TI-GFM: TEST-TIME INVARIANT PROTOTYPE LEARNING FOR TEXT-ATTRIBUTED GRAPH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph foundation models (GFMs) aim to transfer pretrained graph encoders across domains and tasks. For text-attributed graphs, semantic class prototypes provide a direct interface for zero-shot prediction, but cross-domain shift can misalign them with target-domain class distributions. A shifted prototype may drift away from its corresponding class or approach confusing classes, leading to unreliable prototype-based prediction. Here, we introduce a Prototype Shift ($\mathcal{PS}$) metric to quantify this misalignment by combining intra-class deviation and nearest inter-class separation in the target domain. Based on this diagnosis, we propose TI-GFM, a test-time invariant prototype learning framework for text-attributed GFMs. TI-GFM learns perturbation-robust representations through Text-Attributed Graph Pre-training (TGP) and refines semantic prototypes at test time using target nodes with prediction consistency across perturbed graph views. This design injects invariant target-domain evidence into semantic prototypes without retraining the backbone encoder. Experiments on seven cross-domain target datasets show that TI-GFM achieves the best average rank and the best accuracy on six of seven zero-shot node classification benchmarks, with consistent gains in few-shot node classification and zero-shot link prediction. Ablations, $\mathcal{PS}$ analysis, and theoretical margin analysis further indicate that reducing prototype shift is a main factor behind cross-domain transfer performance in text-attributed GFMs.

1 INTRODUCTION

Graph foundation models (GFMs) aim to learn reusable graph encoders transferable across domains and tasks, following a "pretrain-once, generalize-to-many" paradigm Bommasani et al. (2021); Liu et al. (2025). This is especially relevant for text-attributed graphs (TAGs), where node texts, graph topology, and class descriptions can be aligned in a shared semantic space Lin et al. (2021); Yang et al. (2021); Zhao et al. (2022); Brannon et al. (2024). Recent TAG-based GFMs leverage this property for zero-shot and few-shot cross-domain transfer Huang et al. (2024); Zhang et al. (2025); Xu & Ding (2025); Zhu et al. (2025b); Tang et al. (2024). However, such a transfer raises a reliability issue beyond representation quality: class semantics encoded before deployment may not match the target-domain class geometry used for prediction.

Existing Paradigms. Existing few-shot and zero-shot GFMs for TAGs mainly follow two paradigms. One line aligns graph representations with the language space and uses LLMs or language-conditioned modules for task prediction, as in GraphGPT Tang et al. (2024), UniGraph He et al. (2025), TEA-GLM Wang et al. (2024), and GOFA Kong et al. (2025). The other performs semantic prototype matching, where class descriptions are encoded as class prototypes and target nodes are classified by similarity, as in ZeroG Li et al. (2024) and GraphCLIP Zhu et al. (2025b). Although these methods differ in architecture, they share an implicit assumption: the transferred class semantics remain representative of the target-domain class structure. This assumption breaks down under cross-domain shifts in text distribution, graph topology, and class-specific neighborhoods. Such invalidation is especially harmful for prototype-based prediction, where a misplaced class prototype directly changes the decision boundary.

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.

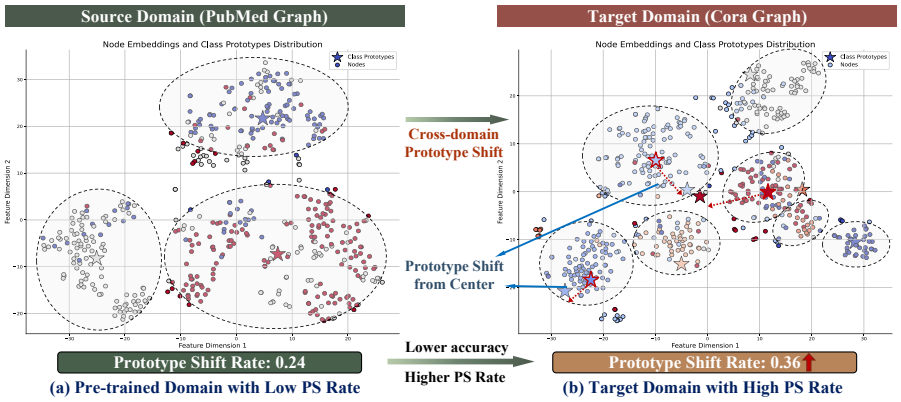


Figure 1: Prototype Shift ($\mathcal{PS}$): cross-domain transfer can misalign semantic prototypes with target-domain class structures, increasing $\mathcal{PS}$ and degrading zero-shot performance.

Prototype Shift Bottleneck. We formulate this phenomenon as **Prototype Shift** ($\mathcal{PS}$), which measures whether the prototypes used for prediction remain close to their corresponding target-domain class distributions and separated from the nearest confusing classes. Larger $\mathcal{PS}$ values indicate more severe prototype misalignment and weaker prototype reliability. Figure 1 illustrates this issue in a cross-domain transfer case. In the source graph, semantic prototypes are close to the empirical class structures, yielding a low $\mathcal{PS}$ of 0.24. After transfer to the target graph, distributional and structural changes move the prototypes away from the corresponding target-domain classes or toward confusing classes, increasing $\mathcal{PS}$ to 0.36. This observation motivates a more explicit treatment of prototype reliability: beyond improving representations, cross-domain GFMs need a mechanism to diagnose and correct prototype misalignment at test time.

Our Solution. Motivated by this observation, we propose TI-GFM, a test-time invariant prototype learning framework for text-attributed GFMs. TI-GFM contains two components. First, **Text-Attributed Graph Pre-training (TGP)** learns perturbation-robust node representations by coupling textual semantics with graph structure during pre-training. Second, **Test-Time Invariant Prototype (TIP)** refines semantic prototypes with an invariance criterion based on cross-view prediction consistency. In this way, TI-GFM injects invariant target-domain information into semantic prototypes without retraining the backbone encoder, thereby reducing prototype shift in zero-shot and few-shot transfer settings.

Empirical and Theoretical Findings. Experiments on seven cross-domain datasets show that TI-GFM achieves the best average rank and the best accuracy on six of seven zero-shot node classification benchmarks, with consistent gains in few-shot node classification and zero-shot link prediction. Ablation and $\mathcal{PS}$ analyses show that TIP mainly alleviates prototype shift, while TGP provides complementary robustness. Theoretical margin analysis further links lower $\mathcal{PS}$ to a larger normalized prototype reliability margin, explaining its connection to transfer performance.

Contributions of this work. (1) We introduce a Prototype Shift ($\mathcal{PS}$) metric to quantify target-domain prototype misalignment via intra-class deviation and nearest inter-class separation, with a margin-based interpretation. (2) We propose TI-GFM, a test-time invariant prototype learning framework that combines text-attributed graph pre-training with consistency-based prototype refinement across perturbed graph views. (3) We evaluate TI-GFM on zero-shot/few-shot node classification and zero-shot link prediction across seven cross-domain datasets, showing that mitigating prototype shift improves transfer in text-attributed GFMs.

2 Related Work

2.1 Text-Attributed Graphs

Text-attributed graphs (TAGs) integrate node texts with graph topology and are widely used in citation networks, social media, and recommendation systems Yan et al. (2023). Early TAG methods follow cascaded pipelines that encode node texts and propagate graph information separately Chen et al.

(2023); Tu et al. (2017). Later approaches jointly model textual semantics and structural context, including BertGCN Lin et al. (2021), GraphFormers Yang et al. (2021), and Patton Jin et al. (2023). Recent LLM-enhanced TAG methods further explore graph reasoning and graph-text alignment Jin et al. (2024); Yang et al. (2024b), including natural-language graph reasoning Zhao et al. (2024); Fatemi et al. (2023), LLM-based graph instruction tuning and embedding alignment Tang et al. (2024); Chen et al. (2024a); Wang et al. (2024), and hybrid LLM-GNN frameworks Chen et al. (2024b); Zhao et al. (2022); Huang et al. (2024).

Limitation. Although these methods improve TAG representation learning, they mainly focus on in-domain or task-specific adaptation and rarely investigate whether semantic representations or prototypes remain aligned with target-domain class structures under cross-domain shifts.

2.2 Graph Foundation Models

Inspired by foundation models in NLP and vision Radford et al. (2021); Devlin et al. (2019), graph foundation models (GFMs) aim to learn transferable representations from large-scale graph data. Existing GFMs can be categorized into text-free and text-attributed paradigms. Text-free GFMs capture transferable structural knowledge from graph topology and attributes, with recent studies such as BRIDGE Yuan et al. (2025) and MDGFM Wang et al. (2025a) exploring multi-domain pre-training for cross-graph transfer. Text-attributed GFMs leverage natural language as a semantic interface, where OFA Liu et al. (2023a), ZeroG Li et al. (2024), and GraphCLIP Zhu et al. (2025b) perform zero-shot prediction via graph-text alignment, while UniGraph He et al. (2025), GraphGPT Tang et al. (2024), TEA-GLM Wang et al. (2024), and GOFA Kong et al. (2025) further incorporate LLMs for graph-language reasoning.

Limitation. Despite improved transferability, existing GFMs mainly rely on pre-deployment semantic alignment. Under shifts in text, topology, or class neighborhoods, fixed semantic prototypes may fail to reflect target-domain class geometry. Existing GFMs rarely quantify or correct such prototype misalignment at test time.

2.3 Prototype Learning for Cross-Domain Transfer

Prototype-based prediction relies on prototypes that are close to same-class samples and separated from competing classes Snell et al. (2017); Weinberger & Saul (2009). This paradigm naturally supports zero-shot graph transfer, where class descriptions act as semantic prototypes Li et al. (2024); Zhu et al. (2025b). However, semantic prototypes derived from source domains may become misaligned with target-domain class structures under cross-domain shifts. Existing prototype-based GFMs rarely quantify or correct such misalignment. To address this issue, TI-GFM introduces a Prototype Shift ($\mathcal{PS}$) metric and performs test-time invariant prototype refinement without retraining the backbone.

3 Problem Formulation

3.1 Notation and Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X}, \mathbf{T}^v, \mathbf{T}^s, \mathbf{T}^p)$ denote a text-attributed graph, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set, $\mathbf{A} \in \{0,1\}^{N\times N}$ is the adjacency matrix, $\mathbf{T}^v$ denotes node-text tokens, and $\mathbf{X} \in \mathbb{R}^{N\times d_x}$ is the text-derived feature matrix. For each node v_i, we extract its ego-network $\mathcal{G}^{(i)} = (\mathcal{V}^{(i)}, \mathbf{A}^{(i)}, \mathbf{X}^{(i)})$ and denote the LLM-generated subgraph summary as $\mathbf{T}^s_i$. In cross-domain transfer, a graph encoder pre-trained on unlabeled source graphs $\mathcal{D}_{\text{train}}$ is transferred to disjoint target graphs $\mathcal{D}_{\text{test}}$. Let $\mathcal{C} = \{1, \ldots, C\}$ be the target label space and $\mathbf{T}^p$ the tokenized class description prompts. Zero-shot transfer uses no target labels, while few-shot transfer adopts a labeled support set $\mathcal{S}_c$ for prototype adaptation.

3.2 Prototype Shift ($\mathcal{PS}$)

For class c, let $\mathcal{V}_c$ be the target-domain nodes with ground-truth label c, and let $\mathbf{z}_i$ be the representation of node v_i. We define the empirical class center $\boldsymbol{\mu}_c$ as the mean of the representation of nodes belonging to $\mathcal{V}_c$. This center is used only as an evaluation reference for $\mathcal{PS}$ analysis and is not

involved in zero-shot inference. Let $\mathbf{p}_c$ denote the class prototype used for prediction, which can be either the semantic prototype or the refined prototype depending on the evaluated model.

Class-wise discrepancy terms. To characterize the relation between a prototype and target-domain class distributions, we define the intra-class discrepancy and the nearest inter-class discrepancy for class c as:

$$\Phi_c^+ = \frac{1}{|\mathcal{V}_c|} \sum_{i \in \mathcal{V}_c} d(\mathbf{p}_c, \mathbf{z}_i) + d(\mathbf{p}_c, \boldsymbol{\mu}_c), \quad \Phi_c^- = \min_{c' \neq c} \left(\frac{1}{|\mathcal{V}_{c'}|} \sum_{i \in \mathcal{V}_{c'}} d(\mathbf{p}_c, \mathbf{z}_i) + d(\mathbf{p}_c, \boldsymbol{\mu}_{c'}) \right), \quad (1)$$

where $d(\cdot, \cdot)$ is a non-negative distance, instantiated as $d(\mathbf{a}, \mathbf{b}) = 1 - \cos(\mathbf{a}, \mathbf{b})$ for cosine-based matching. Φ_c^+ measures the intra-class deviation of $\mathbf{p}_c$, while Φ_c^- measures its nearest inter-class separation; better alignment corresponds to lower Φ_c^+ and higher Φ_c^-.

