta_{\mathrm{lab}}}\right)}{2m}}.\end{aligned} \tag{23}$$

Proof. Construct an η-cover $\mathcal{N}_\eta$ of Φ_F under the infinity norm. Since $\Phi_F \subseteq [-R, R]^{p_{\phi_F}}$,

$$\log|\mathcal{N}_\eta| \leq p_{\phi_F}\log\left(1 + \frac{2R}{\eta}\right). \tag{24}$$

For every $\phi_F \in \Phi_F$, let $\pi(\phi_F) \in \mathcal{N}_\eta$ denote its nearest point in the cover. By Assumption 1 and the Lipschitz property of the loss,

$$\left|\mathcal{R}_{\mathcal{A}}\left(h_{\phi^\dagger,\phi_F}\right) - \mathcal{R}_{\mathcal{A}}\left(h_{\phi^\dagger,\pi(\phi_F)}\right)\right| \leq L_\ell L_{\phi_F}\eta$$

holds for both $\mathcal{A} = \mathcal{S}$ and $\mathcal{A} = \mathcal{U}$. Choose

$$\eta = \frac{\varepsilon_{\mathrm{cov}}}{L_\ell L_{\phi_F}}.$$

Applying Lemma C.1 to the finite class induced by $\mathcal{N}_\eta$ and transferring the result from $\pi(\phi_F)$ to ϕ_F introduces at most $2\varepsilon_{\mathrm{cov}}$. Substituting the covering-number bound proves the lemma. □

C.3 Structure-Branch Bound

Lemma C.3 (Spectral structure-transfer bound)**.** *Under Assumptions 2 and 3, with probability at least $1-\delta_{\mathrm{probe}}$ over the black-box probing randomness, for every admissible ϕ, every $\phi_F \in \Phi_F$, and every node subset $\mathcal{A} \subseteq \mathcal{V}$,*

$$\left|\mathcal{R}_{\mathcal{A}}\left(h_{\phi,\phi_F}\right) - \mathcal{R}_{\mathcal{A}}\left(h_{\phi^\dagger,\phi_F}\right)\right| \le \Delta_{\mathrm{str}}(\phi), \tag{25}$$

where

$$\Delta_{\mathrm{str}}(\phi) = L_{\mathrm{sp}}\left[\sqrt{\mathcal{L}_{\mathrm{SA}}(\phi)} + \varepsilon_{\mathrm{OT}} + \varepsilon_{\mathrm{probe}}\right] + \varepsilon_{\mathrm{sp}}. \tag{26}$$

Proof. Let $\mathcal{E}_{\mathrm{probe}}$ denote the event

$$d_{\mathrm{sp}}\left(\mathbf{p}_{\mathrm{GFM}}, \mathbf{p}_{F_0}\right) \le \varepsilon_{\mathrm{probe}}.$$

By Assumption 3, $\Pr(\mathcal{E}_{\mathrm{probe}}) \ge 1-\delta_{\mathrm{probe}}$. Work on this event. By the triangle inequality,

$$\begin{aligned} d_{\mathrm{sp}}\left(\mathbf{p}^{\phi}_{\mathrm{graph}}, \mathbf{p}_{F_0}\right) \le\ & d_{\mathrm{sp}}\left(\mathbf{p}^{\phi}_{\mathrm{graph}}, \mathbf{p}_{\mathrm{GFM}}\right) \\ & + d_{\mathrm{sp}}\left(\mathbf{p}_{\mathrm{GFM}}, \mathbf{p}_{F_0}\right). \end{aligned}$$

The first term is bounded by

$$\sqrt{\mathcal{L}_{\mathrm{SA}}(\phi)} + \varepsilon_{\mathrm{OT}},$$

and the second term is bounded by $\varepsilon_{\mathrm{probe}}$. Therefore,

$$d_{\mathrm{sp}}\left(\mathbf{p}^{\phi}_{\mathrm{graph}}, \mathbf{p}_{F_0}\right) \le \sqrt{\mathcal{L}_{\mathrm{SA}}(\phi)} + \varepsilon_{\mathrm{OT}} + \varepsilon_{\mathrm{probe}}.$$

Substituting this inequality into the spectral stability condition in Assumption 2 gives the claimed result. □

