Unifying Multi-View Knowledge for Graph Learning via Model Collaboration

Zhihao Wu[1], Jielong Lu[1], Zihan Fang[2], Jinyu Cai[3], Guangyong Chen[4], Jiajun Bu[1], Haishuai Wang[1]*

[1]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]College of Computer and Data Science, Fuzhou University, Fuzhou, China
[3]Institute of Data Science, National University of Singapore, Singapore
[4]Hangzhou Institute of Medicine, Chinese Academy of Sciences, Hangzhou, China
zhihaowu1999@gmail.com, jielonglu2022@163.com, fzihan11@163.com, jinyucai1995@gmail.com, chenguangyong@him.cas.cn, bjj@zju.edu.cn, haishuai.wang@zju.edu.cn

Abstract

With the increasing scale and complexity of graph data, node attributes are also becoming richer and more complex, particularly in the form of informative text. Classic GNNs equipped with shallow attribute encoders are no longer sufficient to handle such data independently, making model collaboration across heterogeneous architectures an inevitable trend. Recently, the integration of Large Language Models (LLMs) and GNNs has attracted significant attention, yet the inherent disparity between these models remains a key challenge. Promising solutions have considered fine-tuning Small Language Models (SLMs) to bridge the gap between GNNs and frozen LLMs. However, this introduces another problem: these heterogeneous models bring complementary knowledge, but how to effectively integrate them and allow mutual refinement becomes a significant research gap. To address these challenges, we introduce COLA, a collaborative large–small model framework that enables seamless cooperation among semantic LLMs, task-specific fine-tuned SLMs, and structure-aware GNNs. COLA features a unique Consensus–Complement Coordination Mechanism (C3M), wherein its Mixture-of-Coordinators (MoC) architecturally aligns the LLM and SLM. Built upon this, a flexible graph-knowledge infusion strategy encourages the joint alignment and graph knowledge learning of textual representations. Extensive evaluations across nine diverse datasets show that COLA consistently achieves state-of-the-art performance, validating the effectiveness and generality of our collaborative paradigm.

Introduction

Recent years have witnessed rapid growth in the scale, diversity, and complexity of graph-structured data (Wu et al. 2020; Li et al. 2023; Wen et al. 2023). To effectively capture the underlying patterns in graphs, a plethora of advanced graph learning paradigms have been continuously developed (Wu, Zhang, and Fan 2023; Tu et al. 2024; Wang et al. 2024; Fan 2025). One of the most important trend is the enrichment of node attributes (Liu et al. 2022; Chen et al. 2023; Yang et al. 2024). Driven by advances in multimedia and natural language processing (Li et al. 2022; Lu et al. 2025), modern attributed graphs feature rich node attributes such as evolving features, multi-modal information and long-form text (Wu et al. 2023; Fang et al. 2025a; Chen et al. 2025). Textual attributes are pervasive in real-world applications, including citation networks (Yang, Cohen, and Salakhudinov 2016; Wang et al.) that interlink millions of scholarly articles, social networks (Cai et al. 2025; Zhuo et al. 2025) that relate billions of users and posts, and e-commerce networks (Wang et al. 2019) that connect products with rich descriptions and co-purchase patterns. Consequently, Text-Attributed Graphs (TAGs) have emerged as a central topic in modern graph learning (Yan et al. 2023). The surge of TAGs, however, has exposed a widening gap: while Graph Neural Networks (GNNs) excel at capturing structural patterns, they are built upon the assumption that node attributes are compact and weakly informative (Yu et al. 2024; Wang et al. 2025; Zhuang et al. 2025). Traditionally, classic GNNs rely on shallow text encoders like word2vec (Mikolov et al. 2013), which lack contextual understanding. This has fueled growing concern about GNNs' ability to fully exploit rich texts in modern graphs, highlighting the increasing necessity of collaborative modeling across different architectures.

The field of Large Language Models (LLMs) has un-

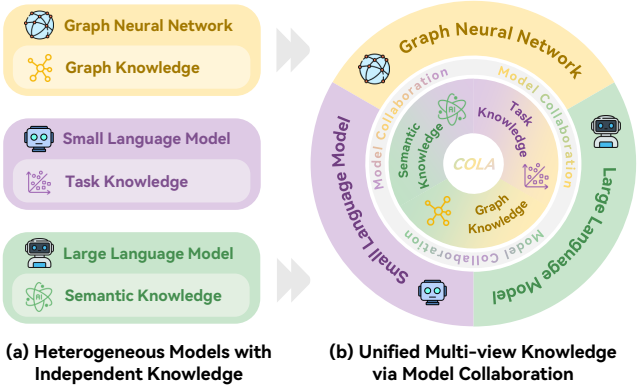

Figure 1: Illustration of the transition from independent models and knowledge sources model collaboration.

*Corresponding Author: Haishuai Wang.

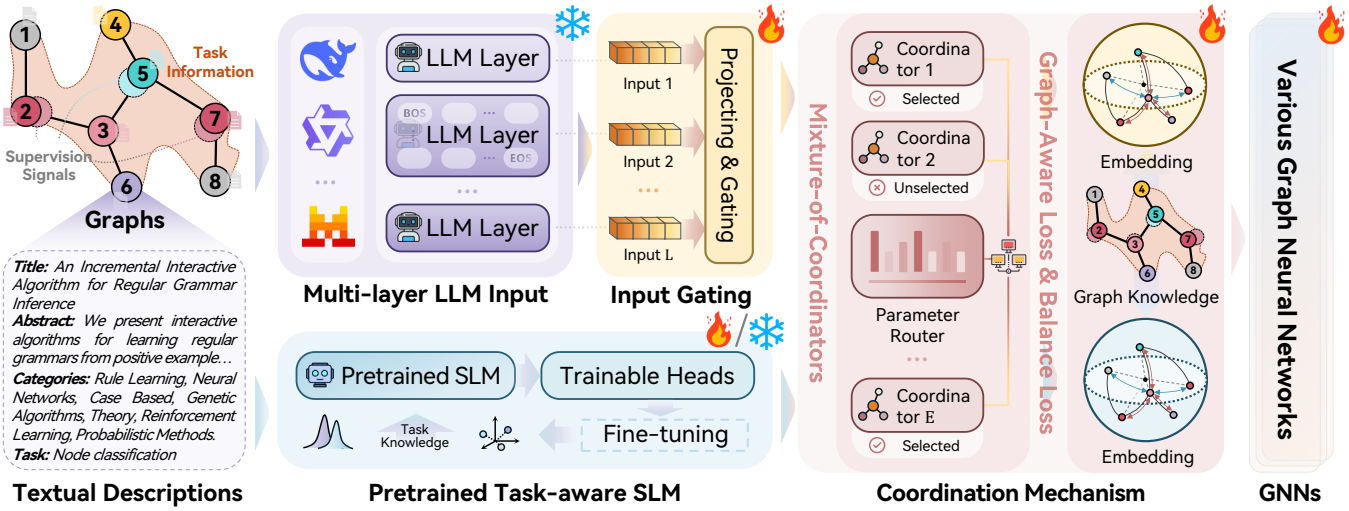

Figure 2: Overview of the proposed collaborative large-small model (COLA) framework.

