COLA: COMPLEMENTARY CONTEXT RETRIEVAL WITH UTILITY CALIBRATION FOR GRAPH FOUNDATION MODEL ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the problem of few-shot adaptation for graph foundation models (GFMs), which aims to transfer pre-trained GFMs to unseen domains with limited support sets. Previous approaches usually introduce learnable task-aware prompts to adapt GFMs to new domains. Despite the progress, these approaches primarily exploit parametric knowledge in pre-trained models while neglecting rich contextual knowledge embedded in source graphs. Towards this end, we propose a novel approach named Complementary Context Retrieval with Utility Calibration (**COLA**) for graph foundation model adaptation. The core of our **COLA** is to retrieve relevant graph candidates from complementary knowledge sources and then refine them into fine-grained contexts guided by their estimated utility for graph representation adaptation. In particular, we first build a graph semantics database and a graph structure database from source domains to capture complementary knowledge. Given a target query, we retrieve relevant graph candidates from these two databases using graph representations and graph kernels, respectively. More importantly, we leverage reinforcement learning with pseudo-target calibration to train a lightweight policy model, which estimates the utility of retrieved graphs to guide fine-grained context selection from abundant candidates. The refined complementary contexts are then incorporated into target graph representations under the guidance of null-aware routing for domain adaptation. Extensive experiments on benchmark datasets validate the effectiveness of the proposed **COLA** in comparison with extensive baselines. Our code is available at https://anonymous.4open.science/r/COLA-D0DC.

1 INTRODUCTION

Graph foundation models (GFMs) have emerged as a promising paradigm for learning transferable knowledge from multiple source graphs and adapting it to previously unseen graph domains and downstream tasks (Guo et al., 2026; Luo et al., 2026a; Mao et al., 2024; Shi et al., 2024). By pre-training on diverse graph data, GFMs aim to capture generalizable representations that can support a wide range of downstream applications with limited supervision (Shi et al., 2026; Yuan et al., 2026a). A central challenge therefore lies in how to effectively exploit the knowledge acquired during pre-training when adapting to a new target domain, especially when only a few labeled instances are available. This makes few-shot target adaptation a critical component in realizing the cross-domain generalization capability of GFMs.

During target adaptation, existing GFMs generally follow a pre-train-and-adapt paradigm, where knowledge acquired from multiple source graphs is transferred to a new downstream task using limited target supervision (Zhao et al., 2024; Yu et al., 2025). A common strategy is to fine-tune the pretrained model on the target task, updating either the entire backbone or a subset of its parameters depending on the amount of available supervision (Wang et al., 2024a;b). More recent approaches adopt parameter-efficient adaptation, such as prompt-based tuning, which introduces task-specific prompts to steer the pretrained representations while largely preserving the backbone (Sun et al., 2022; Liu et al., 2023; Yu et al., 2024a). Another common choice is to keep the pretrained encoder fixed and optimize only lightweight task-specific modules or prediction heads for the target task (Madhu et al., 2026; Hou et al., 2022; 2023). Although these strategies differ in the amount and

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.

form of target-side optimization, they share a common principle: downstream adaptation primarily exploits the knowledge already encoded in the pretrained parameters and representation space.

However, relying primarily on knowledge encoded in pretrained parameters and representations introduces two fundamental limitations for cross-domain graph adaptation. ❶ ***Heterogeneity between semantic and structural semantics in graphs.*** Graph information is jointly characterized by textual semantics and structural patterns, yet their cross-domain transferability does not necessarily coincide. Instances with similar textual semantics may occupy different structural roles, while structurally analogous neighborhoods may contain substantially different textual content (Yu et al., 2025; Guo et al., 2025). Modeling cross-domain knowledge through a unified representation can therefore obscure such complementary relationships and overlook transferable information that is prominent only in one view (Yu et al., 2025; Guo et al., 2025). ❷ ***Entangled representation knowledge with limited context relevance.*** Knowledge acquired from different source graphs is typically compressed into shared model parameters and representations, making it difficult to identify which source patterns are most useful for a particular target instance (Yuan et al., 2026b). This issue is amplified under few-shot supervision, where limited target information is available to select or suppress pretrained knowledge. Retrieval-augmented learning exposes source knowledge as explicit context, but retrieval relevance does not necessarily imply downstream utility, since highly similar contexts may still be uninformative or misleading (Tian et al., 2025; Jin et al., 2025). Effective cross-domain adaptation therefore requires distinguishing retrieval relevance from downstream utility when selecting source contexts.

To address these challenges, we propose `COLA`, a retrieval-augmented framework for few-shot graph adaptation. `COLA` first externalizes source knowledge into complementary semantic and structural graph databases, providing explicit contexts that preserve representation-level semantics and local topological patterns, respectively. Based on these databases, we develop a hierarchical retrieval mechanism that first identifies relevant source contexts and then evaluates their downstream utility, thereby separating retrieval relevance from downstream utility. The graph-kernel encoders are trained during source-side preparation, while the utility scorers and null-aware router are calibrated through source-only pseudo-target episodes without access to the real target domain. The selected semantic and structural contexts are subsequently incorporated through a null-aware routing mechanism, which adaptively enhances the target representation while providing a null route that allows the original backbone representation to be retained when retrieved contexts are not selected. During target adaptation, all source-derived retrieval and routing components remain fixed, and the lightweight downstream predictor is optimized using the few available target labels.

The main contributions are summarized as follows. ❶ ***Conceptual:*** We introduce a utility-oriented formulation for cross-domain graph retrieval, where source evidence is evaluated by its downstream usefulness rather than similarity alone. A source-only pseudo-target strategy enables utility estimation without access to the real target domain. ❷ ***Methodological:*** We design complementary semantic and structural retrieval mechanisms for cross-domain evidence augmentation. In particular, the structural branch employs a learnable random-walk graph kernel to capture transferable structural correspondence across multiple neighborhood scales. ❸ ***Experimental:*** Extensive experiments on few-shot node and ego-graph classification under both cross-dataset and cross-domain transfer settings demonstrate the effectiveness of `COLA`. Further ablation, efficiency, and sensitivity studies validate the contributions of its key components and its efficiency in target adaptation.

2 Related Work

Graph Foundation Models. Early graph pre-training and prompting methods aim to transfer pretrained knowledge across downstream tasks. GPPT connects pre-training with node classification through prompt tuning, while GraphPrompt extends this paradigm to both node- and graph-level tasks (Sun et al., 2022; Liu et al., 2023). OFA and ZeroG further broaden transfer across tasks and datasets (Liu et al., 2024; Li et al., 2024). Multi-domain GFMs additionally need to address shifts in node attributes, topology, and label semantics. Existing approaches employ cross-domain coordination or shared textual spaces (Zhao et al., 2024; He et al., 2025), structure- and domain-aware prompting (Yu et al., 2025), geometry-aware experts (Guo et al., 2025), topology alignment (Wang et al., 2025), bounded knowledge transfer (Yuan et al., 2025), and generative graph vocabularies (Yuan et al., 2026a). These methods mainly encode transferable knowledge in learned repre-

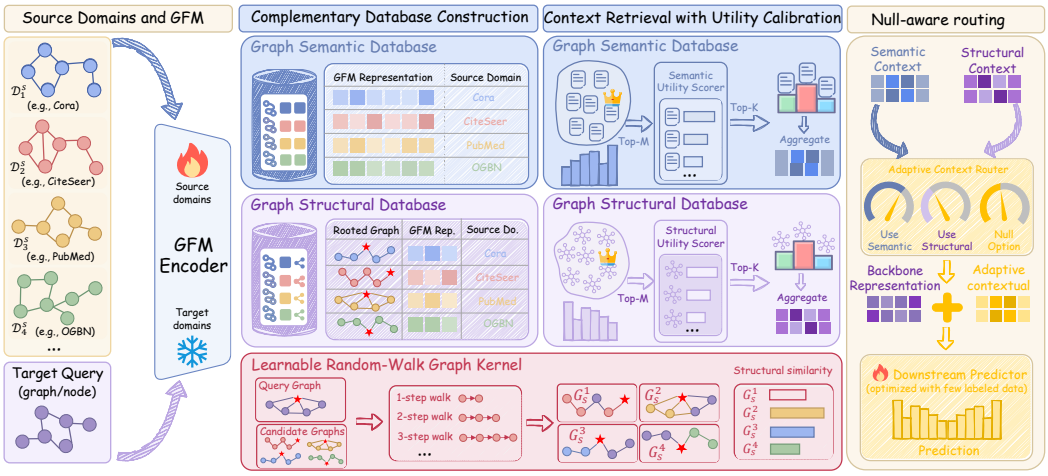


Figure 1: Overall framework of COLA. A frozen GFM encodes source domains and target queries, while complementary semantic and structural databases externalize source knowledge as explicit contexts. Hierarchical retrieval selects useful contexts from the two databases, which are adaptively integrated with the backbone representation through null-aware routing for few-shot prediction.

sentations or adaptation modules, whereas our setting retains explicit source instances and allows their contribution to vary across target queries.