C.4 Unified Generalization Bound

Theorem C.4 (PAC Generalization Bound for TAGFM)**.** *Suppose that Assumptions 1, 2, and 3 hold. Let ϕ^* be the structure-adapter parameter obtained by minimizing the label-free spectral alignment loss, and let $\phi_F^* \in \Phi_F$ be the optimized feature-adapter parameter. Then, with probability at least $1-\delta_{\mathrm{lab}}-\delta_{\mathrm{probe}}$ over the sampling of the support set and the randomness of black-box spectral probing,*

$$\begin{aligned} \mathcal{R}_{\mathcal{U}}\left(h_{\phi^*,\phi_F^*}\right) \le\ & \mathcal{R}_{\mathcal{S}}\left(h_{\phi^*,\phi_F^*}\right) + 2\varepsilon_{\mathrm{cov}} \\ & + \frac{N}{u}\sqrt{\frac{p_{\phi_F}\log\left(1+\frac{2RL_\ell L_{\phi_F}}{\varepsilon_{\mathrm{cov}}}\right) + \log\left(\frac{2}{\delta_{\mathrm{lab}}}\right)}{2m}} \\ & + 2L_{\mathrm{sp}}\left[\sqrt{\mathcal{L}_{\mathrm{SA}}(\phi^*)} + \varepsilon_{\mathrm{OT}} + \varepsilon_{\mathrm{probe}}\right] + 2\varepsilon_{\mathrm{sp}}. \end{aligned} \tag{27}$$

Proof. By Lemma C.2, applied to the reference structure $\phi^\dagger$, with probability at least $1-\delta_{\mathrm{lab}}$,

$$\begin{aligned} \mathcal{R}_{\mathcal{U}}\left(h_{\phi^\dagger,\phi_F}\right) \le\ & \mathcal{R}_{\mathcal{S}}\left(h_{\phi^\dagger,\phi_F}\right) + 2\varepsilon_{\mathrm{cov}} \\ & + \frac{N}{u}\sqrt{\frac{p_{\phi_F}\log\left(1+\frac{2RL_\ell L_{\phi_F}}{\varepsilon_{\mathrm{cov}}}\right) + \log\left(\frac{2}{\delta_{\mathrm{lab}}}\right)}{2m}} \end{aligned} \tag{28}$$

simultaneously for every $\phi_F \in \Phi_F$. Hence, the inequality also holds for the data-dependent parameter ϕ_F^*.

On the probing event from Lemma C.3, applying that lemma with $\mathcal{A} = \mathcal{U}$ gives

$$\mathcal{R}_{\mathcal{U}}\left(h_{\phi^*,\phi_F^*}\right) \le \mathcal{R}_{\mathcal{U}}\left(h_{\phi^\dagger,\phi_F^*}\right) + \Delta_{\mathrm{str}}(\phi^*). \tag{29}$$

Applying the feature-branch bound to the right-hand side yields

$$\begin{aligned}\mathcal{R}_{\mathcal{U}}\left(h_{\phi^*,\phi_F^*}\right) \le \mathcal{R}_{\mathcal{S}}\left(h_{\phi^\dagger,\phi_F^*}\right) + 2\varepsilon_{\mathrm{cov}} \\ + \frac{N}{u}\sqrt{\frac{p_{\phi_F}\log\left(1+\frac{2RL_\ell L_{\phi_F}}{\varepsilon_{\mathrm{cov}}}\right)+\log\left(\frac{2}{\delta_{\mathrm{lab}}}\right)}{2m}} + \Delta_{\mathrm{str}}(\phi^*).\end{aligned} \tag{30}$$

Applying Lemma C.3 again, now with $\mathcal{A} = \mathcal{S}$, gives

$$\mathcal{R}_{\mathcal{S}}\left(h_{\phi^\dagger,\phi_F^*}\right) \le \mathcal{R}_{\mathcal{S}}\left(h_{\phi^*,\phi_F^*}\right) + \Delta_{\mathrm{str}}(\phi^*). \tag{31}$$

Combining the last two inequalities and substituting the definition of $\Delta_{\mathrm{str}}(\phi^*)$ proves Eq. equation 27. Finally, the two events used in the feature and structure bounds hold simultaneously with probability at least $1 - \delta_{\mathrm{lab}} - \delta_{\mathrm{probe}}$ by the union bound. □

D Additional Experiments

D.1 Classification results

This section reports the additional classification results of TAGFM. Table 2 presents the 5-shot node classification results, while Tables 3 and 4 report the 1-shot and 5-shot graph classification results, respectively. All experiments are conducted over 50 random splits, and the results are reported as mean accuracy with sample standard deviation. The source datasets and dataset abbreviations follow Table 1. For each GFM backbone, we compare the original adaptation strategy, GFMate, and TAGFM. Colored values indicate the average improvement over the original adaptation method across the seven target datasets.

From Table 3, compared with the original adaptation method, TAGFM achieves positive average improvements of 7.73%, 4.27%, 3.96%, 3.74%, and 14.28% for GCOPE, SAMGPT, BRIDGE, MDGFM, and GRAVER, respectively. Compared with GFMate, TAGFM provides more stable gains across different backbones, further indicating that target-side feature–structure adaptation is also effective for graph-level prediction. Although the improvement varies across datasets, the consistently positive average gains demonstrate the transferability of TAGFM to graph classification tasks.