dergone remarkable advancements in recent years (Achiam et al. 2023; Bai et al. 2023). By leveraging vast corpora and billions of parameters, these models excel at capturing contextual information and comprehensive semantics, performing zero-shot reasoning, and transferring across domains without explicit supervision (Naveed et al. 2023). Driven by these strengths, a growing line of work has sought to leverage LLMs for TAG learning tasks (Liu et al. 2025). Despite these outstanding advantages of LLMs, some fundamental obstacles remain. LLMs operate on sequential tokens and do not possess inductive bias for graph encoding. Consequently, they are natively graph-blind and lack the efficient mechanism to directly exploit topology (Huang et al. 2024). A few studies, like graph-oriented instruction tuning (Tang et al. 2024), have been devoted to solving this, but the significant training costs are the main bottleneck. When kept frozen, LLMs receive no task-specific gradient and are constrained by limited context windows that prevent them from observing the full textual neighborhood, thereby failing to discern domain-specific feature distributions (Wu et al. 2025a). Attempted remedies mainly focus on integrating LLMs with GNNs (Chen et al. 2024; Zhu et al. 2024), regarded by some research as graph-aware parameter-efficient fine-tuning. This pipeline is quite efficient and partially bridges the gap. However, the disparity between LLMs and GNNs remains substantial because they operate with fundamentally different mechanisms and possess distinct knowledge (Huang et al. 2024). Specifically, LLMs encode rich semantic knowledge, whereas GNNs hold task-specific and graph knowledge. Existing architectures still entangle the two via rigid and coarse-grained coupling, so that their complementary knowledge is merely juxtaposed rather than mutually refined, leaving their joint potential largely untapped.

Pre-trained Small Language Models (SLMs), e.g., the BERT family (Devlin et al. 2019), retain considerable semantic knowledge and strong text-encoding capacity, yet have been overshadowed by LLMs. Interestingly, recent trends have revived these smaller models by repurposing them as lightweight tuners for LLMs: fully fine-tuning them in downstream tasks injects domain-specific knowledge at a far lower cost than LLMs (Zhao et al. 2023). This resurgence is beginning to influence graph learning. Representative work such as TAPE (He et al. 2024) employs SLMs to get domain knowledge and complement LLMs, acting as a bridge between LLMs and GNNs. Intuitively, this approach is simple and reasonable, and the existing pipeline has achieved some empirical success, but this ignores two deeper needs. First, the LMs and GNNs should not just coexist but mutually refine via fine-grained interactions. Second, SLM and LLM must share the same latent space to avoid noisy misalignment. Therefore, existing methods are not able to fundamentally handle the complex dynamics between these models, ultimately suffering from knowledge fracture. Consequently, the central problem arises: *Can we devise a fine-grained model collaboration paradigm that unifies heterogeneous models and bridges this gap?*

This paper introduces COLA, a COllaborative LArge-small model framework for graph learning that orchestrates three complementary model types and unifies multi-view knowledge: semantic LLMs, task-aware fine-tuned SLMs, and structure-aware GNNs. At its core, COLA leverages a carefully designed Consensus-Complement Coordination Mechanism (C3M) to serve as a bridge between different sources of knowledge, rather than a direct coupling of large and small models. C3M consists of two main components. From the architectural lens, we introduce the Mixture of Coordinators (MoC), which combines a set of lightweight coordinators with a router. These components are shared between LLMs and SLMs. MoC adopts a divide-and-conquer strategy, selecting the most suitable combination of coordinators for each model and projecting the two types of embeddings into a unified space. This design models both consensus and complementary knowledge through overlapping and distinct coordinators. In terms of optimization, the embeddings from both models are projected onto a hypersphere and aligned through the joint regularization of a graph in-

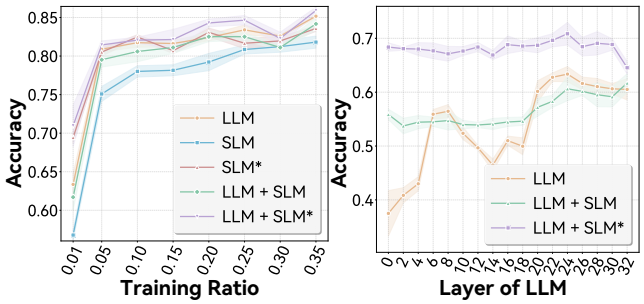

Figure 3: Accuracy on the Cora dataset versus training-label ratio for five encoder baselines (left); Accuracy of three baselines on Cora as a function of the LLM layer whose hidden state is exposed to the classifier (right).

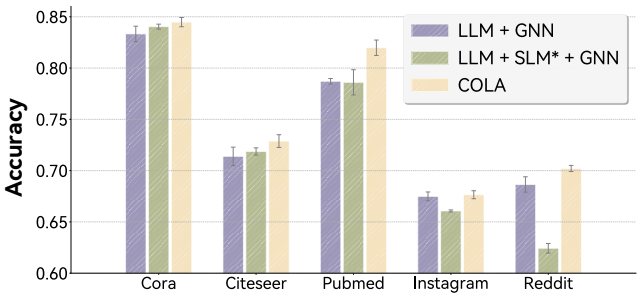

Figure 4: Accuracy of three baselines integrating GNN and LMs on five datasets.

fusion loss and a balancing loss, allowing the effective integration and harmonious interaction of all three types of knowledge. For language models, MoC operates on their multi-layer hidden states, fully leveraging the hierarchical semantic knowledge of the LMs and ensuring proper alignment with small models. Figure 1 illustrates the motivation of moving from independent models and knowledge to unifying multi-view knowledge via model collaboration. Figure 2 presents the overall workflow of the proposed COLA. Our main contributions are as follows:

- We provide a deeper examination of the interplay among LLMs, SLMs, and GNNs, identify the limitations of existing approaches, and outline a new model-collaboration-based graph-learning pipeline.
- We propose a principled yet novel large-small model collaboration framework that effectively unifies semantic, task-specific, and graph knowledge.
- COLA achieves consistent state-of-the-art performance across eight datasets spanning four distinct domains, empirically validating the efficacy of our paradigm.