Based on these quantities, we define the Prototype Shift ($\mathcal{PS}$) metric as

$$\mathcal{PS} = \frac{1}{2} \left(\frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{\Phi_c^+ - \Phi_c^-}{\Phi_c^+ + \Phi_c^- + \epsilon} + 1 \right), \quad (2)$$

where $\epsilon > 0$ ensures numerical stability. The ratio measures the trade-off between intra-class deviation and inter-class separation: aligned prototypes yield lower $\mathcal{PS}$, while larger $\mathcal{PS}$ indicates more severe target-domain prototype misalignment.

Theoretical interpretation. $\mathcal{PS}$ admits a direct margin interpretation: since prototype-based classifiers compare samples with class prototypes Snell et al. (2017) and metric learning links reliability to intra-class compactness and inter-class separation Weinberger & Saul (2009), we show that $\mathcal{PS}$ is an inverse affine transformation of the average normalized prototype reliability margin.

Proposition 1. Assume $|\mathcal{C}| > 1$, $d(\cdot, \cdot)$ is non-negative, and $\epsilon > 0$. For each class c, define

$$\Gamma_c = \frac{\Phi_c^- - \Phi_c^+}{\Phi_c^+ + \Phi_c^- + \epsilon}, \quad \mathcal{PS} = \frac{1}{2} \left(1 - \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \Gamma_c \right), \quad \mathcal{PS}_c = \frac{1}{2} \left(\frac{\Phi_c^+ - \Phi_c^-}{\Phi_c^+ + \Phi_c^- + \epsilon} + 1 \right) \quad (3)$$

Moreover, for the class-wise score

we have

$$\frac{\partial \mathcal{PS}_c}{\partial \Phi_c^+} = \frac{2\Phi_c^- + \epsilon}{2(\Phi_c^+ + \Phi_c^- + \epsilon)^2} > 0, \qquad \frac{\partial \mathcal{PS}_c}{\partial \Phi_c^-} = -\frac{2\Phi_c^+ + \epsilon}{2(\Phi_c^+ + \Phi_c^- + \epsilon)^2} < 0. \quad (4)$$

Proof. By the definition of Γ_c, substituting it into Eq. equation 3 gives

$$\Gamma_c = -\frac{\Phi_c^+ - \Phi_c^-}{\Phi_c^+ + \Phi_c^- + \epsilon}, \qquad \mathcal{PS}_c = \frac{1}{2}(1 - \Gamma_c). \quad (5)$$

Averaging over all classes yields Eq. equation 3. Since $d(\cdot, \cdot)$ is non-negative, $\Phi_c^+ \geqslant 0$ and $\Phi_c^- \geqslant 0$. Together with $\epsilon > 0$, the denominator $\Phi_c^+ + \Phi_c^- + \epsilon$ is strictly positive. Direct differentiation of Eq. equation 3 gives Eq. equation 4, completing the proof.

Proposition 1 shows that reducing $\mathcal{PS}$ is equivalent to increasing the average normalized prototype reliability margin. It further indicates that $\mathcal{PS}$ grows with intra-class discrepancy and decreases with nearest inter-class separation, making it a principled diagnostic for prototype reliability under cross-domain transfer. We further analyze the relationship between $\mathcal{PS}$ and transfer performance in Appendix B.

4 METHOD

To address prototype shift, TI-GFM performs adaptive prototype alignment beyond static semantic transfer. **TGP** learns perturbation-robust representations, while **TIP** refines semantic prototypes via cross-view prediction consistency. Figure 2 provides an overview.

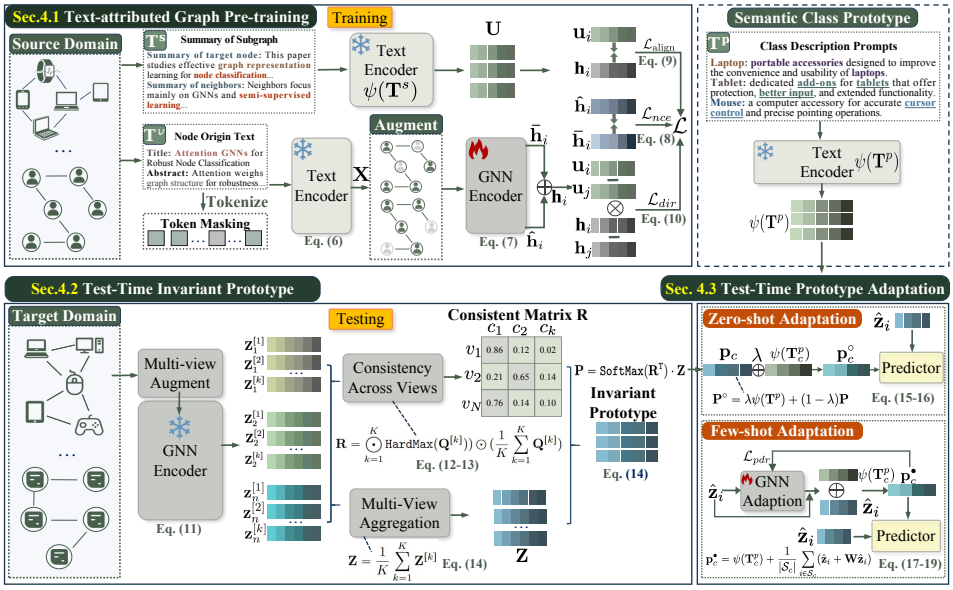


Figure 2: Overview of TI-GFM. TI-GFM learns perturbation-robust graph–text representations via TGP and refines semantic prototypes at test time through TIP construction and adaptation.

4.1 Text-Attributed Graph Pre-training

TGP aims to learn node representations that preserve transferable semantic and structural information across domains. Instead of relying on a single graph–text view, TGP applies textual and structural perturbations and enforces consistency between the resulting representations.

Text-attributed Node Encoding. We first apply stochastic token masking to node texts $\mathbf{T}^v$ and encode the masked texts with the text encoder $\psi(\cdot)$. Following GraphCLIP Zhu et al. (2025b), we construct each subgraph summary by prompting an LLM with the raw texts of all nodes in the corresponding ego-subgraph. The resulting summary $\mathbf{T}^s$ captures high-level semantic context beyond individual node texts and is encoded to provide summary-level semantic supervision:

$$\mathbf{X} = \psi(\texttt{Mask}(\mathbf{T}^v)), \qquad \mathbf{U} = \psi(\mathbf{T}^s) = [\mathbf{u}_1, \cdots, \mathbf{u}_N]^{\mathrm{T}}. \tag{6}$$

Here, $\mathrm{Mask}(\cdot)$ denotes stochastic token masking. This operation reduces dependence on surface-level lexical cues and encourages the encoder to use contextual semantic information.

Graph Augmentation and Encoding.

For each node v_i, we extract its ego-network $\{\mathbf{X}^{(i)}, \mathbf{A}^{(i)}\} = \texttt{SubGraph}(\mathbf{X}, \mathbf{A}, v_i)$, where $\mathbf{X}^{(i)}$ and $\mathbf{A}^{(i)}$ are the local node features and adjacency matrix. We construct a perturbed view by applying stochastic masking to both node features and edges. Specifically, $\texttt{Mask}(\mathbf{X}^{(i)})$ randomly masks the feature vectors of a subset of nodes in the ego-network, while $\texttt{Mask}(\mathbf{A}^{(i)})$ randomly removes a subset of edges by masking the corresponding entries in the adjacency matrix. The clean and perturbed subgraph representations are encoded as:

$$\bar{\mathbf{h}}_i = \texttt{GNN}(\mathbf{X}^{(i)}, \mathbf{A}^{(i)}), \qquad \hat{\mathbf{h}}_i = \texttt{GNN}\left(\texttt{Mask}(\mathbf{X}^{(i)}), \texttt{Mask}(\mathbf{A}^{(i)})\right), \qquad \mathbf{h}_i = \frac{1}{2}(\bar{\mathbf{h}}_i + \hat{\mathbf{h}}_i). \tag{7}$$

These perturbations simulate local structural and feature variations, forcing the encoder to preserve view-consistent information.

Graph Contrastive Loss. We enforce cross-view consistency through an InfoNCE-style objective. For each node, the clean and perturbed representations form a positive pair, while perturbed representations from other nodes are used as negatives:

$$\mathcal{L}_{nce} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\langle \bar{\mathbf{h}}_i, \hat{\mathbf{h}}_i \rangle / \tau\right)}{\sum_{j=1}^{N} \exp\left(\langle \bar{\mathbf{h}}_i, \hat{\mathbf{h}}_j \rangle / \tau\right)}. \tag{8}$$

This loss encourages representations of the same node to remain close under perturbations and separates different nodes in the representation space.

Graph-Summary Alignment Loss. Since perturbation consistency alone does not align graph embeddings with high-level textual semantics, we align each subgraph representation $\mathbf{h}_i$ with its summary embedding $\mathbf{u}_i$ via a symmetric graph-summary contrastive loss:

$$\mathcal{L}_{align} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\frac{\exp(\langle \mathbf{h}_i, \mathbf{u}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathbf{h}_i, \mathbf{u}_j\rangle/\tau)} + \log\frac{\exp(\langle \mathbf{u}_i, \mathbf{h}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathbf{u}_i, \mathbf{h}_j\rangle/\tau)}\right). \tag{9}$$

This objective anchors graph representations to summary-level semantic information and improves class-level coherence across domains.

Graph-Summary Direction Loss. To preserve local relational geometry, we enforce directional consistency between graph embeddings and summary-text embeddings over semantic neighbors:

$$\mathcal{L}_{dir} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\left(1-\frac{(\mathbf{h}_i-\mathbf{h}_j)^\top(\mathbf{u}_i-\mathbf{u}_j)}{\|\mathbf{h}_i-\mathbf{h}_j\|\,\|\mathbf{u}_i-\mathbf{u}_j)\|}\right), \quad \mathcal{N}_i = \texttt{Top-}r_{j\neq i}\left(\frac{\mathbf{u}_i^\top\mathbf{u}_j}{\|\mathbf{u}_i\|\|\mathbf{u}_j\|}\right) \tag{10}$$

where $\mathcal{N}_i$ denotes the top-r semantic neighbors of node i in the summary-text space. This loss aligns local relative directions between the graph and the summary-text embedding spaces.

The overall TGP objective is $\mathcal{L}_{TGP} = \alpha\mathcal{L}_{nce} + \beta\mathcal{L}_{align} + \mathcal{L}_{dir}$, where $\alpha > 0$ and $\beta > 0$ balance different loss components for stable optimization. Together, these objectives learn semantically grounded representations robust to graph perturbations for subsequent prototype refinement.

4.2 Test-Time Invariant Prototype Construction

Although graph–text alignment methods Zhu et al. (2025b); Li et al. (2024) enable semantic prototype matching, fixed class prototypes may shift under cross-domain transfer. TIP alleviates this issue by extracting invariant target-domain evidence from perturbed graph views with a frozen backbone.