Table 2: Five-shot node classification accuracy (mean ± sample standard deviation, %) over 50 splits. Blue and red values beside GFMate and TAGFM report their average accuracy changes relative to the original adaptation of the same backbone across the seven target datasets, in percentage points. The highest mean in each three-row backbone group is bolded. Cornell uses all available examples in the class containing fewer than five nodes, resulting in per-class support counts of $[5, 1, 5, 5, 5]$.

Source	CS PM Ph Com Co Ch	CR PM Ph Com Co Ch	CR CS Ph Com Co Ch	CR CS PM Com Co Ch	CR CS PM Ph Co Ch	CR CS PM Ph Com Ch	CR CS PM Ph Com Co
Model / Target	CR	CS	PM	Ph	Com	Co	Ch
GCN	56.62 ± 4.2	53.11 ± 4.0	65.26 ± 4.6	77.05 ± 3.8	65.48 ± 4.5	37.86 ± 5.4	34.55 ± 3.4
GAT	57.29 ± 3.9	53.62 ± 4.0	65.82 ± 5.3	75.95 ± 4.2	63.85 ± 4.2	34.63 ± 5.9	33.81 ± 3.2
DGI	56.71 ± 3.5	52.25 ± 3.7	52.55 ± 3.9	79.73 ± 3.5	68.59 ± 4.9	36.37 ± 5.7	33.00 ± 3.3
GraphCL	56.26 ± 3.4	52.01 ± 3.7	54.90 ± 4.5	81.63 ± 3.4	70.11 ± 4.7	35.74 ± 5.6	33.18 ± 2.8
DSSL	40.64 ± 4.9	39.94 ± 3.8	34.07 ± 8.3	78.86 ± 4.5	67.68 ± 4.5	32.88 ± 5.5	25.44 ± 1.6
GraphACL	65.72 ± 3.8	54.08 ± 4.3	64.94 ± 6.2	81.28 ± 3.4	71.43 ± 4.7	37.07 ± 5.7	35.31 ± 3.3
GCOPE	60.71 ± 4.3	46.65 ± 6.6	54.61 ± 7.4	63.91 ± 4.2	55.06 ± 6.4	30.98 ± 9.5	24.78 ± 2.4
+ GFMate -2.08	56.76 ± 3.6	49.47 ± 4.5	49.51 ± 8.5	61.10 ± 3.6	51.79 ± 5.1	30.44 ± 5.6	23.07 ± 2.5
+ TAGFM +7.66	**69.07 ± 2.5**	**56.97 ± 2.9**	**69.37 ± 6.4**	**71.93 ± 3.4**	**62.41 ± 9.0**	**32.53 ± 6.4**	**28.07 ± 2.9**
SAMGPT	59.82 ± 4.1	58.34 ± 4.6	55.41 ± 5.8	78.45 ± 4.3	65.86 ± 6.1	40.91 ± 5.8	25.18 ± 3.0
+ GFMate -2.41	59.31 ± 3.7	60.22 ± 2.7	47.37 ± 4.1	69.73 ± 3.3	59.10 ± 4.8	**44.17 ± 4.9**	27.22 ± 3.0
+ TAGFM +6.89	**73.59 ± 3.1**	**61.47 ± 2.8**	**69.80 ± 7.3**	**82.50 ± 3.2**	**72.58 ± 4.8**	36.12 ± 5.5	**36.13 ± 3.1**
BRIDGE	62.90 ± 4.5	61.20 ± 4.2	62.00 ± 4.6	82.09 ± 4.6	70.39 ± 5.8	41.17 ± 7.7	26.68 ± 3.5
+ GFMate -3.32	65.47 ± 3.2	59.02 ± 3.3	61.12 ± 5.9	65.53 ± 4.3	50.02 ± 5.0	**53.14 ± 4.9**	28.89 ± 3.3
+ TAGFM +5.51	**74.90 ± 2.3**	**63.94 ± 2.8**	**70.48 ± 5.7**	**83.11 ± 2.9**	**70.90 ± 4.6**	43.89 ± 4.6	**37.76 ± 3.6**
MDGFM	63.99 ± 3.5	62.01 ± 4.3	65.84 ± 5.7	80.88 ± 3.9	67.44 ± 5.5	40.10 ± 9.0	32.70 ± 4.1
+ GFMate -4.19	68.24 ± 3.2	60.43 ± 2.8	65.01 ± 4.9	69.63 ± 4.4	53.66 ± 4.5	**41.30 ± 5.3**	25.34 ± 3.3
+ TAGFM +3.67	**75.57 ± 2.3**	**64.74 ± 2.2**	**70.36 ± 6.1**	**82.98 ± 3.1**	**73.20 ± 4.8**	34.56 ± 4.9	**37.23 ± 3.5**
GRAVER	43.32 ± 10.9	54.56 ± 9.2	46.75 ± 10.3	66.51 ± 9.6	64.02 ± 7.5	43.21 ± 8.0	30.35 ± 3.6
+ GFMate -0.04	47.51 ± 4.1	59.71 ± 3.0	36.28 ± 5.2	65.46 ± 4.2	56.27 ± 5.3	**51.38 ± 5.5**	31.82 ± 3.2
+ TAGFM +12.12	**71.62 ± 3.0**	**64.28 ± 2.3**	**67.44 ± 6.7**	**81.45 ± 3.1**	**71.40 ± 5.3**	43.22 ± 5.1	**34.18 ± 3.0**