Retrieval-augmented Graph Learning. Retrieval has been incorporated into graph learning along two directions. One uses graph structure to support retrieval for language models, including G-Retriever for graph question answering and GFM-RAG for graph-structured document retrieval (He et al., 2024; Luo et al., 2026b). The other retrieves graph instances to improve graph prediction. RAGraph retrieves task-relevant examples from an external graph library (Jiang et al., 2024), while RAG-GFM externalizes source knowledge into semantic and structural memories for cross-domain adaptation (Yuan et al., 2026b). These methods show that transferable graph knowledge can be accessed explicitly, but retrieval is still largely based on predefined similarity criteria. Our method further distinguishes retrieval relevance from downstream utility and learns which source contexts are useful for each target query.

3 The Proposed COLA

3.1 Problem Setting and Preliminaries

Graph Foundation Models. An attributed graph is denoted by $G = (V, E, \boldsymbol{X})$, with adjacency matrix $\boldsymbol{A} \in \{0,1\}^{|V|\times|V|}$ and node features $\boldsymbol{X} \in \mathbb{R}^{|V|\times d_{\text{in}}}$. Given N_S source domains $\mathcal{D}^S = \{\mathcal{D}_s^S\}_{s=1}^{N_S}$, a GFM learns a shared encoder $f_{\boldsymbol{\Theta}}$ and fixes its parameters as $\boldsymbol{\Theta}^\star$ after source pre-training. For an instance x, we define its backbone representation as

$$\boldsymbol{h}_{\text{base}}(x) = \text{Readout}_x\left(f_{\boldsymbol{\Theta}^\star}(\boldsymbol{A}_x, \boldsymbol{X}_x)\right) \in \mathbb{R}^d, \tag{1}$$

where Readout_x selects the center-node representation for node classification and aggregates node representations for graph classification. The target domain $\mathcal{D}^T$ is unseen during source pre-training.

Few-shot Adaptation with Frozen GFMs. For a C-class target task, the support set $\mathcal{S}^T = \bigcup_{c=1}^{C}\{(x_{c,j}, c)\}_{j=1}^{m}$ contains m labeled instances per class, with an unlabeled query set $\mathcal{Q}^T$. During target adaptation, the pretrained GFM and all source-derived parameters $\boldsymbol{\Phi}$ remain fixed, while the support labels are used to optimize only the downstream predictor $g_{\boldsymbol{\omega}}$, leaving the query labels solely for evaluation:

$$\boldsymbol{\omega}^\star = \arg\min_{\boldsymbol{\omega}} \sum_{(x,y)\in\mathcal{S}^T} \mathcal{L}_{\text{task}}\left(g_{\boldsymbol{\omega}}\left(\boldsymbol{h}_{\boldsymbol{\Phi}}(x)\right), y\right). \tag{2}$$

3.2 Framework Overview

As illustrated in Figure 1, COLA augments a frozen GFM with explicit source contexts through complementary database construction, hierarchical context retrieval with utility calibration, and null-aware representation enhancement. During source-side preparation, the frozen GFM representations form the semantic database, while source-domain structural encoders are trained to characterize local topology and then frozen to construct the structural databases. Given a query, COLA first retrieves compact candidate pools according to semantic or structural relevance and then re-ranks them according to their estimated downstream utility, thereby separating retrieval relevance from context usefulness. The utility scorers and null-aware router are calibrated using source-only pseudo-target episodes while the backbone, structural encoders, and retrieval databases remain fixed. During target adaptation, the selected semantic and structural contexts are aggregated into representation residuals and adaptively combined with the original backbone representation through the null-aware router.

3.3 Complementary Database Construction for Explicit Contexts

A pretrained GFM compresses source knowledge into shared parameters and representations, limiting direct access to instance-level information during target adaptation. We therefore externalize complementary source knowledge into fixed semantic and structural databases.

Graph Semantic Database. The frozen GFM provides a shared representation space in which source instances can be organized according to their semantic characteristics. For each source instance x_i, its backbone representation $\boldsymbol{c}_i^{\mathrm{sem}} = \boldsymbol{h}_{\mathrm{base}}(x_i)$ serves as the semantic context, forming:

$$\mathcal{M}_{\mathrm{sem}}^{S} = \{(\boldsymbol{c}_i^{\mathrm{sem}}, d_i)\}_{i=1}^{N_{\mathrm{db}}}, \tag{3}$$

where d_i records the corresponding source domain and N_{db} is the number of indexed source instances. Since the backbone remains fixed, the database provides a stable source representation space for comparing target queries with source instances. It therefore supports representation-level relevance retrieval while preserving the source-domain provenance of each retrieved context.

Graph Structural Database. To capture local topology beyond semantic proximity, we learn a structural encoder for each source domain using multi-scale random-walk propagation. For a rooted local graph $G = (V, E, r_G)$, nodes are initialized with standardized random-walk return-probability descriptors derived from $\boldsymbol{P}_G = \boldsymbol{D}_G^{-1}\boldsymbol{A}_G$, followed by a lightweight graph encoder producing $\boldsymbol{H}_G^{(0)}$. The node states are propagated and combined across walk lengths as:

$$\boldsymbol{H}_G^{(l+1)} = \boldsymbol{P}_G\boldsymbol{H}_G^{(l)}, \qquad \beta_{d,l} = \frac{\exp(b_{d,l})}{\sum_{j=0}^{L}\exp(b_{d,j})}, \qquad \widetilde{\boldsymbol{H}}_G^{d} = \sum_{l=0}^{L}\beta_{d,l}\boldsymbol{H}_G^{(l)}, \tag{4}$$

where L is the maximum walk depth. A root-aware readout summarizes the mean, maximum, root representation, and neighborhood dispersion, yielding the normalized structural embedding and induced kernel:

$$\boldsymbol{z}_d(G) = E_d(G;\boldsymbol{\psi}_d) = \mathrm{Norm}\Big(\phi_d\Big(\mathrm{Readout}_{r_G}(\widetilde{\boldsymbol{H}}_G^{d})\Big)\Big), \qquad K_d(G,G') = \boldsymbol{z}_d(G)^{\top}\boldsymbol{z}_d(G'). \tag{5}$$

Each E_d is trained only on source data by contrasting the structural and semantic views of the same local graph:

$$\mathcal{L}_{\mathrm{ker}}^{d} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\mathrm{sim}(\boldsymbol{z}_i^{\mathrm{str}}, \boldsymbol{z}_i^{\mathrm{sem}})/\tau)}{\sum_{j=1}^{B}\exp\big(\mathrm{sim}(\boldsymbol{z}_i^{\mathrm{str}}, \boldsymbol{z}_j^{\mathrm{sem}})/\tau\big)}. \tag{6}$$

The semantic encoder is used only for this source-side supervision, after which E_d is frozen. For each source domain d, we store:

$$\mathcal{M}_{\mathrm{str},d}^{S} = \big\{\big(\boldsymbol{z}_{d,i}^{\mathrm{str}}, \boldsymbol{c}_i^{\mathrm{str}}, d\big)\big\}_{i=1}^{N_d}, \qquad \boldsymbol{c}_i^{\mathrm{str}} = \boldsymbol{h}_{\mathrm{base}}(x_i), \tag{7}$$

where $\boldsymbol{z}_{d,i}^{\mathrm{str}}$ supports topology-aware retrieval and $\boldsymbol{c}_i^{\mathrm{str}}$ provides the context representation used for subsequent augmentation.

3.4 Hierarchical Context Retrieval with Utility Calibration

The complementary databases provide source contexts with different degrees of relevance and downstream usefulness (Jiang et al., 2024; Yuan et al., 2026b). We first retrieve compact candidate pools according to semantic or structural relevance and then refine them according to predicted downstream utility, with the utility scorers calibrated entirely through source-only pseudo-target episodes.