Multi-view Target Encoding. For each target node v_i, we generate K perturbed views by independently masking node features and edges with $\texttt{Mask}(\cdot, k)$. Each view is encoded by the frozen encoder $\texttt{GNN}^{\circledast}(\cdot)$, yielding the view-specific target representations:

$$\mathbf{Z}^{[k]} = [\mathbf{z}_1^{[k]}, \cdots, \mathbf{z}_N^{[k]}]^\intercal, \quad \mathbf{z}_i^{[k]} = \texttt{GNN}^{\circledast}\left(\texttt{Mask}(\mathbf{X}^{(i)}, k), \texttt{Mask}(\mathbf{A}^{(i)}, k)\right), \quad k = 1, \ldots, K. \tag{11}$$

Prediction Consistency Across Views. Let $\psi(\mathbf{T}^p)$ denote the semantic class prototypes encoded from the class description prompts. For each view, we compute node-class similarity scores $\mathbf{Q}^{[k]}$:

$$\mathbf{Q}^{[k]} = \texttt{SoftMax}(\mathbf{Z}^{[k]} \cdot \psi(\mathbf{T}^p)^\intercal), \tag{12}$$

where $\texttt{SoftMax}(\cdot)$ refers to the row-wise $\texttt{SoftMax}$ function. We then keep only node-class assignments whose hard predictions are consistent across all views:

$$\mathbf{R} = \left(\bigodot_{k=1}^{K}\texttt{HardMax}(\mathbf{Q}^{[k]})\right) \odot \left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{Q}^{[k]}\right) = \begin{array}{c} \\ v_1 \\ v_2 \\ \vdots \\ v_N \end{array}\begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_{|\mathcal{C}|} \end{array} \\ \begin{bmatrix} 0 & 0.9 & \cdots & 0 \\ 0.8 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0.7 \end{bmatrix} \end{array} \tag{13}$$

where $\texttt{HardMax}(\cdot)$ converts each row into a one-hot vector by selecting the largest class probability. The element-wise product across views retains only perturbation-consistent assignments, and $\bar{\mathbf{Q}}$ provides confidence-aware weights. Thus, $\mathbf{R}$ encodes target nodes that provide invariant evidence for each class.

Multi-view Invariant Prototype. We aggregate cross-view consistent target-domain evidence into class-wise invariant prototypes:

$$\mathbf{P} = \texttt{SoftMax}(\mathbf{R}^\intercal) \cdot \mathbf{Z} = [\mathbf{p}_1, \cdots, \mathbf{p}_C]^\intercal, \quad \mathbf{Z} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{Z}^{[k]}, \tag{14}$$

where $\mathbf{Z}$ is the averaged target-node representation matrix over K views. $\texttt{SoftMax}(\mathbf{R}^{\mathrm{T}})$ normalizes the class-wise weights, so that each prototype $\mathbf{p}_c$ aggregates target nodes with invariant predictions and high confidence. Thus, $\mathbf{P}$ incorporates target-domain invariant evidence and better aligns with the target graph than static semantic prototypes.

4.3 Test-Time Invariant Prototype Adaptation

Zero-shot Adaptation. In zero-shot transfer, no target labels are available. We therefore combine semantic class prototypes with the invariant target-domain prototypes from Eq. equation 14:

$$\mathbf{P}^{\circ} = \lambda\psi(\mathbf{T}^{p}) + (1-\lambda)\mathbf{P} = [\mathbf{p}_1^{\circ}, \cdots, \mathbf{p}_C^{\circ}]^{\mathrm{T}}, \tag{15}$$

where $\lambda \in [0,1]$ balances semantic class priors and target-domain invariant evidence. The final prediction for target node v_i is

$$\hat{y}_i = \texttt{Predictor}(\hat{\mathbf{z}}_i, \mathbf{p}_c^{\circ}), \qquad \hat{\mathbf{z}}_i = \texttt{GNN}^{\circledast}(\mathbf{X}^{(i)}, \mathbf{A}^{(i)}), \tag{16}$$

where $\texttt{Predictor}(\cdot)$ assigns node v_i to the class whose refined prototype has the highest similarity to $\hat{\mathbf{z}}_i$. This prediction rule uses prototypes that remain semantically grounded while being calibrated by target-domain invariant evidence.

Few-shot Adaptation. In the few-shot setting, given a small support set $\mathcal{S}_c$ for each class c, we freeze the backbone and adapt prototypes with support-node evidence via:

$$\mathbf{p}_c^{\bullet} = \psi(\mathbf{T}_c^{p}) + \frac{1}{|\mathcal{S}_c|}\sum_{i\in\mathcal{S}_c}(\hat{\mathbf{z}}_i + \mathbf{W}\hat{\mathbf{z}}_i), \tag{17}$$

where $\mathbf{W}$ is a learnable graph residual adapter, $\psi(\mathbf{T}_c^{p})$ provides the semantic class prior, and support representations shift the prototype toward the target-domain class distribution.

Let $\mathcal{S} = \bigcup_{c\in\mathcal{C}}\mathcal{S}_c$ be all support nodes. We optimize the residual adapter and adapted prototypes by:

$$\mathcal{L}_{pdr} = \underbrace{-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\log\frac{\exp(\langle\hat{\mathbf{z}}_i, \mathbf{p}_{y_i}^{\bullet}\rangle/\tau)}{\sum_{c=1}^{C}\exp(\langle\hat{\mathbf{z}}_i, \mathbf{p}_c^{\bullet}\rangle/\tau)}}_{\text{discrimination}} + \underbrace{\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\|\hat{\mathbf{z}}_i - \mathbf{p}_{y_i}^{\bullet}\|_2}_{\text{reconstruction}}. \tag{18}$$

The discrimination and reconstruction terms enhance class separation and prototype compactness. During inference, target nodes are classified by matching frozen representations with adapted prototypes.

$$\hat{y}_i = \texttt{Predictor}(\hat{\mathbf{z}}_i, \mathbf{p}_c^{\bullet}). \tag{19}$$

This supervised adaptation uses limited target labels to improve prototype alignment in the low-shot regime.

5 Experiments

5.1 Experimental Setup

Dataset configuration.

Following GraphCLIP Zhu et al. (2025b), we create a deliberately shifted source/target split. The training set is $\mathcal{D}_{\text{train}} = \{$*ogbn-ArXiv*, *ArXiv_2023*, *PubMed*, *ogbn-Products*, *Reddit*$\}$, while the zero-shot test set is $\mathcal{D}_{\text{test}} = \{$*Cora*, *CiteSeer*, *Photo*, *Computers*, *History*, *WikiCS*, *Instagram*$\}$.

Baselines. We compare TI-GFM with 18 baselines from three categories: semantic-only methods (BERT Devlin et al. (2019), SBERT Reimers & Gurevych (2019), DeBERTa He et al. (2020), E5 Wang et al. (2022), Qwen2-Instruct Yang et al. (2024a), and LLaMA3.1-Instruct Sam (2024)); graph self-supervised methods (DGI Veličković et al. (2019), GRACE Zhu et al. (2020), BGRL Thakoor et al. (2021), GraphMAE Hou et al. (2022), and G2P2 Wen & Fang (2023)); and TAG/GFM methods (GraphGPT Tang et al. (2024), LLaGA Chen et al. (2024a), OFA Liu et al. (2023a), ZeroG Li et al. (2024), and GraphCLIP Zhu et al. (2025b)). For few-shot node classification, we additionally compare with GPPT Sun et al. (2022), GraphPrompt Liu et al. (2023c), All-in-One Sun et al. (2023), GPF Fang et al. (2022), and GraphCLIP under 1-, 3-, 5-, and 10-shot settings.

Table 1: Zero-shot node classification accuracy (%) with rank analysis. The best, second, and third results are marked in **first**, second, and third. Baseline results are from GraphCLIP Zhu et al. (2025b).

Methods	CiteSeer	Computers	Photo	Cora	History	WikiCS	Instagram	**Avg Rank**
LLM-only Methods								
BERT	33.26±2.35	13.88±0.29	21.80±0.14	19.56±0.98	9.95±0.42	29.37±0.00	57.02±0.57	13.43
SBERT	50.47±0.90	41.82±0.22	35.96±0.44	54.35±1.26	30.45±0.19	48.16±0.00	48.34±1.23	8.29
DeBERTa	16.42±1.26	10.62±0.15	12.38±0.26	16.42±1.26	8.70±0.26	15.29±0.00	39.81±0.58	17.14
E5	42.57±0.54	16.54±0.14	35.14±0.28	44.65±0.82	12.92±0.48	31.49±0.00	61.28±0.97	10.71
Qwen2 (7B)	53.57±0.86	59.18±0.20	45.55±0.12	61.44±1.29	23.79±0.34	58.72±0.25	39.13±0.78	7.57
Qwen2 (72B)	60.97±0.87	60.88±0.30	52.41±0.39	62.18±0.98	53.56±0.64	60.91±0.08	47.70±0.31	5.00
LLaMA3.1 (8B)	53.54±1.71	46.98±0.21	34.38±0.25	57.75±1.21	22.28±0.18	58.32±0.21	39.37±1.14	8.86
LLaMA3.1 (70B)	62.79±1.24	61.62±0.42	51.26±0.53	65.72±1.24	53.33±0.55	62.82±0.04	43.68±0.52	4.86
Traditional Self-supervised Methods								
DGI	18.71±1.22	27.12±0.03	13.96±0.17	24.03±1.40	15.77±0.02	18.86±0.25	61.42±1.12	12.14
GRACE	22.88±1.49	10.94±0.12	10.16±0.13	13.69±1.27	32.39±0.11	16.07±0.32	62.23±0.93	12.71
BGRL	26.50±1.22	24.12±0.22	5.21±0.22	20.80±1.06	16.28±0.35	18.35±0.22	61.45±0.82	13.00
GraphMAE	20.75±0.88	8.36±0.06	12.53±0.08	23.25±1.07	21.76±0.17	12.14±0.20	**62.39±0.84**	12.93
G2P2	51.02±0.62	32.52±0.13	22.21±0.12	41.51±0.78	26.18±0.25	31.92±0.15	52.87±0.78	9.29
TAG Methods								
GraphGPT	18.04±1.45	29.71±0.83	7.62±0.22	23.25±1.45	15.92±0.14	6.30±0.26	45.12±1.16	14.65
LLaGA	16.07±1.15	23.10±0.33	6.50±0.53	21.44±0.65	11.17±0.58	2.65±0.72	41.12±0.94	16.57
OFA	29.64±0.19	22.09±0.39	33.03±0.64	37.25±1.38	16.87±0.93	45.52±1.06	32.71±0.16	12.43
ZeroG	52.55±1.23	56.20±0.35	45.12±0.65	62.32±1.91	40.74±0.65	54.93±0.06	48.97±0.78	6.43
GraphCLIP	63.13±1.13	62.04±0.21	53.40±0.64	67.31±1.76	53.88±0.35	70.19±0.10	60.69±0.34	2.71
Our	**71.50±0.16**	**71.68±0.09**	**57.45±1.43**	**71.48±2.45**	**57.16±1.71**	**72.04±0.35**	61.61±0.73	**1.29**
Δ	**8.37↑**	**9.64↑**	**4.05↑**	**4.17↑**	**3.28↑**	**1.85↑**	**0.78↓**	-

Evaluation protocol. We report ACC with standard deviations over five random seeds. All methods are trained on $\mathcal{D}_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{test}}$. Zero-shot transfer uses no target labels or tuning, while few-shot transfer uses a small labeled support set. We adopt MiniLM Wang et al. (2020b) and GraphGPS Rampášek et al. (2022) as text and graph encoders, respectively. Hyperparameters are searched on training datasets and fixed for targets, with $r = 3$ and $\lambda = 0.3$. More details are in Appendix H.

5.2 Overall Performance

Zero-shot node classification. Table 1 reports zero-shot node classification results on seven target datasets. LLM-only methods perform competitively on some semantic-rich datasets but show unstable cross-domain transfer, while traditional graph self-supervised methods suffer from the lack of semantic interfaces for unseen label spaces. Graph–text methods such as ZeroG and GraphCLIP achieve stronger transfer through semantic alignment, but their fixed prototypes may remain misaligned with target-domain class structures. In contrast, TI-GFM achieves the best average rank of 1.29 and the best accuracy on six of seven datasets, improving over the strongest baseline by 8.37% on CiteSeer, 9.64% on Computers, 4.05% on Photo, 4.17% on Cora, 3.28% on History, and 1.85% on WikiCS. On Instagram, TI-GFM ranks third with a gap of 0.78%, indicating the challenge of weak semantic signals in this dataset. Overall, these results demonstrate that perturbation-robust representation learning and test-time prototype refinement improve cross-domain zero-shot transfer. Appendix I.1 further reports zero-shot link prediction results.

Few-shot node classification. Figure 3 reports few-shot node classification results under different k-shot settings. While all methods benefit from more support nodes, gaps remain in low-shot regimes. GraphCLIP is the strongest baseline, but its semantic prototypes may still suffer from target-domain misalignment. TI-GFM consistently outperforms all baselines on WikiCS and History, improving over GraphCLIP on History by 8.08%, 14.46%, 12.06%, and 4.53% under 1-, 3-, 5-, and 10-shot settings, respectively. These results demonstrate improved prototype alignment with limited target-domain supervision.