Table 3: One-shot graph classification accuracy (mean ± sample standard deviation, %) over 50 splits. Source lists the backbone pretraining datasets. Dataset abbreviations and colors follow Table 1. Blue and red values beside GFMate and TAGFM respectively report their average improvements over the original adaptation of the same backbone across the seven target datasets, in percentage points. The highest mean in each three-row backbone group is bolded.

Source	CS PM Ph Com Co Ch	CR PM Ph Com Co Ch	CR CS Ph Com Co Ch	CR CS PM Com Co Ch	CR CS PM Ph Co Ch	CR CS PM Ph Com Ch	CR CS PM Ph Com Co
Model / Target	CR	CS	PM	Ph	Com	Co	Ch
GCN	46.42 ± 8.0	45.90 ± 6.7	58.47 ± 8.5	63.85 ± 7.8	47.31 ± 6.1	20.57 ± 6.3	27.43 ± 5.1
GAT	46.08 ± 7.6	45.10 ± 7.2	57.14 ± 9.7	60.56 ± 9.4	47.37 ± 7.7	18.62 ± 8.9	27.16 ± 4.6
DGI	49.17 ± 8.5	42.37 ± 6.8	50.31 ± 7.0	68.56 ± 7.6	56.34 ± 6.8	27.38 ± 10.2	27.19 ± 4.2
GCC	17.18 ± 2.6	20.30 ± 4.0	35.55 ± 4.6	27.82 ± 6.6	19.65 ± 4.9	20.93 ± 5.6	24.42 ± 3.7
GraphCL	49.40 ± 8.4	43.89 ± 5.7	50.39 ± 7.3	69.67 ± 7.5	56.92 ± 6.7	26.97 ± 9.3	27.73 ± 4.5
DSSL	37.53 ± 7.1	32.41 ± 6.4	35.07 ± 8.8	66.30 ± 7.4	54.25 ± 7.6	21.46 ± 8.4	24.44 ± 4.8
GraphACL	55.25 ± 7.7	45.19 ± 6.4	56.65 ± 9.4	67.63 ± 8.0	54.41 ± 6.6	24.57 ± 6.5	27.99 ± 4.5
GCOPE	48.16 ± 7.6	35.82 ± 7.4	40.01 ± 8.7	60.62 ± 8.3	48.23 ± 13.4	25.11 ± 10.8	27.17 ± 4.6
+ GFMate +7.74	56.79 ± 6.7	**48.73 ± 5.7**	59.72 ± 8.6	62.27 ± 8.7	56.52 ± 9.0	**28.27 ± 6.6**	27.01 ± 5.0
+ TAGFM +7.73	**58.44 ± 8.2**	42.70 ± 9.5	**62.50 ± 11.6**	**66.27 ± 6.8**	**57.49 ± 9.9**	24.45 ± 14.6	**27.39 ± 5.7**
SAMGPT	53.61 ± 8.6	43.73 ± 6.1	54.53 ± 7.5	67.99 ± 8.5	54.55 ± 7.6	23.27 ± 8.6	27.70 ± 4.4
+ GFMate +1.78	54.52 ± 7.1	49.61 ± 6.5	59.48 ± 7.3	66.72 ± 8.6	51.57 ± 8.6	**28.47 ± 7.3**	27.44 ± 4.4
+ TAGFM +4.27	**59.60 ± 7.6**	**50.45 ± 7.7**	**62.49 ± 11.8**	**69.23 ± 7.3**	**58.51 ± 9.2**	26.62 ± 9.2	**28.36 ± 4.7**
BRIDGE	54.79 ± 7.9	48.66 ± 7.1	58.66 ± 8.8	68.01 ± 8.5	55.82 ± 7.6	23.72 ± 9.4	28.45 ± 4.8
+ GFMate +0.88	60.22 ± 6.5	50.35 ± 6.1	60.97 ± 8.9	66.65 ± 8.4	49.59 ± 6.8	**28.78 ± 7.0**	27.73 ± 4.4
+ TAGFM +3.96	**64.46 ± 7.0**	**52.77 ± 8.0**	**63.48 ± 10.6**	**70.22 ± 7.1**	**58.51 ± 9.1**	27.61 ± 6.3	**28.79 ± 4.7**
MDGFM	57.55 ± 6.6	51.01 ± 7.5	59.41 ± 9.3	68.23 ± 8.7	53.68 ± 8.2	22.87 ± 10.0	27.86 ± 5.0
+ GFMate +0.74	60.29 ± 6.7	48.85 ± 7.6	60.48 ± 8.7	66.14 ± 8.5	52.72 ± 7.0	**30.07 ± 6.8**	27.27 ± 4.6
+ TAGFM +3.74	**64.47 ± 7.1**	**54.46 ± 7.6**	**63.06 ± 10.2**	**70.72 ± 7.0**	**60.41 ± 9.4**	25.27 ± 4.2	**28.40 ± 4.7**
GRAVER	27.94 ± 12.8	22.54 ± 13.0	50.24 ± 14.0	62.95 ± 8.1	54.85 ± 14.4	18.35 ± 13.2	23.09 ± 4.6
+ GFMate +10.20	54.81 ± 5.7	51.52 ± 6.0	45.16 ± 6.7	69.89 ± 6.5	53.72 ± 6.0	**28.74 ± 7.3**	27.51 ± 4.4
+ TAGFM +14.28	**57.61 ± 8.6**	**54.84 ± 7.0**	**57.97 ± 11.5**	**71.34 ± 7.8**	**61.25 ± 8.7**	28.52 ± 6.9	**28.40 ± 4.1**