Related Work

Recent advances in LLMs have sparked growing interest in their integration into TAG learning, owing to their strong contextual semantics and reasoning abilities (Jin et al. 2024). However, LLMs inherently process sequential tokens without graph-aware inductive biases, limiting their ability in graph learning, even if the neighborhood is explicitly provided (Huang et al. 2024). Prior studies have explored multiple ways, including leveraging LLMs as node encoders, explainers, and predictors (Wu et al. 2025a). Encoder-based methods outperform traditional shallow encoders (Zhu et al. 2024). Explainer methods utilize LLMs' generative capabilities to enhance node attributes (He et al. 2024). Predictor methods integrate graphs into LLMs via instruction tuning but exhibit inconsistent performance without abundant supervision (Chen et al. 2024; Tang et al. 2024). Recently, hybrid frameworks combine smaller language models (e.g., BERT) with LLMs and GNNs, improving empirical outcomes but still facing semantic mismatches and coarse-grained interactions (He et al. 2024; Fang et al. 2025b). Thus, a tailored collaborative paradigm to harmonize semantic, task-specific, and graph knowledge remains an open challenge.

Methodology

Notations

In this work, we mainly focus on the undirected textual graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{T} = \{t_1, \ldots, t_n\}$ associates each node v_i with a sequential text. For each node v_i, its neighborhood is denoted by $\mathcal{N}(v_i) = \{v_j : (v_i, v_j) \in \mathcal{V}\}$. A labeled subset $\mathcal{V}^{\mathrm{T}} \subset \mathcal{V}$ is provided with ground-truth labels $\{y_i \in \mathcal{C}\}_{v_i \in \mathcal{V}^{\mathrm{T}}}$.

Motivations

Previous work has utilized SLMs to bridge the knowledge gap between LLMs and GNNs, yet the intrinsic dynamics of model collaboration remain underexplored. To investigate this, we construct the following baselines: "LLM" or "SLM" use frozen LLM/SLM alone as the encoder, followed by a classification head; "LLM + SLM" concatenates the frozen LLM and SLM, shares a classification head, and two embeddings are simply summed. "SLM*" denotes SLM fine-tuned on node classification task. We evaluate five baselines on Cora under different label rates, training only task heads.

Figure 3 (left) shows that the vanilla SLM performs the worst, while SLM* and LLM each excel under different rates. It can be observed that simply combining LLM and SLM* yields consistent, little gains, while LLM + SLM offers no benefits. This suggests ***Finding 1****: Only a fine-tuned SLM can benefit the LLM due to their complementary knowledge, even with simple strategy.*

It is important to note that all LLM-based baselines report the best result across different LLM layers. Figure 3 (right) further shows how baseline performance varies with the chosen LLM layer at 1% label rate. Notably, LLM + SLM* is not always better than single models and can sometimes perform worse. The optimal LLM layer also varies across baselines with no clear trend. Thus, ***Finding 2****: SLM-LLM collaboration is critical but intricate, demanding fine-grained architectures to realize potential.*

Finally, we introduce GNNs to examine the dynamics among three types of models. For a practical setup, we

use the advanced TAPE to implement the "LLM + SLM* + GNN" setting and evaluate across a broader range of datasets. As shown in Figure 4, the results are surprising: while LLM + SLM* previously provided consistent gains, combining them with GNNs does not yield similar results, even with the advanced framework (TAPE). In contrast, simply using a frozen LLM as the text encoder in LLM + GNN delivers stable results. ***Finding 3****: Graph knowledge greatly increases the difficulty of collaboration, and current frameworks are still far from enabling effective model collaboration and heterogeneous knowledge integration.*

Encoding with Language Models

To extract rich semantics from the text associated with each node, researchers widely adopt language models as the text encoder. In what follows, we abstract the LLM and SLM as $\mathcal{M}_{\Theta}^{\text{LLM}}$ and $\mathcal{M}_{\Phi}^{\text{SLM}}$ while omitting some details, e.g., which hidden states or pooling strategies are used. As previously analyzed, LLMs lack task-relevant knowledge; therefore, we explicitly inject task-specific context and constraints into the prompt p_i for each text t_i. The detailed templates are provided in the Appendix. To fully leverage the capabilities of the frozen LLM and lay the groundwork for subsequent alignment, we extract node features at every selected layer that correspond to the generated response tokens:

$$\boldsymbol{x}_{i,(l)}^{\text{LLM}} = \mathcal{M}_{\Theta^{(l)}}^{\text{LLM}}(p_i), \tag{1}$$

where $\mathcal{M}_{\Theta^{(l)}}^{\text{LLM}} : \mathcal{T} \to \mathbb{R}^{d_{\text{LLM}}^{(l)}}$ denotes the first l layers of the LLM parameterized with $\Theta^{(l)}$ as a text encoder, and we apply mean pooling on the hidden states to extract embeddings. Note that we select the last several layers of the LLM in practice. We inject some task priors into the output embeddings solely via prompting, leaving all LLM parameters frozen. Consequently, the extracted embedding carries semantic knowledge and a modest amount of task knowledge.

For the SLM, we feed the text and obtain the corresponding node embedding as

$$\boldsymbol{x}_i^{\text{SLM}} = \mathcal{M}_{\Phi}^{\text{SLM}}(t_i), \tag{2}$$

where $\mathcal{M}_{\Phi}^{\text{SLM}} : \mathcal{T} \to \mathbb{R}^{d_{\text{SLM}}}$ is the SLM with parameter set Φ. In practice, we encode the text by adopting the hidden state of the [CLS] token from the final layer of the SLM. As *Finding 1* indicates, only an SLM endowed with task knowledge possesses the potential to collaborate with the LLM, because this synergy arises from the interaction between task and semantic knowledge. Compared with the LLM, the SLM is substantially more lightweight. Compared with the large model, $\mathcal{M}_{\Phi}^{\text{SLM}}$ is much lighter, so we can inject task-specific knowledge via full fine-tuning. First, we define a classification head $f_\Psi : d_{\text{SLM}} \to d^{|\mathcal{C}|}$, then the final prediction is given by $\hat{y}_i = f_\Psi(\boldsymbol{x}_i^{\text{SLM}})$. During fine-tuning, all parameters of the SLM encoder and the classification head are jointly optimized:

$$\Phi^*, \Psi^* = \underset{\Phi,\Psi}{\operatorname{argmin}} \ \mathbb{E}\big[\ell_{\text{CE}}(f_\Psi(\mathcal{M}_{\Phi}^{\text{SLM}}(t_i)), y_i)\big]. \tag{3}$$