Hierarchical Retrieval via Relevance and Utility. Given $\boldsymbol{q} = \boldsymbol{h}_{\text{base}}(x)$, semantic candidates are retrieved directly in the frozen GFM representation space:

$$\sigma_i^{\text{sem}} = \cos(\boldsymbol{q}, \boldsymbol{c}_i^{\text{sem}}), \qquad \mathcal{C}_{\text{sem}}(x) = \underset{i \in \mathcal{M}_{\text{sem}}^S}{\text{TopM}}\ \sigma_i^{\text{sem}}. \tag{8}$$

For structural retrieval, the target local graph is encoded separately in each source-domain structural space, and the resulting within-domain candidates are merged:

$$\sigma_{d,i}^{\text{str}} = K_d\big(G_x^{\text{loc}}, G_{d,i}^{\text{loc}}\big) = \boldsymbol{z}_d(G_x^{\text{loc}})^\top \boldsymbol{z}_{d,i}^{\text{str}}, \qquad \mathcal{C}_{\text{str},d}(x) = \underset{i \in \mathcal{M}_{\text{str},d}^S}{\text{TopM}}\ \sigma_{d,i}^{\text{str}}, \qquad \mathcal{C}_{\text{str}}(x) = \bigcup_d \mathcal{C}_{\text{str},d}(x). \tag{9}$$

The second level evaluates whether these relevant candidates benefit downstream prediction. For a context subset $\mathcal{A}_b$ from branch $b \in \{\text{sem}, \text{str}\}$, its utility is:

$$\mathcal{U}_b(\mathcal{A}_b \mid x, y) = \frac{\mathcal{L}_0(x, y) - \mathcal{L}_b(x, y; \mathcal{A}_b)}{|\mathcal{L}_0(x, y)| + \epsilon_u}, \tag{10}$$

where $\mathcal{L}_0$ and $\mathcal{L}_b$ denote the losses without and with retrieved contexts, respectively. Candidate interactions are represented as:

$$\boldsymbol{e}_i^{\text{sem}} = [\boldsymbol{q}; \boldsymbol{c}_i^{\text{sem}}; |\boldsymbol{q} - \boldsymbol{c}_i^{\text{sem}}|; \boldsymbol{q} \odot \boldsymbol{c}_i^{\text{sem}}], \tag{11}$$

$$\boldsymbol{e}_{d,i}^{\text{str}} = [\boldsymbol{q}; \boldsymbol{c}_i^{\text{str}}; |\boldsymbol{q} - \boldsymbol{c}_i^{\text{str}}|; \boldsymbol{q} \odot \boldsymbol{c}_i^{\text{str}}; \boldsymbol{z}_d(G_x^{\text{loc}}); \boldsymbol{z}_{d,i}^{\text{str}}; |\boldsymbol{z}_d(G_x^{\text{loc}}) - \boldsymbol{z}_{d,i}^{\text{str}}|; \boldsymbol{z}_d(G_x^{\text{loc}}) \odot \boldsymbol{z}_{d,i}^{\text{str}}]. \tag{12}$$

The branch-specific utility scores combine these interactions with first-level relevance:

$$u_i^{\text{sem}} = U_{\boldsymbol{\theta}_{\text{sem}}}([\boldsymbol{e}_i^{\text{sem}}; \sigma_i^{\text{sem}}]) + \lambda_s \sigma_i^{\text{sem}}, \qquad u_{d,i}^{\text{str}} = U_{\boldsymbol{\theta}_{\text{str}}}(\boldsymbol{e}_{d,i}^{\text{str}}) + \lambda_k \sigma_{d,i}^{\text{str}} + \lambda_d p_i, \tag{13}$$

where $p_i = \cos(\boldsymbol{q}, \boldsymbol{\mu}_{d_i})$ and $\boldsymbol{\mu}_d = |\mathcal{D}_d^S|^{-1} \sum_{x_j \in \mathcal{D}_d^S} \boldsymbol{h}_{\text{base}}(x_j)$ measure source-domain compatibility. The top-K candidates are then aggregated as:

$$\mathcal{I}_b(x) = \underset{i \in \mathcal{C}_b(x)}{\text{TopK}}\ u_i^b, \qquad a_i^b = \frac{\exp(u_i^b/\tau_b)}{\sum_{j \in \mathcal{I}_b(x)} \exp(u_j^b/\tau_b)}, \qquad \Delta \boldsymbol{h}_b(x) = \sum_{i \in \mathcal{I}_b(x)} a_i^b \boldsymbol{c}_i^b. \tag{14}$$

For notational simplicity, the domain index of structural candidates is omitted after merging the per-domain pools.

Group Relative Policy for Utility Calibration. We calibrate utility entirely from source-only pseudo-target episodes. In each episode, one source domain $\mathcal{D}_s^S$ is held out from the retrieval databases and split into calibration support and query sets $(\mathcal{S}_s^{\text{cal}}, \mathcal{Q}_s^{\text{cal}})$. A lightweight predictor $g_{\boldsymbol{\omega}_s}$ is fitted on $\mathcal{S}_s^{\text{cal}}$ and fixed while evaluating query contexts. For branch b, the predicted scores define a sequential policy that samples $\mathcal{A}_j = (i_{j,1}, \ldots, i_{j,K})$ without replacement:

$$\pi_{\boldsymbol{\theta}_b}(\mathcal{A}_j \mid x) = \prod_{t=1}^{K} \frac{\exp(u_{i_{j,t}}^b/\tau_b)}{\sum_{i \in \mathcal{C}_b(x) \setminus \mathcal{A}_{j,<t}} \exp(u_i^b/\tau_b)}. \tag{15}$$

For each calibration query, J sampled subsets are aggregated using Eq. (14) and evaluated by the fixed predictor. Their rewards and group-relative advantages are:

$$r_j = \text{clip}(\mathcal{U}_b(\mathcal{A}_j \mid x, y) + \eta[\mathbf{1}(\hat{y}_j = y) - \mathbf{1}(\hat{y}_0 = y)], -r_{\max}, r_{\max}), \qquad A_j = \frac{r_j - \overline{r}}{\sigma_r + \epsilon_{\text{std}}}, \tag{16}$$

where $\hat{y}_0$ and $\hat{y}_j$ denote predictions without and with the sampled contexts. With $\rho_j(\boldsymbol{\theta}_b) = \pi_{\boldsymbol{\theta}_b}(\mathcal{A}_j \mid x)/\pi_{\boldsymbol{\theta}_b^{\text{old}}}(\mathcal{A}_j \mid x)$, the utility scorer is optimized by:

$$\mathcal{L}_{\text{util}}^b = -\frac{1}{J} \sum_{j=1}^{J} \min(\rho_j A_j, \text{clip}(\rho_j, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}) A_j) + \gamma D_{\text{KL}}(\pi_{\boldsymbol{\theta}_b} \| \pi_{\text{ref}}) - \xi \mathcal{H}(\pi_{\boldsymbol{\theta}_b}). \tag{17}$$

Here, π_{ref} is a fixed reference policy, while γ and ξ control KL and entropy regularization. During calibration, the structural encoders and retrieval databases remain fixed; after calibration, the utility scorers are also frozen for target adaptation.

3.5 Graph Representation Enhancement with Null-aware Routing

The retrieved semantic and structural contexts may contribute differently across target instances, motivating a null-aware router that adaptively determines how much each source of information should influence the final representation. To guide this adaptive integration, the router first characterizes the reliability of each retrieval branch. Specifically, for $b \in \mathcal{B} = \{\mathrm{sem}, \mathrm{str}\}$, we summarize the selected contexts using the mean utility μ_b, maximum utility m_b, top-two margin δ_b, and aggregation entropy H_b, while further incorporating the agreement between the semantic and structural branches:

$$\boldsymbol{c}_b(x) = [\mu_b, m_b, \delta_b, H_b], \qquad \boldsymbol{c}(x) = [\boldsymbol{c}_{\mathrm{sem}}(x); \boldsymbol{c}_{\mathrm{str}}(x); \cos(\Delta\boldsymbol{h}_{\mathrm{sem}}, \Delta\boldsymbol{h}_{\mathrm{str}})]. \tag{18}$$