Table 4: Five-shot graph classification accuracy (mean ± sample standard deviation, %) over 50 splits. Blue and red values have the same meaning as in Table 2. The highest mean in each three-row backbone group is bolded. Traditional methods use frozen encoders with ego-graph mean pooling and a 200-step linear probe without graph-test parameter selection. Cornell uses per-class support counts of $[5, 1, 5, 5, 5]$.

Source	CS PM Ph Com Co Ch	CR PM Ph Com Co Ch	CR CS Ph Com Co Ch	CR CS PM Com Co Ch	CR CS PM Ph Co Ch	CR CS PM Ph Com Ch	CR CS PM Ph Com Co
Model / Target	CR	CS	PM	Ph	Com	Co	Ch
GCN	63.97 ± 3.9	55.04 ± 4.2	68.16 ± 5.9	78.63 ± 2.8	65.92 ± 4.0	36.09 ± 5.1	31.93 ± 3.5
GAT	63.48 ± 3.6	55.21 ± 4.0	67.06 ± 5.8	77.56 ± 2.5	65.39 ± 3.8	32.49 ± 6.0	30.15 ± 3.8
DGI	68.46 ± 4.3	57.44 ± 4.7	62.72 ± 5.5	81.05 ± 2.6	71.36 ± 4.8	34.17 ± 7.4	35.73 ± 4.3
GCC	18.96 ± 2.7	23.19 ± 2.9	36.06 ± 3.5	35.74 ± 4.6	26.06 ± 5.6	23.17 ± 4.3	26.34 ± 3.4
GraphCL	67.87 ± 4.3	57.27 ± 4.8	63.29 ± 6.3	81.30 ± 2.6	71.70 ± 4.8	33.19 ± 6.7	35.57 ± 4.0
DSSL	53.64 ± 4.4	44.75 ± 4.6	35.08 ± 9.1	80.01 ± 3.0	69.39 ± 4.6	31.58 ± 6.0	25.56 ± 4.4
GraphACL	70.18 ± 3.5	55.85 ± 4.6	66.65 ± 6.3	80.96 ± 2.9	70.76 ± 3.9	33.54 ± 5.8	33.48 ± 3.9
GCOPE	66.08 ± 3.8	46.50 ± 7.3	54.95 ± 11.5	73.51 ± 4.9	57.74 ± 14.5	29.20 ± 9.3	34.17 ± 3.6
+ GFMate -1.03	61.75 ± 3.3	51.52 ± 4.7	53.23 ± 11.1	63.19 ± 7.0	61.59 ± 5.3	36.36 ± 6.6	27.32 ± 3.8
+ TAGFM +8.66	**70.08 ± 2.6**	**57.15 ± 3.0**	**69.24 ± 6.2**	**76.76 ± 3.5**	**68.66 ± 3.9**	**43.90 ± 10.3**	**36.98 ± 3.1**
SAMGPT	66.55 ± 3.3	60.45 ± 4.2	61.19 ± 4.4	77.33 ± 3.7	66.11 ± 4.9	35.21 ± 5.9	30.67 ± 3.7
+ GFMate -1.52	68.48 ± 2.9	**62.30 ± 3.1**	53.66 ± 6.3	73.27 ± 4.6	60.16 ± 6.8	**40.36 ± 6.9**	28.64 ± 3.8
+ TAGFM +3.88	**72.21 ± 3.3**	61.73 ± 2.7	**69.73 ± 6.4**	**81.35 ± 3.0**	**70.12 ± 4.1**	36.01 ± 5.5	**33.50 ± 3.7**
BRIDGE	69.56 ± 3.4	62.06 ± 5.0	63.29 ± 5.7	77.55 ± 5.1	67.71 ± 5.4	37.59 ± 7.8	29.94 ± 4.1
+ GFMate -0.85	72.31 ± 3.0	62.87 ± 3.5	65.46 ± 5.8	72.75 ± 4.0	54.81 ± 7.5	**44.60 ± 7.3**	28.93 ± 3.4
+ TAGFM +3.53	**74.72 ± 2.2**	**63.67 ± 3.8**	**69.92 ± 5.8**	**81.55 ± 3.3**	**70.15 ± 4.3**	41.40 ± 5.4	**30.98 ± 3.4**
MDGFM	70.35 ± 3.0	63.82 ± 4.7	67.24 ± 6.1	76.55 ± 5.5	66.14 ± 5.0	**40.26 ± 7.9**	29.64 ± 3.4
+ GFMate -1.15	73.55 ± 2.8	63.40 ± 3.3	67.48 ± 5.7	73.75 ± 4.0	58.64 ± 6.7	39.42 ± 5.9	29.73 ± 3.9
+ TAGFM +2.16	**74.93 ± 2.1**	**64.41 ± 3.5**	**69.88 ± 5.5**	**81.13 ± 3.0**	**71.02 ± 3.9**	35.91 ± 6.8	**31.81 ± 4.1**