where $v_i \in \mathcal{V}^{\text{T}}$. After the full parameter fine-tuning, the small language model $\mathcal{M}_{\Phi^*}^{\text{SLM}}$ not only retains the textual understanding capacity, but also internalises task knowledge that is crucial for downstream inference. After the SLM finetuning, we now have for each node i the embeddings $\{\boldsymbol{x}_{i,(l)}^{\text{LLM}}\}_{l=0}^{L}$ and $\boldsymbol{x}_i^{\text{SLM}^*}$ generated by the LLM and finetuned SLM. Further transformations and gating are applied to these embeddings so that they are projected into a common-dimensional space:

$$\tilde{\boldsymbol{x}}_{i,(l)}^{\text{LLM}} = \lambda^{(l)} p_{\mathcal{W}_{\text{LLM}}^{(l)}}(\boldsymbol{x}_{i,(l)}^{\text{LLM}}), \ \ \tilde{\boldsymbol{x}}_i^{\text{SLM}} = p_{\mathcal{W}_{\text{SLM}}}(\boldsymbol{x}_i^{\text{SLM}^*}), \tag{4}$$

where $\lambda_{(l)}$ is the gating coefficient, and $p_{\mathcal{W}_{\text{LLM}}^{(l)}} : \mathbb{R}^{d_{\text{LLM}}} \to \mathbb{R}^d$ and $p_{\mathcal{W}_{\text{SLM}}} : \mathbb{R}^{d_{\text{SLM}}} \to \mathbb{R}^d$ are the projectors. For convenience, we collect all parameters in this process as $\mathcal{W}_{\text{P}}$.

Coordination Mechanism

In this subsection, we shift from node attribute encoding to COLA's core mechanism. As demonstrated by *Finding 2*, the collaboration between LLM and SLM is intricate, which is attributed to heterogeneous embedding spaces and layer-dependent knowledge disparity. This problem can be even more intractable when considering the graph knowledge, which is pointed out in *Finding 3*. Therefore, to coordinate these large and small models, we first elaborate on the architecture design and propose MoC:

$$\mathcal{Q}_\Upsilon(\tilde{\boldsymbol{x}}_i) = \sum_{e=1}^{E} g_{\mathcal{W}_{\text{R}}}(\tilde{\boldsymbol{x}}_i)_e \ h_{\mathcal{W}_{\text{C}}^e}(\tilde{\boldsymbol{x}}_i), \tag{5}$$

where $\mathcal{Q}_\Upsilon : \mathbb{R}^d \to \mathbb{R}^{d'}$ denotes MoC module parameterized with $\Upsilon = \mathcal{W}_{\text{R}} \cup \mathcal{W}_{\text{C}}^1 \cup \cdots \cup \mathcal{W}_{\text{C}}^E$, $g_{\mathcal{W}_{\text{R}}} : \mathbb{R}^d \to \mathbb{R}^E$ is the router, $h_{\mathcal{W}_{\text{C}}^e} : \mathbb{R}^d \to \mathbb{R}^{d'}$, $e \in [E]$ are the coordinators. *Finding 2* shows that a simple, shared architectural module can deliver additional gains for SLM–LLM collaboration, yet the gains are limited because the shared component is coarse to reconcile their inherent differences. Inspired by the success of MoE in complex tasks like multimodal alignment (Ma et al. 2018; Zhang et al. 2025; Wu et al. 2025b), we design a bank of coordinators and employ a router to dispatch each node's embeddings to the most suitable coordinator combination. The architecture itself is not new and our novelty lies in its use for resolving misalignment in model collaboration. The embeddings originating from SLM and every selected layer of LLM are processed as

$$\boldsymbol{h}_i^{\text{SLM}} = \mathcal{Q}_\Upsilon(\tilde{\boldsymbol{x}}_i^{\text{SLM}}), \ \ \boldsymbol{h}_{i,(l)}^{\text{LLM}} = \mathcal{Q}_\Upsilon(\tilde{\boldsymbol{x}}_{i,(l)}^{\text{LLM}}). \tag{6}$$

Macroscopically, MoC acts as a shared component between LLM and SLM, projecting both into a unified space to distill consensual knowledge. Microscopically, the LLM and SLM can select distinct coordinator combinations, ensuring that each preserves complementary knowledge while maximizing downstream performance.

At this point, we resolve the interaction between the LLM and SLM at a finer architectural granularity; As *Finding 3* reveals, the situation could change once we couple the system to downstream GNNs. GNNs encode graph knowledge via a unique inductive bias that is fundamentally different from language models. Existing methods may overlook this distinction and simply "hard-wire" the two language models to the GNN, while the interaction between the LLM and

the SLM itself remains oblivious to graph knowledge. Motivated by this empirical observation, we claim that directly attaching the LLM and SLM to the GNN constitutes a hard injection of graph knowledge into the representations. Instead, we impose a graph-aware regularization during the alignment phase, allowing the LLM and SLM to softly infuse graph knowledge:

$$\mathcal{L}_{\text{graph}}(\mathcal{W}_{\text{P}}, \Upsilon) = -\mathbb{E}_{(v_i,v_j)\sim\mathcal{P}_{\mathcal{G}}(v_i,v_j)}\Big[\big(\tilde{\boldsymbol{h}}_i^{\text{LLM}}\big)^{\top}\tilde{\boldsymbol{h}}_j^{\text{SLM}}\Big], \tag{7}$$

where $\mathcal{P}_{\mathcal{G}}$ is the discrete probability distribution over the neighborhood $\mathcal{N}(v_i)$ induced by the graph $\mathcal{G}$. Here the two embeddings have been mapped to a unit hypersphere:

$$\tilde{\boldsymbol{h}}_i^{\text{LLM}} = \text{Map}\big(\boldsymbol{h}_i^{\text{LLM}}\big), \;\; \tilde{\boldsymbol{h}}_i^{\text{SLM}} = \text{Map}\big(\boldsymbol{h}_i^{\text{SLM}}\big), \tag{8}$$

where $\text{Map} : \mathbb{R}^{d'} \to \mathcal{S}^{d'-1}$ maps the embedding from Euclidean space onto hyperspherical space to eliminate magnitude bias that may hinder alignment between heterogeneous embeddings. Before that, the LLM embedding is obtained by gating fusion $\boldsymbol{h}_i^{\text{LLM}} = \sum_{l=0}^{L}\lambda^{(l)}\boldsymbol{h}_{i,(l)}^{\text{LLM}}$. Some optional constraints can be imposed on $\lambda^{(l)}$ to enhance efficacy. To further clarify how the regularizer infuses graph knowledge into both LLM and SLM, we provide a theorem:

Theorem 1 (Soft Graph Knowledge Infusion)**.** *Given a text-attributed graph $\mathcal{G}$, assume the node pair (v_i, v_j) is drawn from the following graph-based distribution*

$$\mathcal{P}_{\mathcal{G}}(v_i, v_j) = \alpha_{i,j} / \sum_{v_u\in\mathcal{N}(v_i)}\alpha_{i,u}, \;\; \forall v_i, v_j \in \mathcal{V}, \tag{9}$$

and assume all embeddings $\boldsymbol{h}_i$ lie in a unit hypersphere, then optimizing the following problem

$$\begin{aligned} \underset{\boldsymbol{H}}{\text{minimize}} \;\; & -\mathbb{E}_{(v_i,v_j)\sim\mathcal{P}_{\mathcal{G}}(v_i,v_j)}\Big[\tilde{\boldsymbol{h}}_i^{\top}\tilde{\boldsymbol{h}}_j\Big], \\ \textit{s.t.} \;\; & \tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j \in \mathcal{S}^{d'-1}, \; \forall v_i, v_j \in \mathcal{V}, \end{aligned} \tag{10}$$

by one step gradient descent with specific step size is equal to performing graph encoding by

$$\tilde{\boldsymbol{h}}_i^{(k+1)} = \alpha_{i,i}\tilde{\boldsymbol{h}}_i^{(k)} + \sum_{v_j\in\mathcal{N}(v_i)}\alpha_{i,j}\tilde{\boldsymbol{h}}_j^{(k)}, \tag{11}$$

where $\alpha_{i,j}$ is the aggregation coefficient.

Some detials and proof can be found in Appendix. Theorem 1 establishes that each message-passing operation of a GNN is equivalent to a single, prescribed optimization step applied to Problem (10). Consequently, directly attaching a GNN implicitly enforces a fixed step size and a predetermined number of iterations, which may be the root cause of unstable performance. In contrast, incorporating this objective as $\mathcal{L}_{\text{graph}}$ into the overall loss yields a soft, controllable mechanism for infusing graph knowledge. To encourage thorough alignment between the LLM and the SLM while preventing embedding collapse (inspired by (Wang and Isola 2020)), we enforce a uniform distribution of the two sets of embeddings over the hypersphere, leading to the following balancing loss

$$\mathcal{L}_{\text{bal}}(\mathcal{W}_{\text{P}}, \Upsilon) = \log\mathbb{E}_{(v_i,v_j)\overset{\text{i.i.d}}{\sim}\mathcal{P}_{\text{data}}}\Big[\big(e^{\big(\tilde{\boldsymbol{h}}_i^{\text{LLM}}\big)^{\top}\tilde{\boldsymbol{h}}_j^{\text{SLM}}/\tau}\big)\Big], \tag{12}$$

Dataset	# Nodes	# Edges	# Cls.	Type
Cora	2,708	5,429	7	Academic
Citeseer	3,186	4,277	6	Academic
Pubmed	19,717	44,338	3	Academic
WikiCS	11,701	215,863	10	Web Link
Instagram	11,339	144,010	2	Social
Reddit	33,434	198,448	2	Social
Books	41,551	358,574	12	E-Commerce
Photo	48,362	500,928	12	E-Commerce
arXiv	169,343	1,166,243	40	Academic

Table 1: Statistics of adopted benchmark datasets.

where τ is the temperature parameter that controls the concentration of the similarity distribution, and the node pairs are drawn uniformly from the entire node set.

Node Classification with GNNs

Finally, the aligned representations $\boldsymbol{h}_i^{\text{LLM}}$ and $\boldsymbol{h}_i^{\text{SLM}}$ are fed into a GNN classifier to produce the ultimate predictions.

$$\boldsymbol{Z}_{i,:} = \mathcal{F}_{\mathcal{W}_{\text{GNN}}}(\boldsymbol{H}; \mathcal{G})_{i,:} \tag{13}$$

where $\boldsymbol{Z}_{i,:}$ is the logits of node v_i, $\boldsymbol{H}$ is the fused embedding in which we simply adopt $\boldsymbol{H}_{i,:} = \beta\boldsymbol{h}_i^{\text{LLM}} + (1-\beta)\boldsymbol{h}_i^{\text{SLM}}$. In effect, MoC can also be viewed as an integral component of the GNN, thereby forming a decoupled-style GNN (Liu, Gao, and Ji 2020). Consequently, MoC mediates among the three parties, captures consensus from multi-view knowledge, and enables the large and small models to collaborate effectively. For the overall model training, we have the following losses:

$$\mathcal{L}_{\text{all}}(\mathcal{W}_{\text{GNN}}, \mathcal{W}_{\text{P}}, \Upsilon) = \mathcal{L}_{\text{task}} + \mu(\mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{bal}}), \tag{14}$$

where the task loss is specified as cross-entropy loss.

Experiments

We conduct extensive experiments to validate the effectiveness of COLA and our claims. Specifically, we answer the following research questions: **RQ1**: How does COLA framework perform in comparison to the representative GNN-LLM methods? **RQ2**: Does C3M promote large-small model collaboration and knowledge integration? **RQ3**: How does the selection strategy for C3M hyperparameters affect the performance of COLA?

Experimental Setting

Datasets. All evaluations are performed on a diverse collection of TAG datasets, spanning multiple domains to facilitate a comprehensive analysis across real-world graphs. We adopt the following benchmark datasets (Wu et al. 2025a): citation networks Cora, Citeseer, Pubmed, arXiv, web, and social networks WikiCS, Instagram, Reddit, and e-commerce network Books. All these datasets are equipped with official texts. Detailed statistics are reported in Table 1.
Baselines. We include twelve baselines spanning three categories: the classical GNNs, i.e., GCN (Kipf and Welling 2017), GraphSAGE (SAGE) (Hamilton, Ying, and Leskovec