We consider the null representation $\boldsymbol{h}_0 = \boldsymbol{h}_{\mathrm{base}}$ and branch-enhanced representations $\boldsymbol{h}_b = \boldsymbol{h}_{\mathrm{base}} + \Delta\boldsymbol{h}_b$. The router takes:

$$\boldsymbol{r}(x) = [\boldsymbol{h}_{\mathrm{base}}; \Delta\boldsymbol{h}_{\mathrm{sem}}; \Delta\boldsymbol{h}_{\mathrm{str}}; \boldsymbol{c}(x)], \qquad \boldsymbol{\alpha}(x) = \mathrm{softmax}(R_{\boldsymbol{\varphi}}(\boldsymbol{r}(x))), \tag{19}$$

where $\boldsymbol{\alpha} = [\alpha_0, \alpha_{\mathrm{sem}}, \alpha_{\mathrm{str}}]$, and produces:

$$\boldsymbol{h}_{\boldsymbol{\Phi}}(x) = \sum_{z \in \mathcal{Z}} \alpha_z(x)\boldsymbol{h}_z(x), \qquad \mathcal{Z} = \{0, \mathrm{sem}, \mathrm{str}\}. \tag{20}$$

The router is calibrated on the source-only pseudo-target episodes of Section 3.4. For each calibration query, the three routes are evaluated by the fixed episode predictor, with semantic and structural rewards defined as in Eq. (16) and $r_0 = 0$. The reward-preferred route supervises:

$$z^\star = \arg\max_{z \in \mathcal{Z}} r_z, \qquad \mathcal{L}_{\mathrm{route}} = -\log \alpha_{z^\star}. \tag{21}$$

The null route preserves the backbone representation when no retrieved branch is selected, and the calibrated router remains fixed during target adaptation.

3.6 Summarization

Overall, **COLA** follows a stage-wise optimization strategy consisting of structural representation learning, source-only utility calibration, and target adaptation. The objectives involved in source-side preparation can be compactly summarized as:

$$\mathcal{L}_{\mathrm{source}} = \sum_d \mathcal{L}^d_{\mathrm{ker}} + \mathcal{L}^{\mathrm{sem}}_{\mathrm{util}} + \mathcal{L}^{\mathrm{str}}_{\mathrm{util}} + \mathcal{L}_{\mathrm{route}}. \tag{22}$$

These terms are optimized stage-wise rather than jointly. In particular, the structural encoders are first trained and frozen, after which the utility scorers and null-aware router are calibrated with the backbone, structural encoders, and retrieval databases fixed. For an unseen target task, all source-derived components remain frozen, and only the lightweight downstream predictor is optimized on the m-shot support set using $\mathcal{L}_{\mathrm{task}}$.

3.7 Theoretical Analysis

We analyze the representation-level effect of adaptive context routing under a fixed downstream predictor and characterize when the two retrieval branches provide complementary oracle gains. This analysis isolates the routing mechanism by comparing different counterfactual representations under the same prediction head, rather than separately retraining a predictor for each representation. Fix all source-derived components and condition on the target support set. We consider an affine downstream head $g(\boldsymbol{h}) = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}$ shared across the counterfactual representations. Define $\mathcal{Z} = \{0, \mathrm{sem}, \mathrm{str}\}$ and

$$\boldsymbol{h}_0 = \boldsymbol{h}_{\mathrm{base}}, \qquad \boldsymbol{h}_{\mathrm{sem}} = \boldsymbol{h}_{\mathrm{base}} + \Delta\boldsymbol{h}_{\mathrm{sem}}, \qquad \boldsymbol{h}_{\mathrm{str}} = \boldsymbol{h}_{\mathrm{base}} + \Delta\boldsymbol{h}_{\mathrm{str}}.$$

Since the routing weights satisfy $\alpha_0 + \alpha_{\mathrm{sem}} + \alpha_{\mathrm{str}} = 1$, Eq. (20) can be equivalently written as $\boldsymbol{h}_{\boldsymbol{\Phi}} = \sum_{z \in \mathcal{Z}} \alpha_z \boldsymbol{h}_z$. Let $L_z(x, y)$ and $L_{\boldsymbol{\alpha}}(x, y)$ denote the cross-entropy losses of $\boldsymbol{h}_z(x)$ and $\boldsymbol{h}_{\boldsymbol{\Phi}}(x)$ under the same fixed downstream head, respectively, and define $L_*(x, y) = \min_{z \in \mathcal{Z}} L_z(x, y)$. The target oracle gain is:

$$G_T = \mathbb{E}_T[L_0 - L_*] \geq 0,$$

Table 1: Few-shot classification results under the LODO setting. We report mean accuracy (%) with standard deviation. Best results are presented in **bold** and the runners-up are underlined.