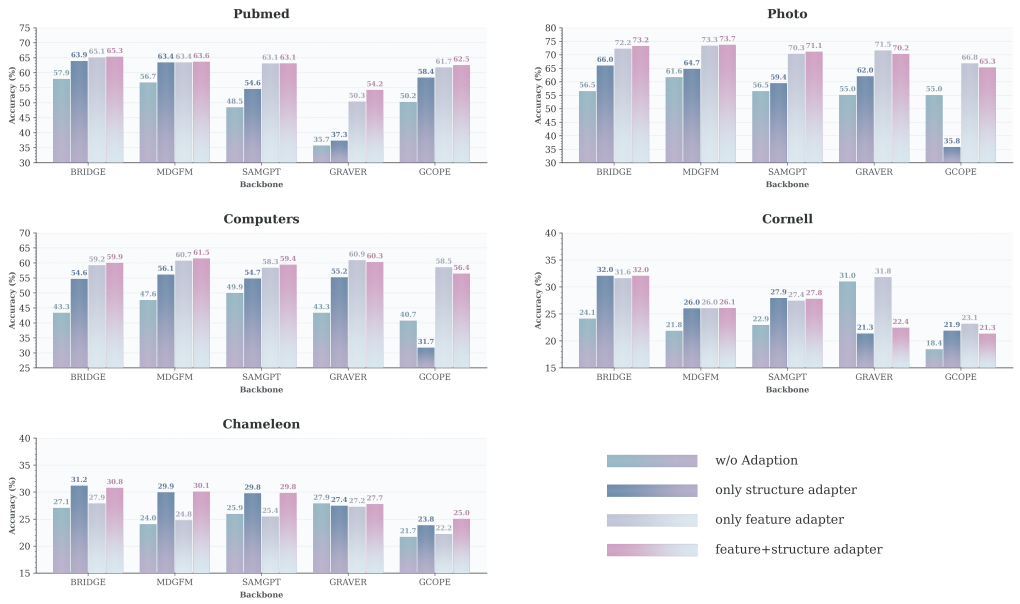

Figure 5: Ablation results on five datasets under the 1-shot setting.

D.2 Ablation Study

Figure 5 reports the ablation results on the remaining five target datasets (*i.e.,* PubMed, Photo, Computers, Cornell, and Chameleon). The results are consistent with the observations on Cora and Citeseer. Both the structure adapter and the feature adapter generally improve the performance over the non-adapted setting, while their combination achieves the best or near-best results for most backbone–dataset pairs. In particular, the joint adapter obtains clear improvements on Pubmed, Photo, and Computers, demonstrating the benefit of combining structural and feature adaptation.