	Cora	Citeseer	Pubmed	WikiCS	Instagram	Reddit	Books	Photo	arXiv	Avg.
Classical GNNs with Shallow Encoder										
GCN	82.30±0.19	70.55±0.32	78.94±0.27	79.86±0.19	63.50±0.11	61.44±0.38	68.79±0.46	69.25±0.81	71.39±0.28	71.78
SAGE	82.27±0.37	69.56±0.43	77.88±0.44	79.67±0.25	63.57±0.10	56.65±0.33	72.01±0.15	78.50±0.15	71.21±0.18	72.37
GAT	81.30±0.67	69.94±0.74	78.49±0.70	80.06±0.65	63.56±0.04	60.60±1.17	74.35±0.16	80.40±0.45	71.57±0.25	73.36
Small Language Model Based										
SenBERT	66.66±1.14	60.52±1.62	36.04±2.92	77.77±0.65	59.00±1.17	56.05±0.41	77.17±0.15	73.89±0.97	72.66±0.24	64.42
RoBERTa	72.24±1.16	66.68±2.03	42.32±1.56	76.81±1.04	63.52±0.44	59.27±0.34	84.62±0.16	74.79±1.13	74.12±0.24	68.26
GLEM	81.30±0.88	68.80±2.46	81.70±1.07	76.43±0.55	60.25±3.66	55.13±1.41	83.28±0.39	76.93±0.49	73.55±0.22	73.04
Large Language Model Based										
GCN_{LLM}	83.33±0.75	71.39±0.90	78.71±0.45	80.94±0.16	67.49±0.43	68.65±0.75	83.03±0.34	84.84±0.47	74.39±0.31	76.97
LLM_{IT}	67.00±1.16	54.26±2.20	80.99±0.43	75.02±1.38	41.83±1.43	54.09±1.02	80.92±1.26	71.28±1.81	**76.08**	66.83
GraphGPT	64.72±1.56	64.58±1.55	70.34±2.27	75.41±2.52	62.88±2.14	58.25±0.37	81.13±1.01	77.48±0.76	75.15±0.14	69.99
LLaGA	78.94±1.16	62.61±2.09	65.91±2.09	76.47±2.20	65.84±0.72	70.10±0.38	83.47±0.77	84.44±0.49	74.49±0.23	73.59
TAPE	84.04±0.24	71.87±0.35	78.61±1.23	**81.94±0.16**	66.07±0.10	62.43±0.47	84.92±0.26	86.46±0.12	74.96±0.14	76.81
ENGINE	84.22±0.46	72.14±0.74	77.84±0.27	80.94±0.27	67.14±0.46	69.67±0.16	82.89±0.16	84.33±0.57	74.69±0.36	77.10
COLA	**85.15±0.60**	**72.94±0.28**	**82.68±0.68**	81.54±0.36	**68.12±0.50**	**70.21±0.30**	**85.13±0.21**	**87.03±0.20**	75.19±0.30	**78.67**

Table 2: Semi-supervised Node classification accuracy (%) of various methods across eight benchmark datasets. Results are averaged over four runs (mean ± std). The best and second-best results per dataset are highlighted.

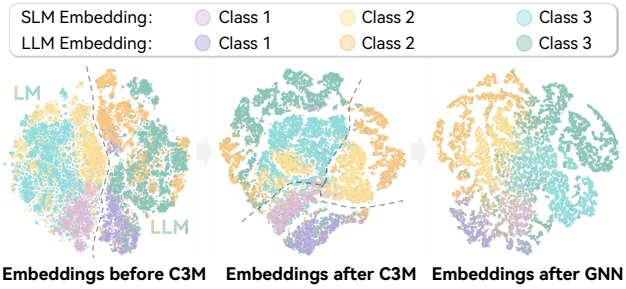

Figure 5: t-SNE visualization of node embeddings on the Pubmed dataset in different stages of COLA.

2017) and GAT (Velickovic et al. 2018), each paired with the shallow features from raw text and thus exploit only structural information; the small-language-model approaches SenBERT (Reimers and Gurevych 2019), RoBERTa (Liu et al. 2019) and GLEM (Zhao et al. 2023), where the first two are canonical sentence-level transformers while GLEM integrates SLMs with GNNs in a hybrid framework; and the LLM-based methods LLM_{IT} (an instruction-tuned LLM (Wu et al. 2025a)), GCN_{LLM} (a straightforward replacement of the text encoder with an LLM), and the state-of-the-art GraphGPT (Tang et al. 2024), LLaGA (Chen et al. 2024), TAPE (He et al. 2024) and ENGINE (Zhu et al. 2024), which jointly leverage LLMs and GNNs—typically treating the LLM as an encoder or predictor—and notably TAPE simultaneously employs LLM, SLM and GNN.

Other Details. For the arXiv dataset, we adopt the official split provided by (Kipf and Welling 2017; Wu et al. 2025a). For all other datasets, we follow the classic semi-supervised classification split (Hu et al. 2020). To ensure fairness, each method that combines a GNN with an LLM or SLM uses Mistral-7B (Jiang et al. 2023) as the LLM component, the 355 M-parameter RoBERTa (Liu et al. 2019) as the SLM component, and GCN as the GNN component. Note that, the preliminary experiments of Figures 3 and 4 are also conducted under this setting. All reported means and standard deviations are computed over four independent runs. All experiments were conducted on the NVIDIA A100 GPU (80GB). Further details are provided in the Appendix.

Node Classification Performance (RQ1)

Table 2 reports the node classification results of COLA and all baseline methods on nine datasets, including the large-scale graph arXiv. We draw the following observations and conclusions: First, GNNs equipped with static, shallow text encoders perform significantly worse than recent methods that integrate language models with GNNs. This suggests that shallow text encoders may struggle to capture the complex semantic information present in node texts, whereas language models can extract richer and more contextually meaningful representations. Thus, leveraging language models to enhance TAG learning is non-trivial. On the other hand, using a language model alone consistently underperforms compared to GNNs with a shallow encoder. This indicates that the unique graph-specific inductive bias of GNNs, i.e. graph knowledge, remains crucial. Even fine-tuned LLM (LLM_{IT}) cannot fully bridge this gap by relying solely on semantic information from text, highlighting the significance of reasonably combining them. Lastly, COLA consistently achieves leading performance across most graphs and obtains the highest average score, which demonstrates the effectiveness of our collaborative framework. It is worth noting that TAPE, which also leverages LLMs, SLMs, and GNNs, performs well on several datasets, highlighting the potential of large and small model collaboration. Interestingly, GCN_{LLM} achieves the third-best average performance, while TAPE underperforms on certain datasets (e.g., Instagram and Reddit). This indicates that existing approaches are still unable to effectively handle the complex coupling among LLMs, SLMs, and GNNs.