Setting	LODO-Dataset						LODO-Domain			
Target Dataset	Cora		CiteSeer		PubMed		OGBN-Products		Wiki-CS	
m-shot	1	5	1	5	1	5	1	5	1	5
Method	**Node Classification**									
GCN (▷ *ICLR'17*)	28.4 ±4.6	50.2 ±4.9	29.3 ±3.4	45.9 ±5.4	40.3 ±6.9	50.7 ±7.5	44.7 ±4.3	48.1 ±3.4	37.2 ±5.1	48.1 ±4.9
GAT (▷ *ICLR'18*)	29.7 ±5.2	49.0 ±7.9	29.3 ±3.5	46.1 ±5.1	40.5 ±4.0	52.2 ±6.3	44.6 ±4.0	48.1 ±4.5	37.9 ±4.5	48.6 ±4.5
DGI (▷ *ICLR'19*)	30.8 ±3.9	49.9 ±6.6	31.4 ±4.1	46.5 ±7.1	40.0 ±5.9	53.6 ±7.1	46.0 ±5.4	50.1 ±4.2	38.1 ±5.1	49.2 ±4.4
GraphCL (▷ *NeurIPS'20*)	33.6 ±5.8	53.2 ±5.4	28.2 ±3.1	48.8 ±7.7	39.0 ±8.7	54.7 ±4.4	46.1 ±5.0	50.5 ±4.6	40.0 ±4.0	50.1 ±5.2
GCOPE (▷ *KDD'24*)	36.3 ±3.9	55.6 ±6.4	40.4 ±4.6	56.9 ±5.8	44.8 ±4.7	53.6 ±8.6	47.7 ±4.9	51.4 ±3.3	45.8 ±5.5	53.5 ±4.7
MDGPT (▷ *arXiv'24*)	42.6 ±6.8	62.7 ±6.0	37.9 ±7.2	55.9 ±3.3	51.0 ±9.0	58.7 ±6.2	49.1 ±6.0	56.6 ±2.7	45.0 ±4.8	54.1 ±5.2
SAMGPT (▷ *WWW'25*)	46.8 ±6.5	64.6 ±6.7	38.7 ±6.4	56.4 ±4.7	51.9 ±9.5	59.1 ±6.0	49.8 ±4.4	56.2 ±3.3	44.4 ±5.5	54.4 ±5.8
MDGFM (▷ *ICML'25*)	47.4 ±6.3	66.0 ±6.5	36.3 ±6.2	55.8 ±4.0	50.2 ±8.8	58.4 ±6.4	48.5 ±4.7	54.7 ±4.9	43.4 ±5.8	53.9 ±4.4
OFA (▷ *ICLR'24*)	45.9 ±6.3	67.7 ±2.9	38.0 ±7.6	52.8 ±6.4	46.3 ±6.0	56.0 ±5.9	49.1 ±5.7	55.3 ±4.2	42.8 ±4.6	54.3 ±4.0
ZeroG (▷ *KDD'24*)	51.8 ±5.6	71.4 ±1.7	39.7 ±5.9	54.6 ±2.0	53.1 ±3.5	63.0 ±3.5	53.2 ±2.9	59.9 ±3.1	46.1 ±3.4	59.0 ±2.0
GraphCLIP (▷ *WWW'25*)	53.9 ±5.3	73.1 ±2.9	40.6 ±3.4	55.2 ±1.2	56.8 ±1.9	65.2 ±2.8	53.4 ±6.1	62.6 ±4.3	45.5 ±2.1	59.9 ±2.8
UniGraph (▷ *KDD'25*)	56.1 ±6.3	74.8 ±1.9	40.5 ±3.1	56.3 ±2.2	57.0 ±3.1	66.8 ±2.4	53.8 ±3.4	61.0 ±3.5	45.1 ±3.6	58.4 ±3.1
RAG-GFM (▷ *WWW'26*)	58.4 ±6.0	76.1 ±0.7	41.5 ±3.0	57.7 ±1.8	**59.2** ±2.7	**68.7** ±1.8	55.4 ±7.6	64.2 ±4.4	47.8 ±3.8	60.9 ±2.5
COLA (Ours)	**59.7** ±5.9	**77.0** ±2.3	**41.8** ±3.2	**57.8** ±1.5	**59.2** ±2.8	67.4 ±1.8	**57.4** ±6.4	**66.5** ±2.6	**47.9** ±3.7	**62.0** ±3.0
Method	**Ego-Graph Classification**									
GCN (▷ *ICLR'17*)	40.1 ±4.8	52.9 ±4.1	29.5 ±5.7	43.9 ±5.9	45.3 ±7.3	55.4 ±5.3	47.6 ±3.2	52.6 ±5.3	38.9 ±4.1	41.5 ±6.4
GAT (▷ *ICLR'18*)	36.0 ±5.1	49.6 ±5.1	26.0 ±7.8	45.3 ±7.3	41.0 ±5.8	54.5 ±7.3	49.2 ±5.6	52.9 ±5.5	38.3 ±4.6	41.1 ±3.5
InfoGraph (▷ *ICLR'20*)	42.2 ±5.2	54.7 ±4.9	30.2 ±4.1	47.2 ±5.2	49.1 ±5.4	59.7 ±7.1	50.7 ±4.3	53.8 ±5.1	40.4 ±4.3	42.4 ±5.0
GraphCL (▷ *NeurIPS'20*)	39.6 ±5.8	55.2 ±5.9	32.6 ±6.5	46.4 ±3.8	47.7 ±7.0	60.0 ±5.4	51.7 ±5.9	53.0 ±5.2	40.8 ±4.6	42.5 ±4.3
GCOPE (▷ *KDD'24*)	55.9 ±7.4	63.9 ±4.8	41.0 ±9.0	58.2 ±5.8	54.4 ±8.6	66.4 ±3.7	55.8 ±4.3	57.7 ±4.8	42.2 ±5.8	49.8 ±3.5
MDGPT (▷ *arXiv'24*)	52.8 ±6.7	65.1 ±4.2	41.0 ±9.7	59.3 ±6.0	55.5 ±8.3	67.6 ±4.6	54.5 ±4.8	60.5 ±3.4	43.2 ±6.2	48.9 ±4.2
SAMGPT (▷ *WWW'25*)	53.3 ±4.3	69.3 ±3.4	42.4 ±7.3	62.4 ±5.7	57.7 ±6.3	68.0 ±4.6	54.4 ±3.2	60.8 ±4.8	43.5 ±5.6	48.3 ±5.7
MDGFM (▷ *ICML'25*)	55.5 ±5.4	69.4 ±2.1	43.4 ±6.4	60.8 ±5.1	56.0 ±5.1	67.1 ±5.1	54.7 ±2.1	59.8 ±5.3	41.8 ±6.7	46.4 ±3.2
OFA (▷ *ICLR'24*)	58.0 ±3.7	65.1 ±3.9	45.4 ±6.6	60.0 ±6.6	59.7 ±4.3	67.2 ±3.4	56.0 ±3.9	60.1 ±3.5	42.4 ±5.8	48.1 ±2.2
ZeroG (▷ *KDD'24*)	65.1 ±2.2	74.2 ±1.6	50.3 ±5.7	64.0 ±5.1	61.4 ±4.0	70.2 ±1.3	58.5 ±4.0	66.2 ±3.9	46.1 ±5.9	57.8 ±3.6
GraphCLIP (▷ *WWW'25*)	65.9 ±3.7	75.1 ±2.2	50.4 ±4.0	63.0 ±3.3	60.7 ±3.8	71.3 ±2.2	58.6 ±2.2	65.7 ±2.2	46.0 ±3.3	58.8 ±4.4
UniGraph (▷ *KDD'25*)	66.5 ±2.5	76.5 ±1.0	50.9 ±4.4	64.0 ±2.4	61.5 ±2.6	71.4 ±2.3	58.1 ±4.0	66.0 ±3.8	47.0 ±2.5	58.9 ±2.2
RAG-GFM (▷ *WWW'26*)	68.7 ±1.5	78.4 ±0.6	52.2 ±6.1	65.5 ±2.2	**62.4** ±2.1	**71.5** ±1.9	60.2 ±4.2	68.0 ±3.1	48.1 ±1.1	62.0 ±4.3
COLA (Ours)	**68.8** ±3.1	**78.6** ±1.7	**53.8** ±7.3	**65.9** ±3.2	61.6 ±3.0	71.2 ±1.5	**62.5** ±3.4	**70.7** ±2.6	**48.5** ±2.4	**63.2** ±3.8

which measures the maximum loss reduction available from the three routing options under this fixed-head comparison.

Let $\boldsymbol{V}$ denote the router input and $Z^\star = \arg\max_{z\in\mathcal{Z}} r_z$ the reward-preferred routing option, with fixed tie-breaking. Source rewards are computed using the episode predictors introduced in source-only calibration. For theoretical analysis, target rewards are analogously defined using the fixed target predictor; these quantities are used only for analysis and are never accessed during target adaptation. Let P_S and P_T denote the source and target distributions of $(\boldsymbol{V}, Z^\star)$. Define:

$$C_S = \mathbb{E}_S[-\log \alpha_{Z^\star}(\boldsymbol{V})], \qquad d_{ST} = \mathrm{TV}(P_S, P_T), \qquad \Xi_T = \mathbb{E}_T[L_{Z^\star} - L_*]. \tag{23}$$

Here, C_S denotes the source population routing loss, d_{ST} measures the joint shift in router inputs and reward-preferred options, and Ξ_T measures reward misalignment with cross-entropy minimization.

Theorem 3.1. *Under the preceding setup, suppose all target risks are finite and*

$$\max_{z\in\mathcal{Z}} L_z(x,y) - \min_{z\in\mathcal{Z}} L_z(x,y) \le B$$

for some $B < \infty$, target-almost surely. Then:

$$R_T(\boldsymbol{\alpha}) - R_T(0) \le -G_T + \Xi_T + B\min\{1,\, 1 - e^{-C_S} + d_{ST}\}, \tag{24}$$

where $R_T(\boldsymbol{\alpha}) = \mathbb{E}_T[L_{\boldsymbol{\alpha}}]$ and $R_T(0) = \mathbb{E}_T[L_0]$.

Theorem 3.1 shows that adaptive routing improves over the null route under the same fixed head when the oracle gain G_T dominates reward mismatch Ξ_T, source routing error C_S, and source–target discrepancy d_{ST}. The two branches are further complementary when each provides positive gains on instances where the other is less effective, thereby enlarging the joint oracle gain. Proofs are provided in Appendix A.

4 Experiments

4.1 Experimental Settings

Datasets. We conduct experiments on five benchmark text-attributed graph datasets covering three different domains. Specifically, the ***citation domain*** contains Cora (McCallum et al., 2000), Cite-

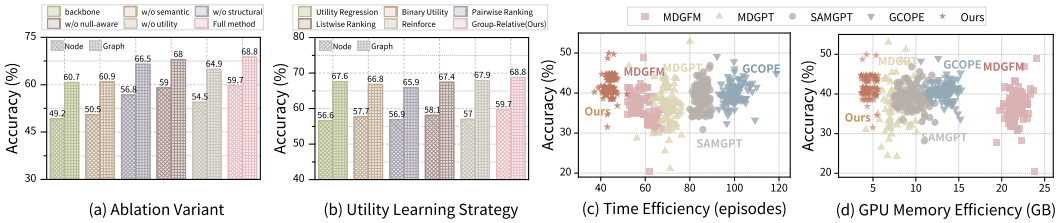


Figure 2: Ablation and adaptation-efficiency analysis under the LODO-Dataset, including component ablations on Cora and adaptation episodes and peak GPU memory comparisons on CiteSeer.

Seer (Giles et al., 1998), and PubMed (Sen et al., 2008); the ***e-commerce domain*** includes OGBN-Products (Hu et al., 2020), a large-scale product co-purchase network with multiple product categories; and the *web-link domain* contains Wiki-CS (Mernyei & Cangea, 2020), a hyperlink network constructed from Wikipedia pages. These datasets form the basis for evaluating cross-dataset and cross-domain transfer under the LODO protocols.