The results on Cornell and Chameleon show that the relative contribution of the two adapters depends on the target graph and the backbone. For example, the structure adapter is more beneficial for some Cornell and Chameleon settings, whereas the feature adapter provides larger gains on others. Moreover, the joint use of the two adapters is not always strictly additive, reflecting the interaction between feature transformation and structural adjustment. Nevertheless, the two components consistently provide useful adaptation signals across different target graphs, which supports the effectiveness of the proposed feature–structure adaptation framework.

D.3 Parameter Sensitivity

We investigate three hyperparameters that act at different stages of TAGFM: the feature-adaptation strength γ, the MCR loss weight ω, and the representation fusion coefficient β_{f}. They control input modification, optimization, and representation fusion, respectively, while the GFM backbone parameters θ remain frozen.

Feature-adaptation strength γ. Let $\Delta\mathbf{X}_{\phi_F} = \mathrm{MLP}_{\phi_F}(\mathbf{X})$ denote the feature residual learned by the feature adapter. As defined in Eq. equation 1, the input to the frozen backbone is

$$\widetilde{\mathbf{X}} = \mathbf{X} + \gamma\Delta\mathbf{X}_{\phi_F}, \qquad \left\|\widetilde{\mathbf{X}} - \mathbf{X}\right\|_F = |\gamma| \, \|\Delta\mathbf{X}_{\phi_F}\|_F \, . \tag{32}$$

Thus, for fixed adapter parameters, γ directly scales the deviation from the original node features. Setting $\gamma = 0$ removes the feature modification, while retaining any structural adaptation. Moreover, the Jacobian of the adapted features with respect to the feature-adapter parameters satisfies

$$\frac{\partial \, \mathrm{vec}(\widetilde{\mathbf{X}})}{\partial \phi_F} = \gamma \frac{\partial \, \mathrm{vec}(\Delta\mathbf{X}_{\phi_F})}{\partial \phi_F}, \tag{33}$$

where $\mathrm{vec}(\cdot)$ denotes vectorization. Consequently, γ also enters the gradient path through which the adaptation objective updates ϕ_F. Its effect on the trained model is therefore not limited to rescaling a fixed residual, since the learned residual itself can change when γ is varied.

MCR loss weight ω. The feature adapter is optimized using the objective in Eq. equation 14:

$$\mathcal{J}_F = \mathcal{L}_{\mathrm{sup}} + \omega \mathcal{L}_{\mathrm{MCR}}, \qquad \nabla_{\phi_F} \mathcal{J}_F = \nabla_{\phi_F} \mathcal{L}_{\mathrm{sup}} + \omega \nabla_{\phi_F} \mathcal{L}_{\mathrm{MCR}}. \tag{34}$$

The supervised term uses support-set labels, whereas the MCR term incorporates the feature distribution of both labeled and unlabeled target nodes through their class assignments. Accordingly, ω controls the contribution of this distributional constraint to feature-adapter optimization. When $\omega = 0$, the MCR term provides no optimization signal; increasing ω gives its gradient greater weight at a fixed parameter state. However, the actual balance also depends on the magnitudes and directions of the two gradients. Importantly, ω weights the entire MCR objective and is distinct from β in Eq. equation 5, which controls the balance between the overall and class-conditional coding rates within that objective.

Representation fusion coefficient β_{f}. In the experimental implementation, downstream prediction uses a weighted combination of the original and jointly adapted backbone representations:

$$\begin{aligned} \mathbf{H}_{\mathrm{orig}} &= f_\theta(\mathbf{X}, \mathbf{A}), \\ \mathbf{H}_{\mathrm{adapt}} &= f_\theta(\widetilde{\mathbf{X}}, \mathbf{A}_{\phi^*}), \\ \mathbf{H}_{\mathrm{fuse}} &= \mathrm{Norm}(\mathbf{H}_{\mathrm{orig}}) + \beta_{\mathrm{f}} \, \mathrm{Norm}(\mathbf{H}_{\mathrm{adapt}}), \end{aligned} \tag{35}$$

where Norm denotes row-wise ℓ_2 normalization. Normalizing the two branches before fusion makes β_{f} control their relative contribution without being confounded by differences in their raw embedding norms. Setting $\beta_{\mathrm{f}} = 0$ removes the adapted branch from the fused representation; increasing it places greater weight on the representation produced by joint feature and structure adaptation. This operation is an additive fusion, so β_{f} is not restricted to $[0, 1]$. It balances the original and jointly adapted representations, rather than separately weighting the feature and structure adapters.

Experimental results. Figure 6 reports the 1-shot classification accuracy across seven target datasets and five frozen GFM backbones. We vary one hyperparameter at a time over $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$, keeping the remaining settings fixed. Each point reports the mean accuracy over the same 50 random splits.