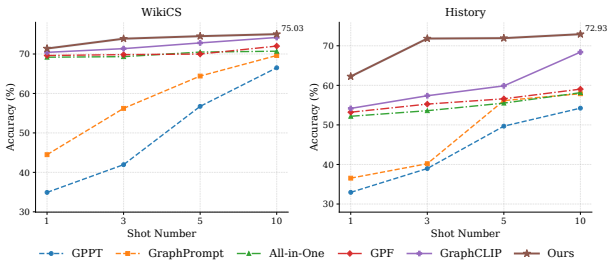


Figure 3: Few-shot node classification accuracy under different k-shot settings.

5.3 Ablation Study

Table 2(a) evaluates TGP and TIP, with Backbone removing both modules. **(1) TIP improves prototype alignment.** +TIP improves Cora (+6.24), Photo (+2.67), History (+6.70), and Instagram (+1.29), showing that invariant prototype construction mitigates semantic prototype misalignment. **(2) TGP improves representation robustness.** +TGP improves most datasets, including WikiCS (+6.39), Computers (+2.56), and History (+3.71), indicating stronger cross-domain stability but limited correction of prototype shift. **(3) TGP and TIP are complementary.** The full model performs best on six of seven datasets, outperforming +TIP by 4.08% on WikiCS and +TGP by 5.99% on History, confirming their complementary roles.

Table 2: Ablation and prototype-shift diagnostics. Left: zero-shot node classification accuracy (%) of different variants. Right: Prototype Shift ($\mathcal{PS}\downarrow$) comparison among ablated variants and GraphCLIP.

(a) Ablation study.

Data	Backbone	+TIP	+TGP	TI-GFM
Cora	64.94	71.18	65.87	**71.48**
CiteSeer	71.16	70.82	71.32	**71.50**
WikiCS	64.23	67.96	70.62	**72.04**
Computers	68.23	67.76	70.79	**71.68**
Photo	54.36	57.03	55.00	**57.45**
History	47.46	54.16	51.17	**57.16**
Instagram	60.08	61.37	**61.81**	61.61

(b) Prototype Shift analysis.

Method	Cora	CiteSeer	WikiCS	Computers	Photo	History
Backbone	0.43	0.43	0.46	0.46	0.46	0.47
+TGP	0.42	0.43	0.45	0.45	0.45	0.47
+TIP	0.39	0.38	0.43	0.43	0.43	0.46
GraphCLIP	0.42	0.43	0.44	0.45	0.45	0.46
TI-GFM	**0.36**	**0.36**	**0.41**	**0.41**	**0.41**	**0.45**

5.4 Impact of $\mathcal{PS}$

Table 2(b) reports Prototype Shift ($\mathcal{PS}$) for different variants, where lower values indicate better target-domain prototype alignment. **(1) TIP mainly reduces $\mathcal{PS}$.** Compared with Backbone, +TIP consistently lowers $\mathcal{PS}$ on all reported datasets, e.g., Cora (0.43 → 0.39), CiteSeer (0.43 → 0.38), and Photo (0.46 → 0.43), validating perturbation-consistent target nodes for test-time refinement. **(2) TGP has smaller positive effects.** +TGP moderately reduces $\mathcal{PS}$ on Cora, WikiCS, Computers, and Photo, suggesting that robust pre-training improves representation stability but cannot fully calibrate prototypes. **(3) The full model performs best.** Compared with GraphCLIP, TI-GFM achieves lower $\mathcal{PS}$ on all reported datasets, including Cora (0.36 vs. 0.42), CiteSeer (0.36 vs. 0.43), and WikiCS (0.41 vs. 0.44). These trends align with Table 1, supporting the link between lower prototype shift and better cross-domain transfer.

6 Limitations

TI-GFM has three main limitations. **(1) Text dependence.** The framework relies on node texts, subgraph summaries, and class descriptions to build graph–text representations and semantic prototypes. Noisy or label-irrelevant texts may weaken the semantic prior, despite the robustness improved by TGP. **(2) Test-time cost and sensitivity.** TIP requires multiple perturbed views and repeated forward passes through the frozen encoder, whose cost is moderate in our experiments but may increase on larger graphs. Moreover, the fixed values of K, masking ratios, and λ may not be optimal for every target domain. **(3) Post-hoc nature of $\mathcal{PS}$.** $\mathcal{PS}$ requires target labels to estimate empirical class distributions, and thus only serves as a diagnostic metric rather than an inference objective. Label-free proxies for prototype shift remain future work.

7 Conclusion

This paper studied prototype shift in cross-domain text-attributed GFMs. We introduced Prototype Shift ($\mathcal{PS}$) to measure whether semantic prototypes remain close to target-domain class distributions and separated from confusing classes, and linked lower $\mathcal{PS}$ to a larger normalized prototype reliability margin. Based on this diagnosis, we proposed TI-GFM, a test-time invariant prototype learning framework that combines Text-Attributed Graph Pre-training (TGP) with Test-Time Invariant Prototype (TIP) refinement. Experiments show that TI-GFM achieves the best average rank and the best accuracy on six of seven zero-shot node classification benchmarks, while also improving few-shot node classification and zero-shot link prediction. Ablation and $\mathcal{PS}$ analyses show that TGP improves representation robustness and TIP reduces prototype misalignment, highlighting prototype alignment as an effective objective for cross-domain transfer in text-attributed GFMs.

REFERENCES

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

William Brannon, Wonjune Kang, Suyash Fulay, Hang Jiang, Brandon Roy, Deb Roy, and Jad Kabbara. Congrat: Self-supervised contrastive pretraining for joint graph and text embeddings. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pp. 19–39, 2024.

Haibo Chen, Xin Wang, Zeyang Zhang, Haoyang Li, Weigao Wen, Ling Feng, and Wenwu Zhu. Curriculum gnn-llm alignment for text-attributed graphs. 2025.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*, 2023.

Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, and Jiliang Tang. Text-space graph foundation models: Comprehensive benchmarks and new insights. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL `https://openreview.net/forum?id=Q2sDuwtutB`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Taoran Fang, Yunchao Mercer Zhang, Yang Yang, and Chunping Wang. Prompt tuning for graph neural networks. 2022.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 747–758, 2024.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: encoding graphs for large language models. arxiv. *arXiv preprint arXiv:2310.04560*, 2023.

C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, 1998.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Bryan Hooi, et al. Explanations as features: Llm-based features for text-attributed graphs. *arXiv preprint arXiv:2305.19523*, 2(4):8, 2023.

Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 448–459, 2025.

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, 2022.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Can gnn be good adapter for llms? In *Proceedings of the ACM web conference 2024*, pp. 893–904, 2024.

Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. Patton: Language model pretraining on text-rich networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7005–7020, 2023.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):8622–8642, 2024.

Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan Zhang. GOFA: A generative one-for-all model for joint graph language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mIjblC9hfm.

Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1725–1735, 2024.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. Bertgcn: Transductive text classification by combining gnn and bert. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1456–1462, 2021.

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. *arXiv preprint arXiv:2310.00149*, 2023a.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Towards graph foundation models: A survey and beyond. corr abs/2310.11829 (2023). *arXiv preprint arXiv:2310.11829*, 2023b.

Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

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, 2023c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Péter Mernyei and Cătălina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 188–197, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Kira Sam. Llama 3.1: An in-depth analysis of the next-generation large language model. *Available at SSRN 6139407*, 2024.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Guangxin Su, Hanchen Wang, Jianwei Wang, Wenjie Zhang, Ying Zhang, and Jian Pei. Large language models meet text-attributed graphs: A survey of integration frameworks and applications. *arXiv preprint arXiv:2510.21131*, 2025.

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, 2022.

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, 2023.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 491–500, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657775. URL https://doi.org/10.1145/3626772.3657775.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, pp. 1–14. OpenReview. net, 2021.

Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1722–1731, 2017.

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. URL https://openreview.net/forum?id=rklz9iAcKQ.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Advances in neural information processing systems*, 37:5950–5973, 2024.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1): 396–413, 2020a.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017*, 2025a.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020b.

Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. Can llms convert graphs to text-attributed graphs? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1412–1432, 2025b.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, et al. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025c.

Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(9):207–244, 2009. URL http://jmlr.org/papers/v10/weinberger09a.html.

Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 506–516, 2023.

Ruiyao Xu and Kaize Ding. Gnn-as-judge: Unleashing the power of llms for graph semi-supervised learning with gnn feedback. In *Machine Learning on Graphs in the Era of Generative Artificial Intelligence*, 2025.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation learning on textual graph. *Advances in Neural Information Processing Systems*, 34:28798–28810, 2021.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a. URL https://api.semanticscholar.org/CorpusID:274859421.

Rui Yang, Jiahao Zhu, Jianping Man, Li Fang, and Yi Zhou. Enhancing text-based knowledge graph completion with zero-shot large language models: A focus on semantic enhancement. *Knowledge-Based Systems*, 300:112155, 2024b.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

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, 2025.

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 *Forty-second International Conference on Machine Learning*, 2025.

Zihao Zhang, Xunkai Li, Rong-Hua Li, Bing Zhou, Zhenjun Li, and Guoren Wang. Toward general and robust llm-enhanced text-attributed graph learning. *arXiv preprint arXiv:2504.02343*, 2025.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022.

Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael M. Bronstein, Zhaocheng Zhu, and Jian Tang. Graphtext: Graph reasoning in text space. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024. URL https://openreview.net/forum?id=6cRM0OT03F.

Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, Imran Razzak, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025a.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep Graph Contrastive Representation Learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020. URL http://arxiv.org/abs/2006.04131.

Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pp. 2183–2197, 2025b.

A Appendix

B Experimental Analysis of Prototype Shift $\mathcal{PS}$

To further examine the role of Prototype Shift ($\mathcal{PS}$) in cross-domain transfer, we analyze its effect from both quantitative and qualitative perspectives. Overall, three consistent patterns emerge.

(1) Reducing prototype shift is generally accompanied by larger transfer gains. As reported in Table 3, $\Delta\mathcal{PS}$ denotes the reduction in Prototype Shift achieved by TI-GFM relative to GraphCLIP, while ΔACC denotes the corresponding gain in classification accuracy. Across target datasets, larger reductions in $\mathcal{PS}$ are generally associated with larger improvements in downstream performance. Quantitatively, the Pearson correlation reaches r = 0.7225, indicating a positive association between prototype shift reduction and transfer gain. This result suggests that prototype quality is closely associated with cross-domain generalization performance.

(2) The gains of TI-GFM are consistently reflected in more representative prototype placement. Figure 4 visualizes the node embedding and class prototype distributions on four representative target datasets. Compared with GraphCLIP, TI-GFM consistently achieves lower $\mathcal{PS}$ values, reducing it from 0.43 to 0.36 on CiteSeer, from 0.42 to 0.36 on Cora, from 0.44 to 0.41 on WikiCS, and from 0.45 to 0.41 on Photo. A common pattern can be observed across these datasets: the prototypes produced by TI-GFM are located closer to the high-density regions of the corresponding node distributions, while the prototype-to-cluster mismatch is visibly reduced. This indicates that TI-GFM improves the representativeness of transferred prototypes in the target domain.

(3) The benefit is more pronounced when prototype displacement is more severe. As shown in Figure 4, the visual improvements are especially evident on CiteSeer and Photo, where GraphCLIP exhibits relatively clear prototype offset from the dominant class clusters, and TI-GFM moves the prototypes noticeably closer to the geometric centers of these distributions. On more challenging datasets such as WikiCS, where class overlap remains substantial, the improvement is less reflected in perfect class separation and more in reduced prototype displacement and more stable prototype placement. Therefore, the role of TI-GFM is not necessarily to eliminate all class overlap, but to make the transferred prototypes better aligned with the intrinsic target-domain structure. Overall, the visual evidence in Figure 4 is highly consistent with the quantitative $\mathcal{PS}$ results, further supporting the claim that prototype misalignment is a key factor limiting cross-domain generalization.

Table 3: Prototype Shift ($\mathcal{PS}\downarrow$) comparison among ablated variants and GraphCLIP, demonstrating the effectiveness of TGP and TIP in improving cross-domain prototype alignment.