Baselines. We compare `COLA` with 14 representative baselines from four categories. Vanilla GNNs include GCN (Kipf & Welling, 2017) and GAT (Velickovic et al., 2018), trained without graph pre-training. Graph pre-training methods include DGI (Velickovic et al., 2019), InfoGraph (Sun et al., 2020), and GraphCL (You et al., 2020). Text-free graph foundation models (GFMs) include GCOPE (Zhao et al., 2024), MDGPT (Yu et al., 2024b), SAMGPT (Yu et al., 2025), and MDGFM (Wang et al., 2025). Text-attributed GFMs include OFA (Liu et al., 2024), ZeroG (Li et al., 2024), GraphCLIP (Zhu et al., 2025), UniGraph (He et al., 2025), and RAG-GFM (Yuan et al., 2026b). Together, these baselines cover conventional GNNs, graph pre-training approaches, and recent graph foundation models under the same transfer protocols.

Pre-training and Fine-tuning Settings. We evaluate node and graph classification under the m-shot setting, with m labeled instances per class for target adaptation. For graph classification, each graph instance is an ego-graph assigned the label of its center node. We consider two leave-one-out transfer settings. In *LODO-Dataset*, one dataset is held out as the target and the others are used for source pre-training. In *LODO-Domain*, all datasets from the target semantic domain are excluded from source pre-training, imposing a stronger domain shift without same-domain source graphs.

4.2 Performance Comparison

Transfer Performance. We compare `COLA` with representative GNN, graph pre-training, and graph foundation model baselines on few-shot node and ego-graph classification under both LODO-Dataset and LODO-Domain settings. The results are reported in Table 1. We make the following observations. ❶ *Overall performance.* `COLA` achieves the best or tied-best results in 17 of the 20 evaluation settings, demonstrating consistent effectiveness across different target datasets, task formulations, and supervision levels. This advantage indicates that utility-guided retrieval can provide transferable source contexts beyond the representations encoded by the frozen backbone. ❷ *Cross-domain transfer.* Under the more challenging LODO-Domain setting, `COLA` achieves the best results in all eight settings on OGBN-Products and Wiki-CS, where source graphs from the target semantic domain are excluded during pre-training. This result suggests that combining semantic and structural evidence is particularly beneficial when same-domain source knowledge is unavailable. ❸ *Gains under domain shift.* On OGBN-Products, `COLA` improves over the strongest baseline by 2.0 and 2.3 percentage points under one-shot and five-shot node classification, respectively, and by 2.3 and 2.7 percentage points for ego-graph classification. The larger improvements in this setting further indicate the benefit of explicitly retrieving and refining complementary source contexts under substantial distribution shifts.

Target Adaptation Cost. We compare `COLA` with four representative GFMs on CiteSeer under the LODO-Dataset setting in terms of target adaptation episodes and peak GPU memory, excluding one-time source-side preparation. The results are shown in Figures 2(c) and 2(d). ❶ *Adaptation efficiency.* `COLA` reaches a comparable accuracy level with fewer target adaptation episodes, indicating that the fixed retrieval-enhanced representation can be effectively exploited with limited target-side optimization. ❷ *Memory efficiency.* `COLA` incurs the lowest peak GPU memory among the compared methods, which follows from freezing the backbone and all source-derived retrieval and routing components and optimizing only the lightweight downstream predictor.

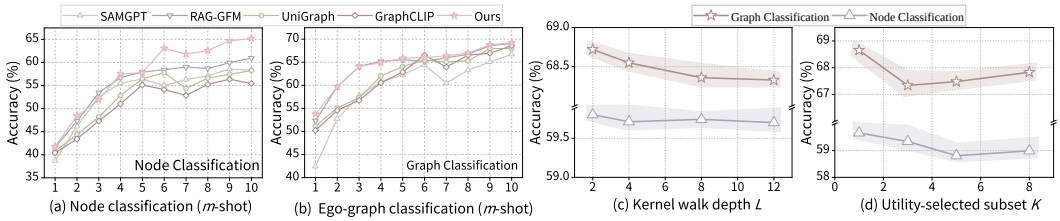


Figure 3: Label efficiency and hyperparameter sensitivity of COLA under the LODO-Dataset setting.

4.3 Further Analysis

Ablation Study. We compare the full COLA with five variants, including the vanilla backbone, semantic-only retrieval, structural-only retrieval, removal of utility-based refinement, and fixed routing. As shown in Figure 2(a), both semantic and structural retrieval improve over the backbone, while their combination performs best, supporting the complementarity of the two context sources. Removing utility-based refinement causes a clear degradation, indicating that retrieval relevance alone is insufficient for selecting useful contexts. Fixed routing further reduces performance, showing the benefit of adapting semantic, structural, and null contributions to individual queries.

Utility Learning Strategy. We compare Regression, Binary Utility, Pairwise Ranking, Listwise Ranking, REINFORCE, and our Group-Relative strategy under identical candidate pools and training budgets. As shown in Figure 2(b), Regression and Binary Utility are consistently weaker, suggesting that absolute utility prediction provides limited guidance for context selection. Pairwise and Listwise Ranking better exploit relative preferences among sampled context sets, while policy-based learning achieves stronger performance. In particular, Group-Relative performs best on both node and ego-graph classification, indicating the benefit of learning from within-query relative rewards.

Utility Effectiveness. We analyze whether predicted utility reflects downstream usefulness and whether the two retrieval branches are complementary. As shown in Figure 4, higher predicted utility generally corresponds to larger downstream gains in both branches, supporting utility-guided context refinement. Moreover, 38.5% of queries exhibit branch-specific benefits and at least one branch improves 87.0% of queries, confirming their complementarity and motivating instance-specific routing.

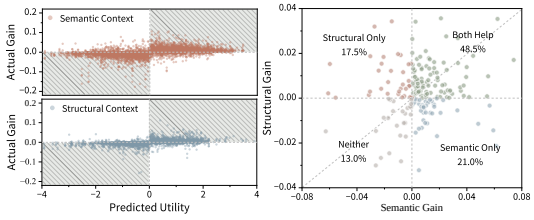


Figure 4: Utility effectiveness and semantic–structural complementarity on Cora.

Performance on Different Numbers of Shots. We compare COLA with various baselines on CiteSeer by varying the number of labeled examples per class from one to ten. As shown in Figures 3(a) and 3(b), COLA consistently benefits from additional supervision, with a more pronounced advantage for node classification when more than five labeled examples are available and the best ego-graph performance in most settings. The improvement gradually stabilizes with increasing supervision, suggesting that the fixed representations can effectively exploit additional target labels.

Hyperparameter Sensitivity. We evaluate the sensitivity of COLA to the context size $K \in \{1, 3, 5, 8\}$ and random-walk depth $L \in \{2, 4, 8, 12\}$ on Cora under the LODO-Dataset setting. Figures 3(c) and 3(d) shows stable performance across both representative ranges. Increasing the context size yields only marginal changes, while varying the random-walk depth causes limited fluctuations, demonstrating that COLA is consistently robust and not sensitive to either hyperparameter.

5 Conclusion

We propose COLA, a calibrated framework for few-shot graph transfer that augments a frozen GFM with explicit source contexts. The method externalizes source knowledge through complementary semantic and structural databases, distinguishes retrieval relevance from downstream utility through hierarchical context selection, and adaptively integrates retrieved evidence using null-aware routing. All retrieval and routing components are calibrated using source-only pseudo-target episodes and remain fixed during target adaptation. Extensive experiments demonstrate consistent improvements, strong label efficiency, and robustness to hyperparameters, while further analyses verify the effectiveness of utility calibration and the complementarity of semantic and structural contexts.

AI USE STATEMENT

In this work, we did not use generative AI tools to generate synthetic datasets, develop theoretical or conceptual frameworks, formulate mathematical claims or proofs, propose hypotheses, design research methodology or experiments, or interpret experimental results. In addition to our method which is required to utilize the LLM, we used generative AI tools to assist with editing the manuscript for grammar, and readability, and to generate or refine portions of software code used in our experiments. All AI-assisted text was carefully reviewed and revised by the authors to ensure technical accuracy and consistency with the intended meaning. AI-generated or AI-assisted code was manually inspected, tested, and validated by the authors before use. We take full responsibility for the final content of this work, including any text or code produced with the aid of generative AI.

REFERENCES

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.

Zihao Guo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, Min Zhou, Yisen Gao, and Jianxin Li. Graphmore: Mitigating topological heterogeneity via mixture of riemannian experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11754–11762, 2025.

Zihao Guo, Qingyun Sun, Ziwei Zhang, Haonan Yuan, Huiping Zhuang, Xingcheng Fu, and Jianxin Li. Graphkeeper: Graph domain-incremental learning via knowledge disentanglement and preservation. *Advances in Neural Information Processing Systems*, 38:1145–1171, 2026.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

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.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM web conference 2023*, pp. 737–746, 2023.