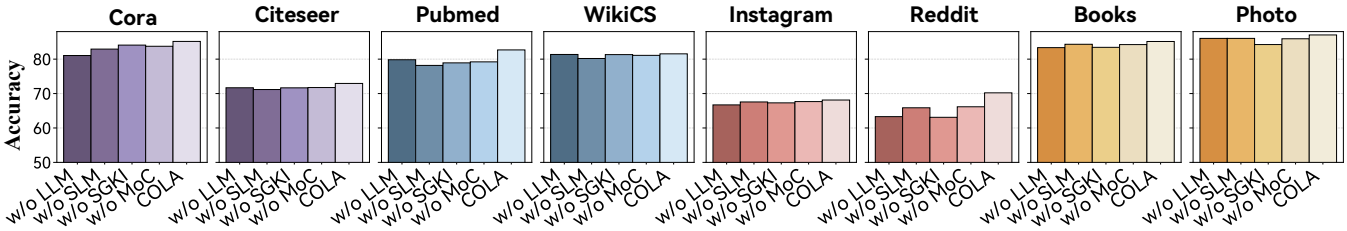

Figure 6: Ablation study for four modules of our proposed COLA on eight datasets.

Embedding Visualization (RQ2)

To better understand how COLA integrates knowledge, we extracted the embeddings from both the LLM and SLM before and after the C3M module, as well as the final node embeddings learned by the GNN. These embeddings are visualized using t-SNE in Figure 5, where light-colored points represent node embeddings from the SLM and dark-colored points denote those from the LLM, with different color schemes indicating different classes. It can be observed that, prior to C3M, the two types of embeddings are separated by clear boundaries, indicating that they capture distinct feature spaces. After passing through the C3M module, the embeddings obtain clustered patterns, demonstrating that C3M effectively aligns semantics with the task knowledge. In addition, the post-C3M embeddings exhibit clear class boundaries and the two embeddings have similar distributions. This experiment clearly illustrates the process of interaction among the three types of knowledge.

Ablation Study (RQ2)

We conduct ablation experiments to evaluate the importance of the LLM and SLM, as well as the two key components in C3M. As shown in Figure 6, removing either MoC or SGKI leads to a clear drop across all datasets. This demonstrates that both modules are essential: MoC for aligning and routing semantic and task knowledge, and SGKI for softly injecting structural information. Removing either the LLM or SLM also leads to a substantial drop. Note that this implies SGKI is also removed; however, the result is not always worse than removing SGKI alone, suggesting that misaligned LLM and SLM may conflict with each other. The full COLA model consistently achieves the best results, highlighting the necessity of all these components.

Parameter Sensitivity (RQ3)

We further examine how COLA's performance changes with respect to the trade-off parameter μ which balances the task loss with the graph infusion and balancing losses, and the temperature τ which controls the sharpness of the loss. Figure 7 shows that COLA demonstrates a high degree of stability across a broad range of τ values on representative datasets, suggesting that the model is not overly sensitive to the temperature setting once set within a reasonable interval. However, the performance is more sensitive to μ, reflecting its critical role in mediating the strength of knowledge alignment and structural regularization. As μ varies, we observe that accuracy initially increases, reaches an optimum,

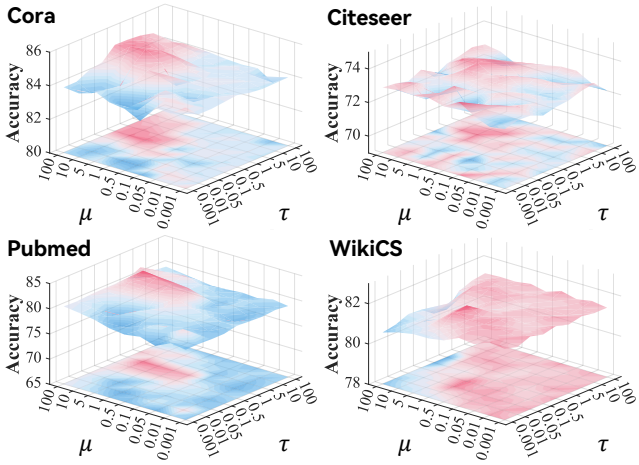

Figure 7: Sensitivity analysis of COLA with respect to the trade-off parameter μ and the temperature coefficient τ.

and then gradually declines if overemphasized—indicating that while some tuning of μ is beneficial, the degradation is smooth and COLA remains robust across a wide operational range. These results provide practical guidance: default hyperparameter choices are sufficient for strong performance, and the model does not require laborious tuning.

Conclusion

In this work, we presented COLA, a principled framework for unifying multi-view knowledge in text-attributed graph learning via collaborative modeling. By seamlessly integrating a frozen LLM, a task-aware fine-tuned SLM, and a structure-aware GNN through the novel Consensus–Complement Coordination mechanism, COLA effectively bridges the gap between semantic, task-specific, and structural knowledge. Our Mixture-of-Coordinators module and soft graph-knowledge infusion enable fine-grained knowledge alignment and flexible interaction among all components. Extensive experiments across diverse benchmarks demonstrate that COLA consistently achieves state-of-the-art performance and is robust to hyperparameter choices. These results underscore the potential of model collaboration as a general paradigm for advanced graph learning. We believe COLA paves the way for future research on scalable, flexible, and unified model collaboration in complex graph-based systems. One possible limitation is that we did not discuss the impact of heterophily.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62202422, 62372408, 62376254, 32341017, and 32341018) and the Zhejiang Province Vanguard Goose-Leading Initiative (No. 2025C01114).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Cai, J.; Zhang, Y.; Liu, F.; and Ng, S. 2025. Leveraging Diffusion Model as Pseudo-Anomalous Graph Generator for Graph-Level Anomaly Detection. In *Forty-second International Conference on Machine Learning*.

Chen, J.; Gao, K.; Li, G.; and He, K. 2023. NAGphormer: A Tokenized Graph Transformer for Node Classification in Large Graphs. In *The Eleventh International Conference on Learning Representations*.

Chen, R.; Zhao, T.; Jaiswal, A. K.; Shah, N.; and Wang, Z. 2024. LLaGA: Large Language and Graph Assistant. In *Proceedings of the 41st International Conference on Machine Learning*, 7809–7823.

Chen, W.; Zhao, Z.; Yao, J.; Zhang, Y.; Bu, J.; and Wang, H. 2025. Multi-modal Medical Diagnosis via Large-small Model Collaboration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30763–30773.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. 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*, 4171–4186.

Fan, J. 2025. Graph Minimum Factorization Distance and Its Application to Large-Scale Graph Data Clustering. In *Forty-second International Conference on Machine Learning*.

Fang, H.; Wang, H.; Gao, Y.; Zhang, Y.; Bu, J.; Han, B.; and Lin, H. 2025a. InsGNN: Interpretable spatio-temporal graph neural networks via information bottleneck. *Information Fusion*, 119: 102997.