Method	Cora	CiteSeer	WikiCS	Computers	Photo	History	Instagram
Backbone	0.43	0.43	0.46	0.46	0.46	0.47	0.49
+TGP	0.42	0.43	0.45	0.45	0.45	0.47	0.50
+TIP	0.39	0.38	0.43	0.43	0.43	0.46	0.49
GraphCLIP	0.42	0.43	0.44	0.45	0.45	0.46	**0.49**
TI-GFM	**0.36**	**0.36**	**0.41**	**0.41**	**0.41**	**0.45**	0.50
$\Delta\mathcal{PS}$	+0.06	+0.07	+0.03	+0.04	+0.04	+0.01	-0.01
Δ ACC	+4.17	+8.37	+1.85	+9.64	+4.05	+3.28	-0.78

C Time Complexity Analysis

Pre-training Phase. Let N be the number of training nodes, l be the text sequence length, d be the hidden dimension, B be the mini-batch size, L be the number of GNN layers, s be the maximum ego-subgraph size, and e_s be the maximum number of edges in an ego-subgraph. The summary embeddings $\mathbf{U} = \psi(\mathbf{T}^s)$ are pre-computed once and reused during training. The pre-training complexity of TGP consists of the following components:

- **Textual encoding:** encoding masked node texts $\texttt{Mask}(\mathbf{T}^v)$ with the text encoder $\psi(\cdot)$ costs $\mathcal{O}(Nl^2d)$ per epoch.

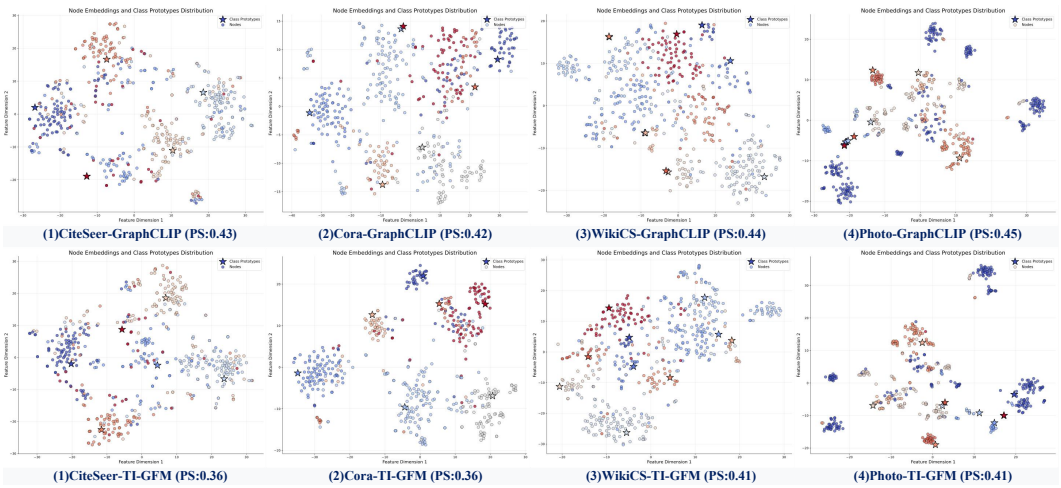


Figure 4: Visualization of node embeddings and class prototypes under GraphCLIP and TI-GFM on representative target datasets.

- **Graph augmentation and encoding:** for each node, TGP encodes both the clean ego-subgraph and its perturbed view with an L-layer GNN, leading to $\mathcal{O}(NL(e_s d + sd^2))$ per epoch, where the constant factor of two is omitted.
- **Graph contrastive loss:** computing pairwise similarities in $\mathcal{L}_{nce}$ costs $\mathcal{O}(B^2 d)$ per mini-batch and $\mathcal{O}(NBd)$ per epoch.
- **Graph-summary alignment loss:** computing graph-summary similarities in $\mathcal{L}_{align}$ costs $\mathcal{O}(B^2 d)$ per mini-batch and $\mathcal{O}(NBd)$ per epoch.
- **Directional consistency:** computing semantic-neighbor similarities and enforcing local directional consistency costs $\mathcal{O}(B^2 d)$ per mini-batch and $\mathcal{O}(NBd)$ per epoch.

Thus, the per-epoch pre-training complexity is:

$$\mathcal{O}\left(Nl^2 d + NL(e_s d + sd^2) + NBd\right). \tag{20}$$

For T pre-training epochs, the total complexity is:

$$\mathcal{O}\left(T\left(Nl^2 d + NL(e_s d + sd^2) + NBd\right)\right), \tag{21}$$

excluding the one-time pre-computation cost of summary embeddings. When ego-subgraphs are treated as dense graphs or the encoder uses dense attention within each ego-subgraph, e_s can be upper-bounded by s^2, yielding the graph encoding term $\mathcal{O}(NL(s^2 d + sd^2))$. With fixed B, s, and e_s, the pre-training complexity scales approximately linearly with N.

In practice, the dominant cost arises from textual encoding and subgraph-based GNN computation. Since $s \ll N$ and $B \ll N$, the overall complexity scales approximately linearly with the number of nodes. In practice, the pre-training process takes approximately 9 hours using two NVIDIA A40 GPUs with 48GB of memory.

Test-Time Adaptation Phase. Let N_t be the number of target nodes, C be the number of classes, K be the number of perturbed views, and e_t be the maximum number of edges in a target ego-subgraph. The semantic class prototypes $\psi(\mathbf{T}^p)$ are pre-computed once and reused during inference. The test-time complexity of TIP consists of the following components:

- **Multi-view target encoding:** for each target node, K perturbed ego-subgraph views are encoded by the frozen GNN, resulting in $\mathcal{O}(N_t K L(e_t d + sd^2))$.
- **Cross-view prediction consistency:** computing $\mathbf{Q}^{[k]} = \texttt{SoftMax}(\mathbf{Z}^{[k]}\psi(\mathbf{T}^p)^\top)$ for K views costs $\mathcal{O}(N_t K C d)$. Constructing $\mathbf{R}$ through hard prediction consistency and confidence averaging costs $\mathcal{O}(N_t K C)$, which is dominated by the similarity computation.

- **Invariant prototype aggregation:** averaging multi-view representations costs $\mathcal{O}(N_t K d)$, and computing $\mathbf{P} = \texttt{SoftMax}(\mathbf{R}^\top)\mathbf{Z}$ costs $\mathcal{O}(N_t C d)$.
- **Final prediction:** encoding the clean target ego-subgraph and matching each node with refined prototypes costs $\mathcal{O}(N_t L(e_t d + s d^2) + N_t C d)$.

Therefore, the total zero-shot test-time complexity is:

$$\mathcal{O}\left(N_t(K+1)L(e_t d + s d^2) + N_t K C d + N_t C d\right). \tag{22}$$

The dominant cost comes from multi-view GNN encoding. Since no backpropagation is required and K, C, s, and e_t are fixed in practice, TIP scales approximately linearly with the number of target nodes.

Few-shot Adaptation Phase. In the few-shot setting, let $|\mathcal{S}|$ be the number of support nodes and T_a be the number of adapter optimization epochs. Since the backbone encoder is frozen, support-node representations are computed once with complexity $\mathcal{O}(|\mathcal{S}|L(e_t d + s d^2))$. For each adaptation epoch, applying the residual adapter $\mathbf{W}$ costs $\mathcal{O}(|\mathcal{S}|d^2)$. Computing the discrimination term over C prototypes costs $\mathcal{O}(|\mathcal{S}|Cd)$, while the reconstruction term costs $\mathcal{O}(|\mathcal{S}|d)$. Thus, the few-shot adaptation complexity is:

$$\mathcal{O}\left(|\mathcal{S}|L(e_t d + s d^2) + T_a\left(|\mathcal{S}|d^2 + |\mathcal{S}|Cd\right)\right). \tag{23}$$

After adaptation, predicting all target nodes by matching their frozen representations with adapted prototypes costs $\mathcal{O}(N_t C d)$, assuming target-node representations have been pre-computed. Since $|\mathcal{S}| \ll N_t$, the few-shot adaptation overhead is lightweight compared with full target-graph encoding.

The dominant cost arises from multi-view GNN encoding. Since K, s, and C are typically small, and $|\mathcal{S}| \ll N$, the overall complexity scales approximately linearly with the number of nodes. Notably, for zero-shot TIP, no backpropagation is required because the backbone is frozen and prototypes are constructed by forward aggregation. In few-shot adaptation, backpropagation is only applied to the lightweight residual adapter, while the backbone remains frozen. In practice, the final prediction stage across all seven datasets takes approximately 43 seconds on a single NVIDIA A40 GPU with 48GB of memory. This timing excludes prototype construction and does not represent the end-to-end inference cost.

D Algorithmic Overview

We present the two core procedures of TI-GFM in algorithmic form. Algorithm 1 summarizes the pre-training stage based on Text-Attributed Graph Pre-training (TGP), while Algorithm 2 summarizes the downstream stage based on the Test-time Invariant Prototype (TIP) mechanism for prototype refinement and prediction. This algorithmic view highlights the full pipeline of TI-GFM, from robust cross-domain representation learning to target-domain prototype alignment.

Algorithm 1: Text-Attributed Graph Pre-training (TGP)

Input: Training graph $\mathcal{G} = (\mathbf{T}^v, \mathbf{A})$, textual summaries $\mathbf{T}^s$
Parameters : Epoch number T; GNN layers L; neighbor size r
Initialize parameters of $\psi(\cdot)$ and $\texttt{GNN}(\cdot)$
for each epoch $t = 1, 2, \ldots, T$ **do**
  Apply textual masking to $\mathbf{T}^v$ to obtain node features $\mathbf{X}$ and subgraph summary representations $\mathbf{U}$ via Eq. (6).
  for each node v_i **do**
    Extract ego-subgraph $(\mathbf{X}_i, \mathbf{A}_i)$ via $\texttt{SubGraph}(\cdot)$
    Generate masked graph view using $\texttt{Mask}(\cdot, \cdot)$
    Compute node embedding $\mathbf{h}_i$ via Eq. (7)
  end
  Compute graph contrastive loss $\mathcal{L}_{nce}$ via Eq. (8)
  Compute graph-summary alignment loss $\mathcal{L}_{align}$ via Eq. (9)
  Retrieve top-r neighbors $\mathcal{N}_i$ for each node.
  Compute directional consistency loss $\mathcal{L}_{dir}$ via Eq. (10)
  Compute total loss $\mathcal{L}_{TGP} = \alpha\mathcal{L}_{nce} + \beta\mathcal{L}_{align} + \mathcal{L}_{dir}$
  Update model parameters via gradient descent
end

Algorithm 2: Test-Time Invariant Prototype (TIP) for Construction and Adaptation

Input: Target graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, class texts $\mathbf{T}^p$
Parameters : Frozen encoder $\texttt{GNN}^{\circledast}$; view number K
Compute semantic prototypes $\psi(\mathbf{T}^p)$
for each node v_i **do**
  for $k = 1, 2, \ldots, K$ **do**
    Generate masked views,
    Compute representation $\mathbf{z}_i^{[k]}$ via $\texttt{GNN}^{\circledast}$
  end
end
Construct $\mathbf{Z}^{[k]}$ and compute similarity matrices $\mathbf{Q}^{[k]}$ via Eq. (12)
Compute invariant indicator $\mathbf{R}$ via cross-view consistency Eq. (13)
Aggregate multi-view embeddings $\mathbf{Z} = \frac{1}{K}\sum_k \mathbf{Z}^{[k]}$
Compute invariant prototypes P via Eq. (14)
Compute refined prototypes $\mathbf{P}^{\circ}$ via Eq. (15)
for each node v_i **do**
  Compute $\hat{\mathbf{z}}_i = \texttt{GNN}^{\circledast}(\mathbf{X}_i, \mathbf{A}_i)$
  Predict label $\hat{y}_i$
end
(Few-shot Adaptation)
for each class c **do**
  Update prototype $\mathbf{p}_c^{\bullet}$ via Eq. (17)
end
Optimize $\mathcal{L}_{pdr}$ via Eq. (18)
for each node v_i **do**
  Predict label $\hat{y}_i$
end

E Detailing Related Work

E.1 Text-Attributed Graph Learning

Text-attributed graphs (TAGs), which integrate node-level textual attributes with graph topology, provide a unified representation for real-world relational systems such as citation networks, social media, and recommendation platforms. Early approaches Chen et al. (2023); Tu et al. (2017) typically follow a two-stage pipeline that independently encodes text features and then propagates them over the graph, leading to limited interaction between semantic and structural information. To address this issue, subsequent works focus on joint modeling of text and graph structures. Methods such as BertGCN Lin et al. (2021), GraphFormers Yang et al. (2021), and Patton Jin et al. (2023) tightly couple textual semantics with structural dependencies, enabling more expressive representation learning. Meanwhile, self-supervised frameworks further improve robustness by leveraging contrastive objectives and data augmentation strategies to enhance cross-view consistency. Recent works further explore LLM-driven augmentation for TAGs, such as GAugLLM Fang et al. (2024), which leverages large language models to generate semantically consistent graph views and alleviate the mismatch between textual attributes and graph structures.