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.

Xinke Jiang, Rihong Qiu, Yongxin Xu, Wentao Zhang, Yichen Zhu, Ruizhe Zhang, Yuchen Fang, Xu Chu, Junfeng Zhao, and Yasha Wang. Ragraph: A general retrieval-augmented graph learning framework. *Advances in Neural Information Processing Systems*, 37:29948–29985, 2024.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag. In *International Conference on Learning Representations*, volume 2025, pp. 37784–37822, 2025.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *International conference on learning representations*, volume 2024, pp. 20188–20210, 2024.

Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM web conference 2023*, pp. 417–428, 2023.

Jiayi Luo, Qingyun Sun, Lingjuan Lyu, Ziwei Zhang, Haonan Yuan, Xingcheng Fu, and Jianxin Li. Towards effective, stealthy, and persistent backdoor attacks targeting graph foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24142–24150, 2026a.

Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: graph foundation model for retrieval augmented generation. *Advances in Neural Information Processing Systems*, 38:36371–36405, 2026b.

Hiren Madhu, João Felipe Rocha, Tinglin Huang, Siddharth Viswanath, Smita Krishnaswamy, and Rex Ying. Heist: a graph foundation model for spatial transcriptomics and proteomics data. In *International Conference on Learning Representations*, volume 2026, pp. 65453–65476, 2026.

Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. *arXiv preprint arXiv:2402.02216*, 2024.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

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

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Chuan Shi, Junze Chen, Jiawei Liu, and Cheng Yang. Graph foundation model. *Frontiers of Computer Science*, 18(6):186355, 2024.

Junhua Shi, Qingyun Sun, Haonan Yuan, and Xingcheng Fu. Sa^2gfm: Enhancing robust graph foundation models with structure-aware semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15716–15724, 2026.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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.

Fangzheng Tian, Debasis Ganguly, and Craig Macdonald. Is relevance propagated from retriever to generator in rag? In *European Conference on Information Retrieval*, pp. 32–48. Springer, 2025.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems*, 37:107403–107443, 2024a.

Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Towards graph foundation models: Learning generalities across graphs via task-trees. *arXiv preprint arXiv:2412.16441*, 2024b.

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

Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16578–16586, 2024a.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024b.

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.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. Graver: Generative graph vocabularies for robust graph foundation models fine-tuning. *Advances in Neural Information Processing Systems*, 38:16639–16684, 2026a.

Haonan Yuan, Qingyun Sun, Jiacheng Tao, Xingcheng Fu, and Jianxin Li. Rag-gfm: Overcoming in-memory bottlenecks in graph foundation models via retrieval-augmented generation. In *Proceedings of the ACM Web Conference 2026*, pp. 626–637, 2026b.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4443–4454, 2024.

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, 2025.

A Proofs for the Theoretical Analysis

All maps are assumed measurable. We condition throughout on the calibrated source-derived retrieval and routing components and the target support set. Source expectations refer to the population evaluation distribution induced by pseudo-target episodes for this fixed system, rather than to the empirical samples used to fit the router. Each source episode uses its own support-fitted predictor and episode-specific graph databases constructed from the remaining source domains, whereas the target uses its support-fitted predictor and the calibrated source graph databases. In each case, $Z^\star$ is defined from the corresponding rewards using the same tie-breaking rule. On the target domain, these rewards and $Z^\star$ are introduced only for theoretical analysis and are never used during adaptation. Thus, d_{ST} captures differences in both router inputs and reward-preferred routing options. All

target losses are evaluated using the same fixed predictor. Let $\mathcal{V}$ denote the space of router inputs. We use

$$\mathrm{TV}(P, Q) = \sup_A |P(A) - Q(A)|.$$

Proof of Theorem 3.1. Because the prediction head is affine and the routing weights sum to one,

$$\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{\Phi}} + \boldsymbol{b} = \sum_{z \in \mathcal{Z}} \alpha_z (\boldsymbol{W}\boldsymbol{h}_z + \boldsymbol{b}). \tag{25}$$

Cross-entropy is convex in its logit vector. Let $\boldsymbol{p}$ denote the corresponding softmax vector. Its Hessian is $\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top$, and for any vector $\boldsymbol{v}$,

$$\boldsymbol{v}^\top \left(\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top\right) \boldsymbol{v} = \sum_j p_j v_j^2 - \left(\sum_j p_j v_j\right)^2 \geq 0.$$

Jensen's inequality applied to Eq. (25) therefore gives

$$L_{\boldsymbol{\alpha}} \leq \sum_{z \in \mathcal{Z}} \alpha_z L_z. \tag{26}$$

For a fixed labeled instance, let $\zeta = L_{Z^\star} - L_*$. By the assumed loss-oscillation bound,

$$\begin{aligned} \sum_{z \in \mathcal{Z}} \alpha_z L_z - L_* &= \zeta + \sum_{z \neq Z^\star} \alpha_z (L_z - L_{Z^\star}) \\ &\leq \zeta + B(1 - \alpha_{Z^\star}). \end{aligned} \tag{27}$$

Combining Eqs. (26) and (27), and taking target expectations, yields

$$R_T(\boldsymbol{\alpha}) \leq R_T(0) - G_T + \Xi_T + B\mathbb{E}_T[1 - \alpha_{Z^\star}(\boldsymbol{V})]. \tag{28}$$

For any measurable $f : \mathcal{V} \times \mathcal{Z} \to [0, 1]$, the layer-cake identity gives

$$\begin{aligned} |\mathbb{E}_P f - \mathbb{E}_Q f| &= \left| \int_0^1 \{P(f > t) - Q(f > t)\}\, dt \right| \\ &\leq \mathrm{TV}(P, Q). \end{aligned}$$

Applying this inequality to $f(\boldsymbol{V}, Z) = 1 - \alpha_Z(\boldsymbol{V})$ gives

$$\mathbb{E}_T[1 - \alpha_{Z^\star}(\boldsymbol{V})] \leq \mathbb{E}_S[1 - \alpha_{Z^\star}(\boldsymbol{V})] + d_{ST}. \tag{29}$$

For $C_S < \infty$, Jensen's inequality further implies

$$\mathbb{E}_S[\alpha_{Z^\star}(\boldsymbol{V})] = \mathbb{E}_S\left[\exp\{\log \alpha_{Z^\star}(\boldsymbol{V})\}\right] \geq \exp\left\{\mathbb{E}_S[\log \alpha_{Z^\star}(\boldsymbol{V})]\right\} = e^{-C_S}. \tag{30}$$

The conclusion is immediate for $C_S = \infty$ by taking $e^{-\infty} = 0$. Since $1 - \alpha_{Z^\star} \leq 1$, Eqs. (29) and (30) yield

$$\mathbb{E}_T[1 - \alpha_{Z^\star}(\boldsymbol{V})] \leq \min\{1,\, 1 - e^{-C_S} + d_{ST}\}.$$

Substituting this result into Eq. (28) proves Eq. (24). □

A sufficient condition for bounded loss oscillation. The theorem does not require cross-entropy to be globally bounded. Suppose $\|\boldsymbol{W}\|_{\mathrm{op}} \leq M_W$ and $\|\Delta\boldsymbol{h}_{\mathrm{text}}(x)\|_2, \|\Delta\boldsymbol{h}_{\mathrm{str}}(x)\|_2 \leq D$ target-almost surely. For logits $\boldsymbol{v}$ and label y, the cross-entropy gradient is $\nabla_{\boldsymbol{v}}\ell(\boldsymbol{v}, y) = \boldsymbol{p}(\boldsymbol{v}) - \boldsymbol{e}_y$ and satisfies

$$\|\nabla_{\boldsymbol{v}}\ell(\boldsymbol{v}, y)\|_2^2 = (1 - p_y)^2 + \sum_{j \neq y} p_j^2 \leq 2(1 - p_y)^2 \leq 2.$$