The results show that sensitivity depends on both the backbone and the target graph. Performance is relatively stable over a range of values, whereas extreme values lead to degradation in some settings. These observations are consistent with the distinct roles of the three hyperparameters: controlling the magnitude of input modification, the balance of optimization signals, and the contribution of adapted representations. They do not imply that stronger adaptation consistently improves performance or that a single configuration is optimal for every backbone and target graph. Since this analysis varies one hyperparameter at a time, it characterizes sensitivity conditional on the remaining settings rather than interactions among all three hyperparameters.

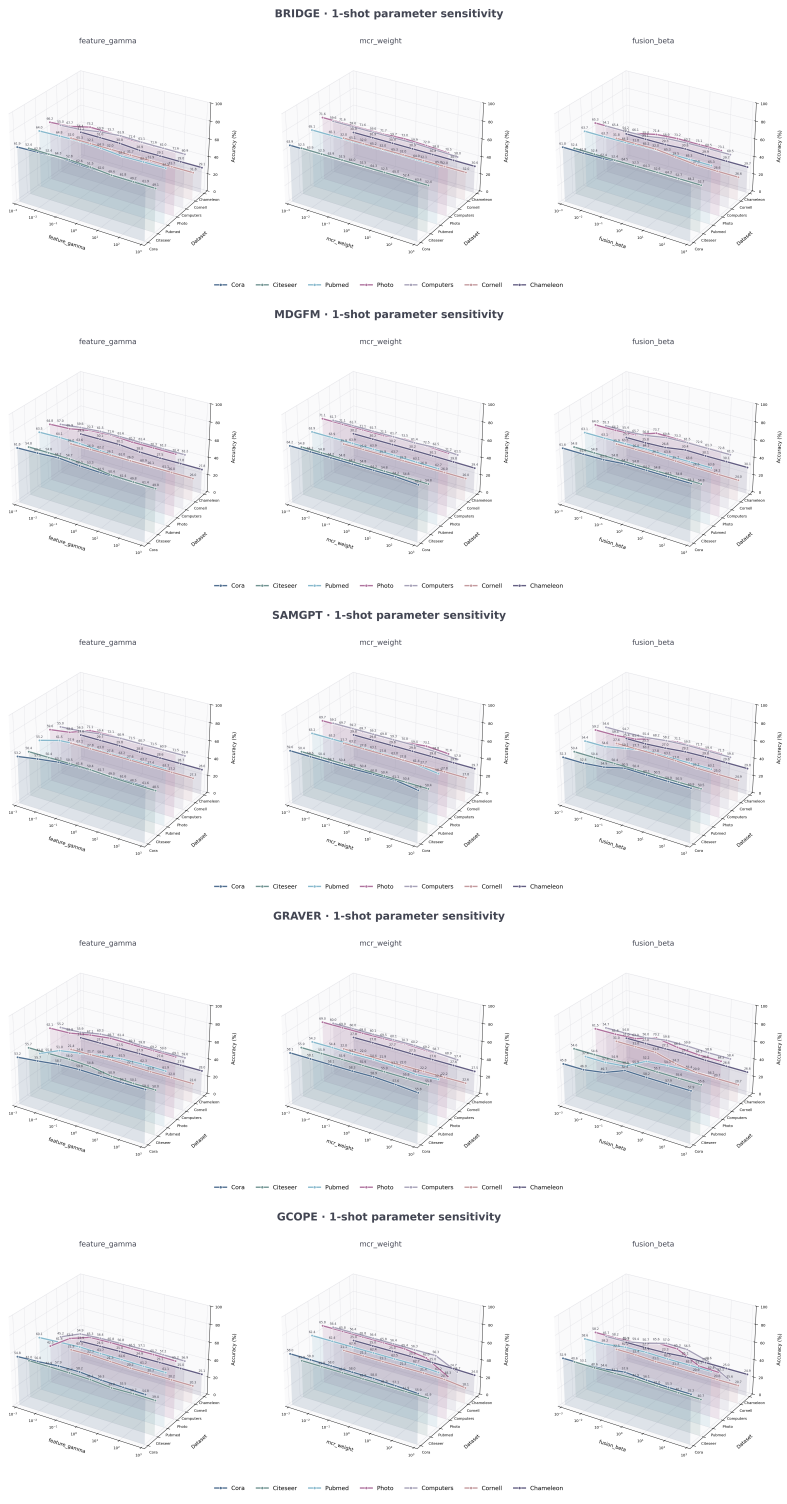

Figure 6: Parameter sensitivity under the 1-shot setting. Each hyperparameter is varied individually while the remaining settings are fixed. Results are averaged over 50 random splits.