Recently, large language models (LLMs) have significantly reshaped TAG learning. Existing approaches can be broadly categorized into three directions. (1) **LLM-enhanced representation learning**, where LLMs are used to generate or refine node features, improving semantic expressiveness. However, such methods often suffer from misalignment between textual and structural information due to inconsistent learning difficulty across nodes, as highlighted by recent alignment frameworks Chen et al. (2025). (2) **Text-space graph reasoning**, where graph tasks are reformulated as natural language reasoning problems, as in GraphText Zhao et al. (2024) and TalkGraph Fatemi et al. (2023). (3) **LLM–GNN hybrid frameworks**, which align graph representations with language space through instruction tuning or adapter mechanisms, including GraphGPT Tang et al. (2024), LLaGA Chen et al. (2024a), and GraphAdapter Huang et al. (2024).

More recent studies further explore converting graph structures into textual representations. For instance, topology-aware node description synthesis methods Wang et al. (2025b) leverage LLMs to generate node descriptions from graph topology, enabling text-free graphs to be transformed into TAGs. This line of work highlights that language can serve as a universal interface for graph learning, facilitating cross-domain transfer and improving scalability. From a broader perspective, recent surveys emphasize that TAG learning and LLMs exhibit strong complementarity: TAGs provide structured relational inductive bias, while LLMs contribute rich semantic understanding Su et al. (2025). Emerging frameworks further attempt to unify graph and language representations, such as graph vocabulary learning approaches Zhu et al. (2025a), which treat graph reasoning as a form of language modeling.

E.2 Graph Foundation Models

Inspired by the success of foundation models in natural language processing and computer vision Radford et al. (2021); Devlin et al. (2019), Graph Foundation Models (GFMs) aim to learn general-purpose representations from large-scale graph data, enabling a "pretrain-once, generalize-to-many" paradigm Liu et al. (2023b). Recent studies further formalize GFMs as a unified framework integrating backbone architectures, pretraining strategies, and adaptation mechanisms, highlighting their role as foundational infrastructure for scalable graph intelligence Wang et al. (2025c).

Existing GFMs can be broadly categorized into two main directions. **(1) Text-free GFMs.** These methods focus on learning transferable knowledge purely from graph structures and numerical attributes. Representative approaches such as BRIDGE Yuan et al. (2025), SAMGPT Yu et al. (2025), and MDGFM Wang et al. (2025a) adopt multi-domain pretraining strategies to capture domain-invariant structural patterns. These methods demonstrate that structural pretraining can provide strong inductive bias for cross-domain transfer, but are inherently limited by the absence of semantic interfaces, which restrict their ability to generalize across heterogeneous label spaces. **(2) Text-attributed GFMs.** To overcome the lack of semantic transferability, recent works leverage natural language as a unified interface for cross-domain graph learning. In zero-shot settings, methods such as OFA Liu et al. (2023a), ZeroG Li et al. (2024), and GraphCLIP Zhu et al. (2025b) perform prediction via semantic prototype matching between node representations and class descriptions. In particular, GraphCLIP introduces graph-summary contrastive pretraining to align graph representations with semantic summaries, significantly improving cross-domain zero-shot generalization.

Another line of work employs large language models (LLMs) as the prediction backbone, including UniGraph He et al. (2025), GraphGPT Tang et al. (2024), TEA-GLM Wang et al. (2024), and GOFA Kong et al. (2025), which project graph representations into the language space and formulate downstream tasks as generative reasoning problems. These approaches highlight a paradigm shift where graph learning is increasingly treated as a form of language understanding. Recent trends further explore deeper integration between LLMs and GFMs. For example, PromptGFM introduces graph vocabulary learning to unify graph and language representations, addressing the incompatibility between graph-specific tokens and natural language vocabularies, and improving transferability across graphs and tasks.

F Description of Datasets

Following GraphCLIP Zhu et al. (2025b), we conduct experiments on seven benchmark datasets drawn from four groups: **(1) Academic Domain**, **(2) E-commerce Domain**, **(3) Social Domain**, and **(4) Wikipedia Domain**. For each domain, one or several large-scale datasets are selected as source datasets for pre-training, while the remaining datasets are used as target datasets for evaluation. This domain-aware setup supports knowledge transfer across datasets with different characteristics and provides a benchmark for evaluating the generalization performance of the proposed model. Table 4 summarizes the statistics of all datasets.

- **ogbn-ArXiv** Hu et al. (2020); Wang et al. (2020a) is a large-scale citation network from the Open Graph Benchmark, constructed from the Microsoft Academic Graph (MAG). It contains about 169K papers and over 1.1M citation edges, where nodes correspond to arXiv computer science papers and directed edges indicate citation relationships. The task is to classify papers into 40 subject categories, such as cs.AI and cs.LG.

Table 4: The statistics of datasets.

Dataset	Train	Test	#Nodes	#Edges	#C
Academic Domain					
ogbn-ArXiv	✓	-	169,343	1,166,243	40
ArXiv_2023	✓	-	46,198	78,543	40
PubMed	✓	-	19,717	44,338	3
Cora	-	✓	2,708	5,429	7
CiteSeer	-	✓	3,186	4,277	6
E-commerce Domain					
ogbn-Products	✓	-	2,449,029	61,859,140	47
Photo	-	✓	48,362	500,928	12
Computers	-	✓	87,229	721,081	10
History	-	✓	41,551	358,574	12
Social Domain					
Reddit	✓	-	33,434	198,448	2
Instagram	-	✓	11,339	144,010	2
Wikipedia Domain					
WikiCS	-	✓	11,701	215,863	10

- **ArXiv_2023** He et al. (2023) is a large-scale directed citation graph of computer science papers published on arXiv in 2023 or later. Nodes correspond to papers with textual attributes (titles and abstracts), and directed edges represent citation relationships between papers. The learning objective is node classification, where each paper is assigned to one of 40 computer science categories defined by the authors and arXiv moderators.
- **PubMed** Yang et al. (2016) is a biomedical citation network constructed from scientific articles in the PubMed database related to diabetes research. In this graph, nodes represent individual publications and edges correspond to citation relationships between papers. The dataset contains 19,717 articles and tens of thousands of citation links, and the node classification task aims to assign each paper to one of three diabetes-related research topics.
- **Cora** Sen et al. (2008) is a widely used citation network benchmark composed of 2,708 machine learning publications connected through 5,429 citation links. In this graph, each node represents a research paper, and edges correspond to citation relationships between papers. The node classification task aims to assign each paper to one of seven research topics.
- **CiteSeer** Giles et al. (1998) is a citation network dataset constructed from papers collected in the CiteSeer digital library. The graph contains approximately 3186 scientific publications connected by about 4,277 citation links, where nodes represent individual papers and edges indicate citation relationships between them. The node classification task aims to assign each paper to one of six computer science research areas.
- **ogbn-Products** Hu et al. (2020) is a large-scale Amazon product co-purchasing network. Nodes denote products, while edges indicate co-purchase relationships between products. It contains around 2.4 million nodes and 61 million edges, and the task is to classify each node into one of 47 product categories.
- **Computers and Photo** Yan et al. (2023); Ni et al. (2019) are two product co-interaction graphs extracted from the Amazon dataset. The former contains products from the Computers category, while the latter focuses on items under the Photo category. In both graphs, nodes denote electronic products and edges indicate that two products are frequently co-purchased or co-viewed. Product labels correspond to the third-level categories in the Amazon product taxonomy, resulting in 10 classes for Computers and 12 classes for Photo. The task is formulated as node classification over the product graph.
- **History** Yan et al. (2023) is a book-domain graph derived from the Amazon-Books collection, focusing on items under the History branch. In this dataset, nodes correspond to individual books, while edges capture behavioral associations such as frequent co-purchase or co-view patterns. Each book is annotated with a third-level category label, and its title and description are used as textual attributes. The task is to classify books into 12 fine-grained categories.

- **Reddit** Huang et al. (2024) is a large-scale social interaction graph in which nodes represent users in an online discussion network, and edges denote reply interactions between users. The dataset captures communication relationships in social communities and is widely used as a benchmark for node classification on large graphs.
- **Instagram** Huang et al. (2024) is a social network where nodes correspond to users and edges indicate following relationships. The task is to classify users in the network into commercial and normal accounts.
- **WikiCS** Mernyei & Cangea (2020) is a Wikipedia-based document graph where nodes represent computer science articles and edges denote hyperlink connections between them. Each article belongs to one of 10 categories, and the dataset is widely used for evaluating node classification and link prediction models.

G Description of Baselines

To provide a more comprehensive empirical assessment, we include baselines covering several closely related paradigms, including LLM-only modeling, traditional graph self-supervised representation learning, and methods specifically designed for text-attributed graphs. These baselines together span a broad spectrum of modeling strategies, ranging from purely textual understanding to joint exploitation of textual attributes and graph structure, thereby providing a more complete view of the relative strengths of our method under different evaluation settings.

LLM-only Methods: This category includes both pretrained language encoders and instruction-tuned large language models, including BERT Devlin et al. (2019), SBERT Reimers & Gurevych (2019), DeBERTa He et al. (2020), E5 Wang et al. (2022), Qwen2-7B-Instruct Yang et al. (2024a), Qwen2-72B-Instruct Yang et al. (2024a), LLaMA3.1-8B-Instruct Sam (2024), and LLaMA3.1-70B-Instruct Sam (2024), which focus on textual semantic modeling without explicitly exploiting graph structure. Given the textual attributes of each node, these methods generate semantic representations or textual predictions for downstream tasks. They provide strong semantic-only baselines for evaluating the contribution of graph structural information.

Traditional Self-supervised Methods: Traditional self-supervised graph representation learning methods aim to learn informative node embeddings without relying on labeled data by designing pretext tasks such as contrastive learning, bootstrap prediction, and masked reconstruction.

- DGI Veličković et al. (2019) maximizes the mutual information between local node representations and a global graph summary by training a discriminator to distinguish true node–summary pairs from those constructed using embeddings derived from a corrupted graph.
- GRACE Zhu et al. (2020) is a graph contrastive learning framework that generates two augmented graph views through structural and feature perturbations. It learns node representations by maximizing the agreement between embeddings of the same node across the two views while contrasting them against embeddings of other nodes.
- BGRL Thakoor et al. (2021) removes negative samples and adopts a bootstrap prediction mechanism inspired by BYOL, where an online encoder predicts representations produced by a slowly updated target encoder.
- GraphMAE Hou et al. (2022) is a generative self-supervised graph learning framework based on masked autoencoders. It randomly masks a portion of node features and employs a graph encoder to learn latent representations from the partially observed graph, while a decoder is trained to reconstruct the masked node attributes.
- G2P2 Wen & Fang (2023) proposes a graph-grounded pre-training and prompting framework for low-resource text classification. It jointly pre-trains text and graph encoders using multiple graph interaction-based contrastive objectives and performs downstream classification through prompt-based adaptation to effectively leverage both textual semantics and graph structural information.

TAG Methods: Recent studies have explored integrating large language models with text-attributed graphs to jointly model textual semantics and graph structures. These methods typically encode node

textual attributes with language models and incorporate structural information through alignment, prompting, or instruction tuning mechanisms, enabling improved generalization across diverse graph learning tasks.

- GraphGPT Tang et al. (2024) introduces a graph instruction tuning framework that integrates graph structural knowledge into large language models. It aligns graph encoder representations with natural language semantics through a text–graph grounding mechanism and employs a lightweight projector to bridge the graph encoder and the LLM. A dual-stage instruction tuning strategy further enables the model to perform various graph learning tasks using natural language and improves its zero-shot generalization ability.
- LLaGA Chen et al. (2024a) converts graph structures into structure-aware node sequences using node-level templates and maps them into the token embedding space through a projector. This design enables large language models to process graph-structured data and improves versatility, generalization, and interpretability across multiple graph tasks.
- OFA Liu et al. (2023a) constructs text-attributed graphs by representing nodes and edges with textual descriptions and encodes them into a unified embedding space using language models. It further introduces a graph prompting paradigm that inserts task-specific prompting substructures into the input graph, enabling the framework to adapt to diverse downstream tasks.
- ZeroG Li et al. (2024) addresses cross-dataset zero-shot graph learning by leveraging a language model to encode node attributes and class semantics into a unified representation space. It further introduces a prompt-based subgraph sampling strategy to capture structural and semantic patterns across graphs and adopts lightweight fine-tuning to preserve the zero-shot generalization capability.
- GraphCLIP Zhu et al. (2025b) enhances the transferability of graph foundation models for text-attributed graphs by introducing a graph–summary contrastive pretraining framework. It constructs large-scale graph–summary pairs and aligns graph structural representations with textual semantics through self-supervised contrastive learning, further incorporating invariant learning to improve cross-domain zero-shot and few-shot transferability.