The line-segment integral formula therefore gives $|\ell(\boldsymbol{v}, y) - \ell(\boldsymbol{v}', y)| \leq \sqrt{2}\|\boldsymbol{v} - \boldsymbol{v}'\|_2$. Any two representations among $\boldsymbol{h}_0$, $\boldsymbol{h}_{\mathrm{text}}$, and $\boldsymbol{h}_{\mathrm{str}}$ differ by at most $2D$. Hence, one may take

$$B = 2\sqrt{2} M_W D. \tag{31}$$

Branch complementarity. Let $g_{\text{text}} = L_0 - L_{\text{text}}$ and $g_{\text{str}} = L_0 - L_{\text{str}}$. The textual-only and structural-only oracle gains are respectively

$$G_{\text{text}} = \mathbb{E}_T[\max\{0, g_{\text{text}}\}], \qquad G_{\text{str}} = \mathbb{E}_T[\max\{0, g_{\text{str}}\}],$$

whereas the joint oracle gain is

$$G_T = \mathbb{E}_T[\max\{0, g_{\text{text}}, g_{\text{str}}\}].$$

Therefore,

$$G_T - G_{\text{text}} = \mathbb{E}_T\left[\left(g_{\text{str}} - \max\{0, g_{\text{text}}\}\right)_+\right],$$

and symmetrically,

$$G_T - G_{\text{str}} = \mathbb{E}_T\left[\left(g_{\text{text}} - \max\{0, g_{\text{str}}\}\right)_+\right].$$

Hence, $G_T > G_{\text{text}}$ whenever

$$\Pr_T\left(g_{\text{str}} > \max\{0, g_{\text{text}}\}\right) > 0,$$

and $G_T > G_{\text{str}}$ whenever

$$\Pr_T\left(g_{\text{text}} > \max\{0, g_{\text{str}}\}\right) > 0.$$

Thus, the textual and structural contexts provide strictly complementary oracle gains when both conditions hold.

B Algorithms and Training Procedure

We summarize the source-only preparation and target few-shot adaptation of `COLA` in Algorithms 1 and 2, respectively. For each held-out target, all source-stage components are trained exclusively on the source graphs permitted by the corresponding leave-one-out protocol.

Algorithm 1 Source-only preparation and calibration of `COLA`

Require: Held-out target t; permitted source graphs $\mathcal{D}_{-t}$ with source labels
Ensure: Frozen source artifacts $\mathcal{A}_t$

Initialize the source backbone, structural encoders, utility scorers, and null-aware router
// Source Backbone and Database Construction
Train the source backbone $f_{\boldsymbol{\theta}}^{(t)}$ on $\mathcal{D}_{-t}$
for all $G_d \in \mathcal{D}_{-t}$ **do**
    Encode source instances and construct semantic database $\mathcal{M}_{\text{sem}}^{(t)}$
    Construct rooted local graph views and train the structural encoder E_d
    Encode source local graphs and construct structural database $\mathcal{M}_{\text{str},d}^{(t)}$
end for
Freeze the source backbone, structural encoders, and retrieval databases
// Source-Only Utility and Router Calibration
for all pseudo-target graphs $G_d \in \mathcal{D}_{-t}$ **do**
    Use $\mathcal{D}_{-t} \setminus \{G_d\}$ as the retrieval source and sample disjoint support and query sets $(\mathcal{S}_d, \mathcal{Q}_d)$ from G_d
    Construct a frozen support-based predictor using $\mathcal{S}_d$
    for all calibration batches $B \subseteq \mathcal{Q}_d$ **do**
        Retrieve semantic and structural candidate pools
        Update the utility scorers using downstream rewards
        Update the null-aware router from null, semantic, and structural preferences
    end for
end for
Freeze the utility scorers and null-aware router
Save $\mathcal{A}_t = \{f_{\boldsymbol{\theta}}^{(t)}, \mathcal{M}_{\text{sem}}^{(t)}, \{\mathcal{M}_{\text{str},d}^{(t)}\}, \{E_d\}, U_{\text{sem}}^{(t)}, U_{\text{str}}^{(t)}, R^{(t)}\}$

Table 2: Statistics of the processed benchmark graphs. Home and Tech denote two components of Ogbn-Products.

Graph	Domain	Nodes	Edges	Feature dim.	Classes
Cora	Citation	2,708	5,429	1,433	7
CiteSeer	Citation	3,186	4,277	3,703	6
PubMed	Citation	19,717	44,338	500	3
Wiki-CS	Web link	11,701	216,123	300	10
Ogbn-Products (Home)	E-commerce	9,790	131,841	100	5
Ogbn-Products (Tech)	E-commerce	47,428	2,077,241	100	3

Algorithm 2 Target few-shot adaptation and inference with `COLA`

Require: Target graph G_t; labeled m-shot support set $\mathcal{S}_t$; evaluation instances $\mathcal{Q}_t$; frozen source artifacts $\mathcal{A}_t$
Ensure: Predictions for $\mathcal{Q}_t$
// Utility-Aware Evidence Augmentation
for all $x \in \mathcal{S}_t \cup \mathcal{Q}_t$ **do**
  Encode the target instance with the frozen backbone: $h_b(x) \leftarrow f_\theta^{(t)}(x)$
  Retrieve semantic and structural candidate pools from $\mathcal{M}_{\text{sem}}^{(t)}$ and $\mathcal{M}_{\text{str}}^{(t)}$
  Re-rank candidates using $U_{\text{sem}}^{(t)}$ and $U_{\text{str}}^{(t)}$
  Aggregate the selected evidence into $\Delta h_{\text{sem}}(x)$ and $\Delta h_{\text{str}}(x)$
  Compute null-aware routing weights $(\alpha_0, \alpha_{\text{sem}}, \alpha_{\text{str}}) \leftarrow R^{(t)}(\cdot)$
  Fuse the retrieved evidence:
    $h'(x) \leftarrow h_b(x) + \alpha_{\text{sem}} \Delta h_{\text{sem}}(x) + \alpha_{\text{str}} \Delta h_{\text{str}}(x)$
end for
// Few-Shot Adaptation and Prediction
Fit the downstream node- or graph-level readout using $\{(h'(x), y_x) : x \in \mathcal{S}_t\}$
Predict labels for all $x \in \mathcal{Q}_t$ using the fitted readout
Evaluate predictions after inference

The structural encoders are trained during source-side preparation and remain fixed throughout pseudo-target calibration. During calibration, only the semantic and structural utility scorers and the null-aware router are optimized. For each pseudo-target episode, the held-out source domain is excluded from the retrieval databases used by that episode, while the source backbone remains the one pretrained on all source domains permitted by the corresponding LODO protocol. At target time, all source-stage components are frozen and only the lightweight downstream predictor is fitted using the m-shot support set.

C Experiment Details

C.1 Datasets and Evaluation Protocols

- **Citation Domain:** This domain includes Cora, CiteSeer, and PubMed. In these graphs, research papers are treated as nodes and citation relationships as edges. Node attributes are constructed from textual information associated with the papers, such as titles and abstracts.
- **E-Commerce Domain:** We use the Home and Tech components extracted from Ogbn-Products. Each graph models Amazon products as nodes and co-purchase relations as edges, while the prediction labels indicate product categories.
- **Web Link Domain:** Wiki-CS is used as the representative web-link graph. Its nodes correspond to Wikipedia pages and its edges represent hyperlinks between pages. Node features are derived from the textual content of the corresponding articles.

Table 2 summarizes the statistics of the processed graphs.

Pre-training. For each held-out target, we independently train a source-only backbone using the graphs permitted by the corresponding LODO protocol. A 50-dimensional projection of the raw node attributes is concatenated with a 50-dimensional BERT-based text representation, yielding a 100-dimensional model input. The backbone is trained for up to 10,000 epochs using Adam with a learning rate of 10^{-5} and an early-stopping patience of 50 epochs. We subsequently train the graph-kernel encoders, construct the semantic and structural memories, and calibrate the utility scorers and null-aware router through leave-one-source-out pseudo-target episodes. All source-stage components are trained exclusively with source data, without access to labels from the held-out target.

Fine-tuning. During target adaptation, the source backbone, graph-kernel encoders, retrieval memories, utility scorers, and null-aware router are kept frozen. For each query, `COLA` retrieves up to 32 semantic candidates and 16 structural candidates, from which the five candidates with the highest estimated utility are retained in each branch. Only the lightweight node- or graph-level downstream module is optimized using the labeled m-shot support set. We use Adam for up to 400 adaptation episodes with an early-stopping patience of 20 episodes. Target evaluation labels are accessed only after the predictions have been produced.

Environment. All source-stage training, downstream evaluation, and auxiliary experiments are conducted on a single NVIDIA A100-SXM4 GPU with 80 GB of memory. The software environment consists of Python 3.10, PyTorch 2.7.1 with CUDA 12.8, and PyTorch Geometric 2.6.1. All methods are evaluated on the same 10 fixed support splits for each setting, and we report the mean accuracy and sample standard deviation across these splits.