Fang, Z.; Cai, Z.; Zheng, Y.; Du, S.; Tan, Y.; and Wang, S. 2025b. HiTuner: Hierarchical Semantic Fusion Model Fine-Tuning on Text-Attributed Graphs. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 5110–5117.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

He, X.; Bresson, X.; Laurent, T.; Perold, A.; LeCun, Y.; and Hooi, B. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. In *The Twelfth International Conference on Learning Representations*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*.

Huang, J.; Zhang, X.; Mei, Q.; and Ma, J. 2024. Can LLMs Effectively Leverage Graph Structural Information through Prompts, and Why? *Transactions on Machine Learning Research*.

Jiang, W.; Touvron, H.; Jaszczur, S.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Jin, B.; Liu, G.; Han, C.; Jiang, M.; Ji, H.; and Han, J. 2024. Large Language Models on Graphs: A Comprehensive Survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12): 8622–8642.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*.

Li, S.; Liu, F.; Jiao, L.; Chen, P.; Liu, X.; and Li, L. 2022. MFNet: A novel GNN-based multi-level feature network with superpixel priors. *IEEE Transactions on Image Processing*, 31: 7306–7321.

Li, X.; Sun, Y.; Sun, Q.; Ren, Z.; and Sun, Y. 2023. Cross-view graph matching guided anchor alignment for incomplete multi-view clustering. *Information Fusion*, 100: 101941.

Liu, M.; Gao, H.; and Ji, S. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 338–348.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. S. 2022. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Liu, Y.; Zhang, G.; Wang, K.; Li, S.; and Pan, S. 2025. Graph-augmented large language model agents: Current progress and future prospects. *IEEE Intelligent Systems*.

Lu, Z.; Yu, Y.; Ma, L.; Nie, F.; and Wang, R. 2025. Capturing Individuality and Commonality Between Anchor Graphs for Multi-View Clustering. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 5860–5868.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations*.

Naveed, H.; Khan, A. U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; and Mian, A. 2023. A

comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Reimers, N.; and Gurevych, I. 2019. 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*, 3980–3990.

Tang, J.; Yang, Y.; Wei, W.; Shi, L.; Su, L.; Cheng, S.; Yin, D.; and Huang, C. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 491–500.

Tu, W.; Guan, R.; Zhou, S.; Ma, C.; Peng, X.; Cai, Z.; Liu, Z.; Cheng, J.; and Liu, X. 2024. Attribute-Missing Graph Clustering Network. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, 15392–15401.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations*.

Wang, L.; He, D.; Zhang, H.; Liu, Y.; Wang, W.; Pan, S.; Jin, D.; and Chua, T.-S. 2024. Goodat: Towards test-time graph out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15537–15545.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 9929–9939.

Wang, X.; Ji, H.; Shi, C.; Wang, B.; Ye, Y.; Cui, P.; and Yu, P. S. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*, 2022–2032.

Wang, Y.; Liu, Y.; Liu, N.; Miao, R.; Wang, Y.; and Wang, X. 2025. AdaGCL+: An Adaptive Subgraph Contrastive Learning Towards Tackling Topological Bias. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, Y.; Wang, J.; Wang, J.; Cui, M.; Gao, J.; Guo, J.; et al. ???? Hybrid-Collaborative Augmentation and Contrastive Sample Adaptive-Differential Awareness for Robust Attributed Graph Clustering. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Wen, Y.; Liu, S.; Wan, X.; Wang, S.; Liang, K.; Liu, X.; Yang, X.; and Zhang, P. 2023. Efficient multi-view graph clustering with local and global structure preservation. In *Proceedings of the 31st ACM international conference on multimedia*, 3021–3030.

Wu, X.; Shen, Y.; Ge, F.; Shan, C.; Jiao, Y.; Sun, X.; and Cheng, H. 2025a. When Do LLMs Help With Node Classification? A Comprehensive Analysis. In *Forty-second International Conference on Machine Learning*.

Wu, Z.; Lin, X.; Lin, Z.; Chen, Z.; Bai, Y.; and Wang, S. 2023. Interpretable Graph Convolutional Network for Multi-View Semi-Supervised Learning. *IEEE Transactions on Multimedia*, 25: 8593–8606.

Wu, Z.; Lu, J.; Yu, J.; Zhou, S.; Pi, Y.; and Wang, H. 2025b. Divide and Conquer: Coordinating Multiplex Mixture of Graph Learners to Handle Multi-Omics Analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 6615–6623.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Wu, Z.; Zhang, Z.; and Fan, J. 2023. Graph Convolutional Kernel Machine versus Graph Convolutional Networks. In *Advances in Neural Information Processing Systems*, volume 36, 19650–19672.

Yan, H.; Li, C.; Long, R.; Yan, C.; Zhao, J.; Zhuang, W.; Yin, J.; Zhang, P.; Han, W.; Sun, H.; Deng, W.; Zhang, Q.; Sun, L.; Xie, X.; and Wang, S. 2023. A Comprehensive Study on Text-attributed Graphs: Benchmarking and Rethinking. In *Advances in Neural Information Processing Systems*, volume 36, 17238–17264.

Yang, Y.; Sun, Y.; Wang, S.; Guo, J.; Gao, J.; Ju, F.; and Yin, B. 2024. Graph neural networks with soft association between topology and attribute. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 9260–9268.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, 40–48.

Yu, J.; Wu, Z.; Cai, J.; Jia, A. L.; and Fan, J. 2024. Kernel Readout for Graph Neural Networks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2505–2514.

Zhang, Y.; Cai, J.; Wu, Z.; Wang, P.; and Ng, S.-K. 2025. Mixture of Experts as Representation Learner for Deep Multi-View Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22704–22713.

Zhao, J.; Qu, M.; Li, C.; Yan, H.; Liu, Q.; Li, R.; Xie, X.; and Tang, J. 2023. Learning on Large-scale Text-attributed Graphs via Variational Inference. In *The Eleventh International Conference on Learning Representations*.

Zhu, Y.; Wang, Y.; Shi, H.; and Tang, S. 2024. Efficient tuning and inference for large language models on textual graphs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 5734–5742.

Zhuang, S.; Wu, Z.; Chen, Z.; Dai, H.; and Liu, X. 2025. Refine then Classify: Robust Graph Neural Networks with Reliable Neighborhood Contrastive Refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 13473–13482.

Zhuo, J.; Liu, Y.; Lu, Y.; Ma, Z.; Fu, K.; Wang, C.; Guo, Y.; Wang, Z.; Cao, X.; and Yang, L. 2025. Dualformer: Dual graph transformer. In *The Thirteenth International Conference on Learning Representations*.