H Details of Implementation

H.1 Dataset Usage and Splitting Protocol

Following GraphCLIP Zhu et al. (2025b), we use 12 open-text-attributed graphs from four domains, where 5 large-scale TAGs are used as source-domain graphs for pre-training and 7 small-scale TAGs are used as target-domain graphs for evaluation. During pre-training, only source-domain graphs are used, and no target-domain labels are accessed. For ogbn-Products, we use its official training set as the pre-training corpus, which contains around 200K products, to avoid overwhelming other source graphs and to balance the contribution of different source datasets.

For target-domain evaluation, we follow the public split for WikiCS. For the remaining target datasets, we randomly select 20% of nodes as the test set. The random split is performed in a class-stratified manner when label distributions are available, so that the test set preserves the original class distribution. The remaining 80% of nodes are not used as supervised training data in the zero-shot setting. They only provide unlabeled graph context for target-domain inference and test-time prototype refinement. During TIP, target labels are never used, and prototype refinement relies only on model predictions and perturbation-consistent target representations.

In the few-shot setting, we sample labeled support nodes from the non-test portion of each target graph. The support nodes and test nodes are strictly disjoint. The held-out 20% test nodes are kept unchanged for evaluation. We report the mean accuracy and standard deviation over five runs with different random seeds.

H.2 Pre-training Settings

Following GraphCLIP Zhu et al. (2025b), we adopt GraphGPS Rampášek et al. (2022) as the graph encoder and MiniLM Wang et al. (2020b) as the text encoder. Specifically, the graph encoder consists

of 12 layers with a hidden size of 1024, while the text encoder is a fine-tuned 6-layer MiniLM with a hidden size of 384. To align the two modalities in a shared embedding space, a linear projector is applied to map the 1024-dimensional graph representations into the 384-dimensional text space, with a dropout rate of 0.1 for regularization. We use AdamW Loshchilov & Hutter (2017) as the optimizer, with both the learning rate and weight decay set to 1×10^{-5}. During pre-training, the text encoder is frozen, and only the graph encoder and the projection layer are updated. The model is trained for 10 epochs with a batch size of 800 on 2 NVIDIA A40 GPUs, taking approximately 9 hours. For subgraph construction, we adopt k-hop ego-network sampling for smaller citation graphs (Cora and CiteSeer), while random walk sampling is used for larger graphs to improve scalability. The sampled subgraphs are limited to a fixed maximum size to control computational cost. The textual summary embeddings $\psi(\mathbf{T}^s)$ are pre-computed and reused across training, which significantly reduces computational overhead. For the loss balancing hyperparameters in TGP, we empirically set $\alpha = 2.0$ and $\beta = 0.2$ throughout all experiments.

H.3 Zero-shot Evaluation Settings

Zero-shot Node Classification. In the zero-shot setting, we incorporate class information into class-specific natural language prompts to maintain consistency with the pre-training format. For each target dataset, we design dataset-specific prompt templates, where $\{\texttt{class}\}$ denotes the class label text and $\{\texttt{class_desc}\}$ denotes a descriptive sentence that provides additional semantic information about the label. The class descriptions are generated by a large language model, where the generation process is guided to emphasize class-specific characteristics and avoid semantic overlap across categories, thus improving class separability. The detailed prompt templates are summarized in Table 5. During inference, node representations are matched with class prototypes derived from textual prompts. The refined class prototypes are obtained by combining semantic class prototypes and invariant target-domain prototypes with $\lambda = 0.3$, as defined in Eq. 15. To ensure robustness, all results are averaged over five runs with different random seeds.

Zero-shot Link Prediction. Following the protocol of GraphCLIP Zhu et al. (2025b), we evaluate the pretrained model on zero-shot link prediction without any fine-tuning. For each dataset, 50% of candidate links are randomly selected as test samples. We compute similarity scores between node representations and report the mean AUC along with standard deviation over five runs. Negative samples are generated by randomly sampling node pairs that are not connected in the graph.

H.4 Few-shot Evaluation Settings

In the few-shot setting, we freeze the pretrained encoder and update only the graph adapter to achieve lightweight adaptation. The optimizer is AdamW with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}, and the dropout rate is set to 0.1. For each target dataset, we construct a strict k-shot support set by randomly sampling up to k labeled nodes per class. The support set is used for adapter optimization and prototype construction, while the model is evaluated on the remaining unlabeled nodes. During adaptation, prototypes are constructed by combining semantic class descriptions and structural evidence from support nodes. The training objective includes both a classification loss and a prototype reconstruction term to enhance intra-class compactness. The adaptation process is performed for a small number of epochs to avoid overfitting under limited supervision.

All few-shot results are reported as the average over five independent runs with different sampled support sets.

H.5 Node Texts and Subgraph Summaries

To illustrate the textual inputs used in our framework, we provide several representative examples, including raw node texts from different domains and subgraph-level summary texts. The raw node texts are directly taken from the original attributes of each dataset, such as paper titles and abstracts in citation graphs (e.g., Cora). In addition, we incorporate subgraph summaries as high-level semantic supervision. These summary texts are constructed following the same strategy as GraphCLIP, and are used to capture the semantic and structural context of local subgraphs. Together, these examples provide an intuitive view of the heterogeneous textual information involved in our framework.

Table 5: Prompts for zero-shot learning

Datasets	The Class Prompts and Description for Zero-shot Prediction
Cora	"this paper is a study on {class}, this paper addresses {class_desc}"
CiteSeer	"A good paper of {class}, this paper addresses {class_desc}"
WikiCS	"it belongs to {class} research area {class_desc}"
Instagram	"An account of {class} {class_desc}"
Photo	"this product falls into a category {class} {class_desc}"
Computers	"is {class} category {class_desc}"
History	"this book belongs to {class}, and {class_desc}"

Original Text of the Node(Cora)

Node ❶ Title: *The megaprior heuristic for discovering protein sequence patterns.*
Abstract: Several computer algorithms for discovering patterns in groups of protein sequences are in use that are based on fitting the parameters of a statistical model to a group of related sequences. These include hidden Markov model (HMM) algorithms for multiple sequence alignment, and the MEME and Gibbs sampler algorithms for discovering motifs...

Node ❷ Title: *Applications of machine learning: a medical follow up study.*
Abstract: This paper describes preliminary work that aims to apply some learning strategies to a medical follow-up study. An investigation of the application of three machine learning algorithms-1R, FOIL and InductH to identify risk factors that govern the colposuspension cure rate has been made. The goal of this study is to induce a generalised description or explanation of the classification attribute...

Summary of subgraph (PubMed)

Paper Summary and Context Analysis: The paper represented by node 'n0' is titled "Retinal metabolic abnormalities in diabetic mouse: comparison with diabetic rat." The study examines metabolic abnormalities in the retinas of diabetic mice and rats to assess their relevance to diabetic retinopathy. Key findings include increased retinal oxidative stress, protein kinase C (PKC) activity, and nitric oxide levels in diabetic mice, similar to those observed in diabetic rats. Additionally, sorbitol levels increased threefold. The paper suggests that mice can serve as a promising model for studying the pathogenesis of diabetic retinopathy, given the comparable levels of metabolic abnormalities to those found in rats.
Structural Summary: The papers that cite or are cited by 'n0' (its direct neighbors) share common themes related to diabetic retinopathy, including studies on the pathogenesis, biochemical changes, and the impact of diabetes on retinal metabolism. For instance, 'n2918' investigates retinal ion regulation in a mouse model of diabetic retinopathy, emphasizing the role of ionic dysregulation and the effect of Cu/Zn superoxide dismutase overexpression. 'n12388' compares early diabetes-induced biochemical changes in the retina of rats and mice, noting differences in biochemical markers that suggest mice may not manifest all changes seen in rats under diabetic conditions. 'n14295' explores genetic differences in susceptibility to blood-retina barrier breakdown in diabetic rats, while 'n14618' examines the role of tumor necrosis factor-alpha in promoting apoptosis and loss of retinal microvascular cells in type 1 and type 2 models of diabetic retinopathy. Research Area Classification The specific research area to which 'n0' primarily contributes is "diabetic retinopathy", focusing on the pathogenesis and metabolic effects of diabetes on the retina...

I SUPPLEMENTARY EXPERIMENTS

I.1 ZERO-SHOT LINK PREDICTION.

Table 6 reports zero-shot link prediction results on Cora, WikiCS, and History. TI-GFM achieves the best performance on all three datasets, reaching 92.32%, 94.11%, and 97.50%, respectively. Compared with GraphCLIP, TI-GFM improves AUC by 9.17% on Cora, 1.44% on WikiCS, and 1.46% on History. The results reveal three trends. First, semantic-only models such as SBERT and E5 show competitive performance on some datasets, especially SBERT on History, but they still lag behind TI-GFM by 14.56%, 12.56%, and 10.48% on Cora, WikiCS, and History, respectively. This indicates that textual semantics alone are insufficient for reliable relation-level transfer. Second, traditional graph self-supervised methods, including GRACE and GraphMAE, perform worse than TAG-based methods, suggesting that topology-only pre-training lacks an explicit semantic interface for zero-shot transfer across heterogeneous graphs. Third, TAG-based GFMs such as ZeroG and GraphCLIP achieve stronger results, confirming the benefit of graph–text alignment. However, TI-GFM further surpasses ZeroG by 11.12%, 5.29%, and 4.65%, and consistently outperforms GraphCLIP on all datasets.

Since zero-shot link prediction relies on node-pair similarity rather than class-prototype matching, these gains suggest that TGP learns a more transferable representation space where semantically compatible and structurally related nodes are better aligned. The large improvement on Cora shows that TI-GFM can substantially enhance relation-level generalization when baseline representations remain insufficient, while the consistent gains on WikiCS and History demonstrate that the improvement persists even when GraphCLIP already performs strongly.

Table 6: Zero-shot link prediction AUC (%).

Methods	Cora	WikiCS	History
SBERT	77.76±0.54	81.55±0.09	87.02±0.07
E5	69.57±0.60	80.69±0.06	69.93±0.07
GRACE	67.19±0.56	69.33±0.06	76.90±0.05
GraphMAE	70.17±0.77	73.39±0.04	83.26±0.07
OFA	75.36±0.89	89.03±0.09	86.47±0.10
ZeroG	81.20±0.82	88.82±0.07	92.85±0.12
GraphCLIP	83.15±0.76	92.67±0.10	96.04±0.05
Ours	**92.32±0.32**	**94.11±0.07**	**97.50±0.03**
Δ	**9.17↑**	**1.44↑**	**1.46↑**

I.2 CROSS-DOMAIN TRANSFER ANALYSIS.

Figure 5 shows the transfer performance under different source-domain settings. The *Full-Domain* setting achieves the most balanced and strongest results, indicating that diverse source domains encourage learning domain-invariant representations. In contrast, single-domain pretraining exhibits clear domain bias: ***only-Academic*** performs better on Cora and CiteSeer, ***only-E-commerce*** on Computers and Photo, and ***only-Social*** on Instagram. These results suggest that broader source-domain coverage mitigates domain bias and improves transfer stability across target graphs.

I.3 SENSITIVITY TO λ

Figure 6 studies the effect of the fusion weight λ, which controls the balance between semantic prototypes and invariant prototypes. A consistent observation is that most datasets achieve strong or near-best performance around $\lambda = 0.3$, indicating that neither semantic prototypes nor invariant prototypes alone are sufficient, and that an appropriate combination of the two is essential for effective transfer. Moreover, except for Cora, all datasets are relatively sensitive to the choice of λ. When λ deviates from 0.3, performance generally drops to varying degrees, especially on WikiCS, Computers, and History. This trend suggests that the fusion ratio plays a critical role in determining prototype quality. When λ is too small, the model relies excessively on invariant prototypes, weakening the contribution of semantic priors; when λ is too large, the model overemphasizes semantic prototypes and cannot sufficiently benefit from invariant target evidence. In contrast, Cora remains comparatively

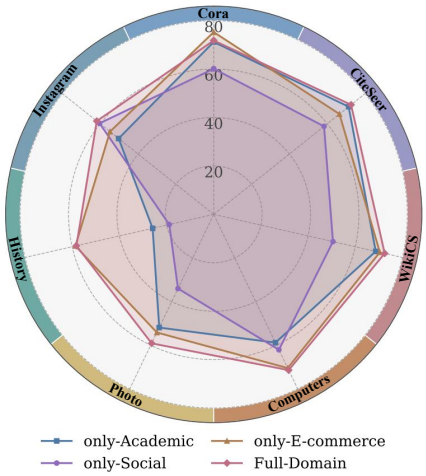


Figure 5: Cross-domain transfer performance across different datasets.

stable across different values of λ, implying that its prediction is less sensitive to the prototype fusion ratio. Overall, these results show that robust target adaptation requires a proper balance between semantic prototypes and invariant prototypes, and further support our motivation that semantic prototypes alone are insufficient for reliable transfer.

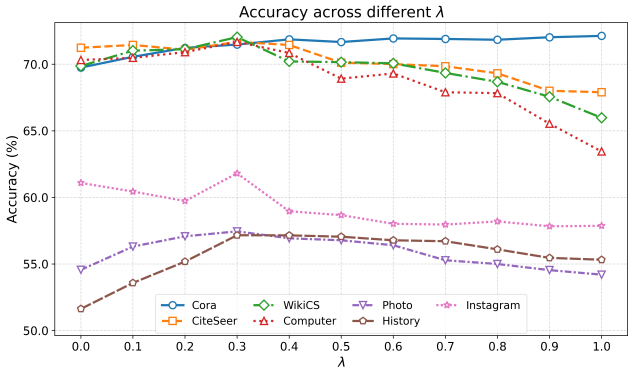


Figure 6: Sensitivity analysis of the fusion weight λ across different datasets.

I.4 Sensitivity to views K

We further investigate the effect of the number of views K on model performance across different datasets. As shown in Figure 7, the overall trend exhibits clear non-monotonic behavior: most datasets achieve their best performance at a moderate number of views (e.g., $K = 3$), while performance tends to plateau or slightly decline when K becomes larger. Meanwhile, the variation across datasets indicates that the effectiveness of multi-view learning is sensitive to data characteristics.

Specifically, the Cora dataset achieves its best performance at $K = 10$, suggesting that introducing more views helps better exploit complementary structural and semantic information on this dataset. In contrast, the Photo dataset reaches its optimum at $K = 1$, and its performance slightly decreases as K increases. This indicates that the original view already provides sufficiently discriminative information, while additional views may introduce noise or redundant perturbations, leading to representation drift and degraded classification performance. For the remaining datasets, including CiteSeer, WikiCS, Computers, History, and Instagram, the model generally achieves the best or near-best performance at $K = 3$. This observation suggests that a moderate number of views provides a favorable balance between information diversity and representation consistency: a small

K is insufficient to fully exploit complementary multi-view information, whereas an excessively large K may introduce redundancy and even disrupt the existing structural consistency.

Overall, these results show that the effectiveness of multi-view learning depends on the match between the number of views and the underlying data distribution. A moderate number of views can maximize complementary information while avoiding noise accumulation and representation drift. Therefore, considering the overall performance across datasets, we adopt $K = 3$ as the default setting in subsequent experiments to achieve a good trade-off between performance and stability.

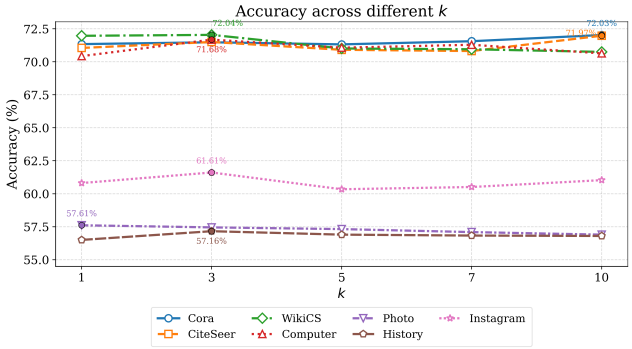


Figure 7: Sensitivity analysis of the views K across different datasets.

I.5 Limitation Analysis on Instagram.

The relatively limited performance on Instagram mainly arises from the weak correspondence between textual attributes and class semantics. Although the dataset contains only two classes, *Normal Users* and *Commercial Users*, user descriptions are often fragmented personal bios, interest tags, or occupational information, which do not strictly align with the class definitions. As shown in Figure 8, the two classes remain heavily mixed in the embedding space, with blurred class boundaries, making it difficult to form clearly separable prototype-centered structures. Compared with GraphCLIP, TI-GFM exhibits a similar prototype distribution on Instagram, with substantial class overlap still remaining in the embedding space. This suggests that, when textual signals lack sufficiently discriminative semantic cues, prototype refinement alone is insufficient to fully resolve class mixing and prototype misalignment.

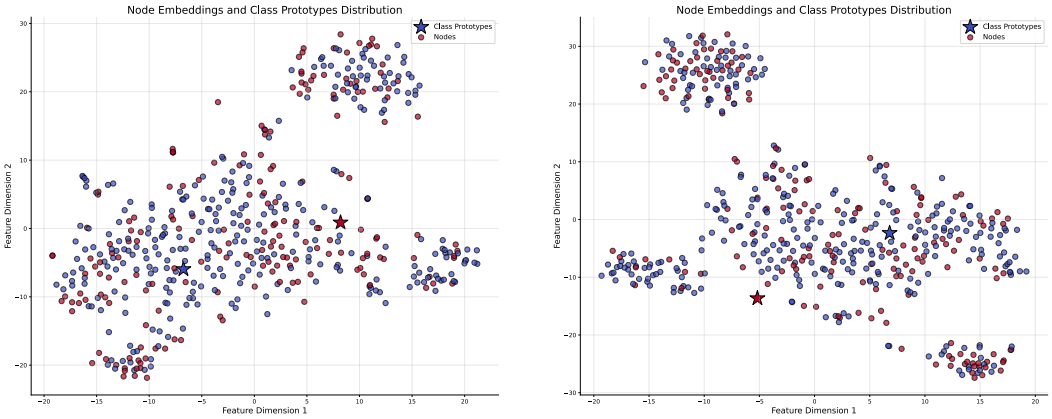
This case highlights a limitation of TI-GFM. When node texts are noisy, semantically loose, or weakly correlated with class labels, the quality of text-induced prototypes can degrade substantially. Although test-time adaptation can leverage target-graph structure for compensation, such correction remains limited under weak semantic supervision. A possible direction for future work is to incorporate more robust text denoising strategies and stronger structural augmentation, so as to improve class-semantic extraction under noisy text conditions and enhance prototype adaptation to weak semantic signals.

Original Text of the Node (Instagram)

❶ **Comment:** The News-What It Means, Why It Matters. Listen on AM 640 & iHeartRadio app. Los Angeles & Orange County. News Links below...
❷ **Comment:** Small town Pa boy mom+wife life Makeup+skincare enthusiast Brca previvor Summer forever...
❸ **Comment:** Curator of fashion popups — Celebrity Stylist — Content Creator Kolkata, India...

I.6 Few-shot Evaluation on Additional Target Datasets

To complement the few-shot evaluation on WikiCS and History, we extend the experiments to Cora, CiteSeer, Photo, Computers, and Instagram under 1-, 3-, 5-, and 10-shot settings. Together, these experiments cover all seven target datasets. The additional results compare TI-GFM with GPPT,



(a) GraphCLIP Instagram visualization. (b) TI-GFM Instagram visualization.

Figure 8: Embedding and prototype distribution on Instagram.

GraphPrompt, All-in-One, GPF, and GraphCLIP. The Average panel reports the unweighted mean accuracy over the five additional datasets. TI-GFM achieves the highest accuracy in 15 of the 20 additional dataset–shot settings and the best average performance at every shot level. Its average accuracies are 65.54%, 67.83%, 68.72%, and 69.90%, respectively. The strongest baseline by average accuracy is All-in-One for 1-shot, GraphCLIP for 3-shot, and GPF for both 5-shot and 10-shot. The largest average advantage occurs in the 3-shot setting, where TI-GFM exceeds GraphCLIP by 5.58 percentage points. At the dataset level, TI-GFM consistently ranks first on CiteSeer, Photo, and Computers across all four shot levels. On Photo, it surpasses the strongest baseline by 7.58 and 8.35 percentage points under 1-shot and 3-shot supervision, respectively, providing further evidence of its effectiveness with limited labeled target nodes. The improvements are dataset dependent. On Cora, TI-GFM ranks first in three settings but trails GPF under 3-shot supervision (74.72% versus 77.16%). On Instagram, it does not achieve the highest accuracy at any shot level, and additional support labels do not consistently improve performance. These results demonstrate strong average performance across shot levels while highlighting limitations on challenging target domains.

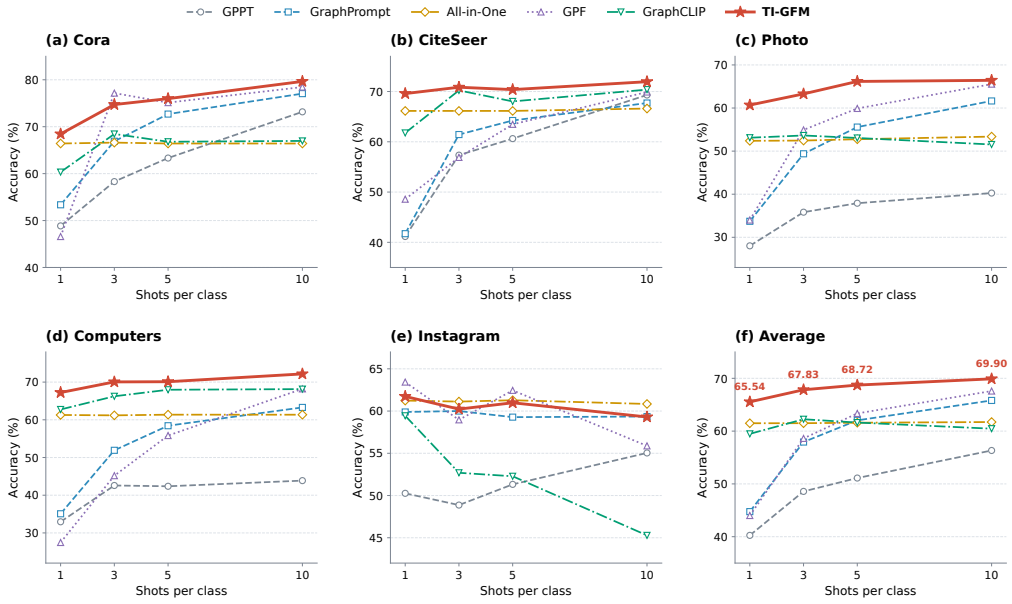


Figure 9: Few-shot node classification accuracy (%) on five additional datasets. Average denotes the unweighted mean across datasets.

J DECLARATION OF LLM USAGE

We used Large Language Models (LLMs) as auxiliary tools to polish parts of the manuscript, including grammar correction and fluency improvement. In our experiments, we reused the subgraph summaries generated offline by Qwen2-72B-Instruct and released with GraphCLIP, and additionally used LLMs to generate class prompts. The authors are responsible for the study conception, method design, experimental implementation, data interpretation, and scientific conclusions. All technical content, experimental results, and conclusions presented in this manuscript are the sole responsibility of the authors.

K REPRODUCIBILITY STATEMENT

To facilitate the reproducibility of TI-GFM, we provide the following resources and implementation details:

- **Code Availability**: The implementation of TI-GFM is available at `https://anonymous.4open.science/r/GFM-B20A/`.
- **Datasets and Textual Inputs**: The appendix describes the datasets, source–target partition, evaluation splits, and subgraph sampling procedures. We reuse the offline subgraph summaries released with GraphCLIP and provide dataset-specific class prompt templates in Table 5.
- **Experimental Setup**: Appendix H describes the model architecture, pre-training settings, hyperparameters, and evaluation protocols for zero-shot node classification, few-shot node classification, and zero-shot link prediction. Results are averaged over five runs with different random seeds.